

that a smart card 11 may carry all or only some of these components. The display device 4 and keypad 3 obviously form part of the mobile phone. The direct hardwired connections 31, 41 between the display 4 and keypad 3, respectively, and the memory 14 must also be implemented at least in part in the mobile phone. These lines in addition to the I/O bus will therefore be present in the terminal adapters 110 and 111 of the mobile phone and the card. However, the switches 32, 42, could either be integrated in the mobile phone platform or be disposed on the smart card, that is on either side of the smart card adapters 110, 111 provided in the mobile phone and the card 11, respectively. It is, however, preferable that the switches be provided within the mobile phone to simplify their control by the mode selector 7. Preferably, the secure memory 14 and the CPU 12 are implemented on the mobile phone platform to ensure minimum delay when accessing the keyboard 3 and display 4. The applications 131 and cryptographic module 15 with the stored keys could be provided on the smart card 11, in which case the application could usefully be downloaded into a reserved memory area on the mobile phone for execution.

While in the embodiment described above, the secure part 10 includes a dedicated CPU, this need not be the case. The central controller of the mobile phone part 1 may accomplish the tasks effected by this CPU. Furthermore, the secure memory 14 need not be a separate component, but could be a reserved area of memory provided in the mobile phone part. Access to this reserved memory area would be provided only for the cryptographic module 15 and the display 4 and keypad 3 when the phone is in secure mode.

The arrangement according to the invention is not limited to mobile phones, but may be employed in any mobile or indeed fixed terminal having access to a communications network. Examples of these include terminals, PCs, laptop computers, electronic organisers, personal digital assistants and internet access

devices. The division of these devices into a normal operating part and a secure part will be analogous to that illustrated in the figures for the mobile phone part 1 and the secure part 10. In these other devices, a dedicated secure memory or memory area will be provided that can be accessed only by the cryptographic module and the display and possibly also the keyboard in the secure mode of operation. As for the mobile phone described above, the secure part 10 could be provided at least in part by a smart card. Also, the secure part need not dispose of a dedicated CPU but could use the processor already present in the platform device.

10

## Claims:

- 5        1.     An arrangement for effecting secure transactions including  
         a device having an interface to at least one communications network,  
         and having at least two modes of operation,  
         the device further including:  
         means (1) for controlling the device in a first mode of operation,  
10       display means (4) coupled to said control means (1) in said first mode  
         of operation, and  
         secure means (10) for controlling the device in a second mode of  
         operation  
         characterised in that said secure means (10) includes storage means  
15       (14) for storing secure data, and said display means (4) are directly  
         coupled to said storage means (14) in said second mode of operation.
2.     An arrangement as claimed in claim 2, further characterised by input  
         means (3) coupled to said control means (1) in a first mode of  
20       operation and directly coupled to said storage means (14) in said  
         second mode of operation.
3.     An arrangement as claimed in claim 1 or 2, further characterised by  
         switching means (7, 8) for switching the mode of operation of said  
25       device between said first and second modes.
4.     An arrangement as claimed in any previous claim, characterised in that  
         display means are coupled to said storage means (14) by a hardwired  
         connection comprising a switch (42).  
30

5. An arrangement as claimed in any previous claim, characterised in that said input means display is coupled to said storage means (14) by a hardwired connection comprising a switch (32).
- 5 6. An arrangement as claimed in claims 3 and 4 or 5, characterised in that said switch (32, 42) is controlled by said switching means (7, 8).
7. An arrangement as claimed in claim 3, characterised in that said switching means include a manual switch (8) on said device.
- 10 8. An arrangement as claimed in any previous claim, further characterised by means (9) for indicating the operating mode of the device.
- 15 9. An arrangement as claimed in claim 8, characterised in that said indicator means includes an LED (9).
10. An arrangement as claimed in any previous claim, characterised in that said secure means (10) includes at least one application (13, 131, 132) for processing secure data.
- 20 11. An arrangement as claimed in claim 10, characterised in that said secure means (10) includes processing means (12) for executing said secure data application (131, 132) when the device is in said secure mode of operation.
- 25 12. An arrangement as claimed in claim 11, characterised in that said processing means (12) are inoperative in the normal mode of operation.
- 30 13. An arrangement as claimed in any previous claim, characterised in that said secure means (10) includes a data generation means (15) coupled



to said storage means (14) for generating secure data.

14. An arrangement as claimed in any previous claim, characterised in that at least part of said secure means (10) is provided on a separate carrier (11) adapted to connect with said device.

15. An arrangement as claimed in claim 11, characterised in that said carrier (11) is a smart card.

16. An arrangement as claimed in any previous claim, characterised in that said device is a mobile communications device.

17. An arrangement for effecting secure transactions characterised by including at least one application (13, 131, 132) for processing secure data, processing means (12) for executing said application, and secure storage means (14) for storing secure data, further comprising means (111) for connecting with a communications device having display means (4) wherein said connection means (111) includes at least one line (41) for directly coupling said storage means (14) with said display means.

18. An arrangement as claimed in claim 17, characterised in that said connection means (111) includes at least one further line for directly coupling said storage means (14) with input means (3) present in said communications device.

19. An arrangement as claimed in claim 17 or 18, further including data generation means (15) for generating secure data.

20. A carrier characterised by incorporating an arrangement as claimed in any one of claims 17 to 19.

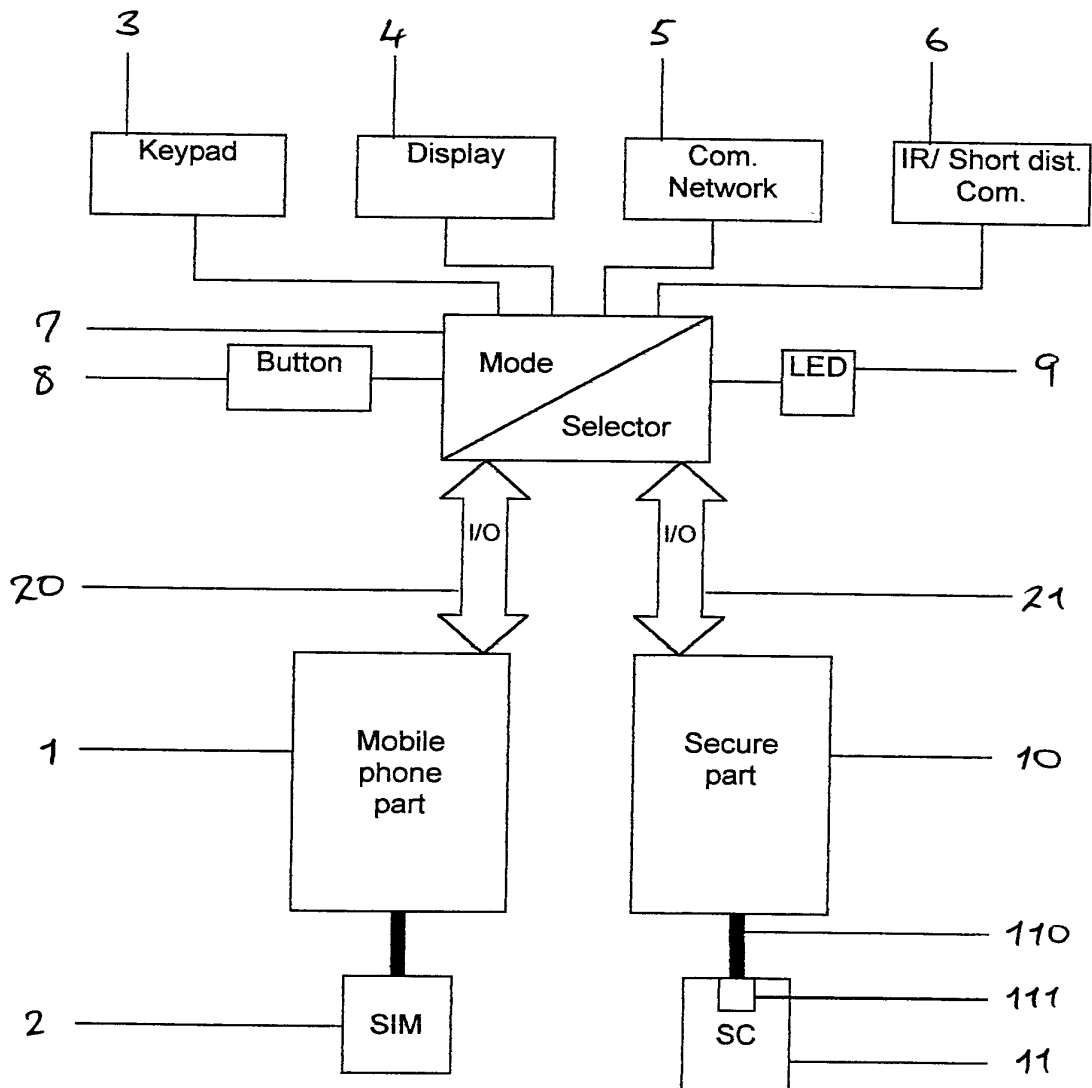


Fig. 1

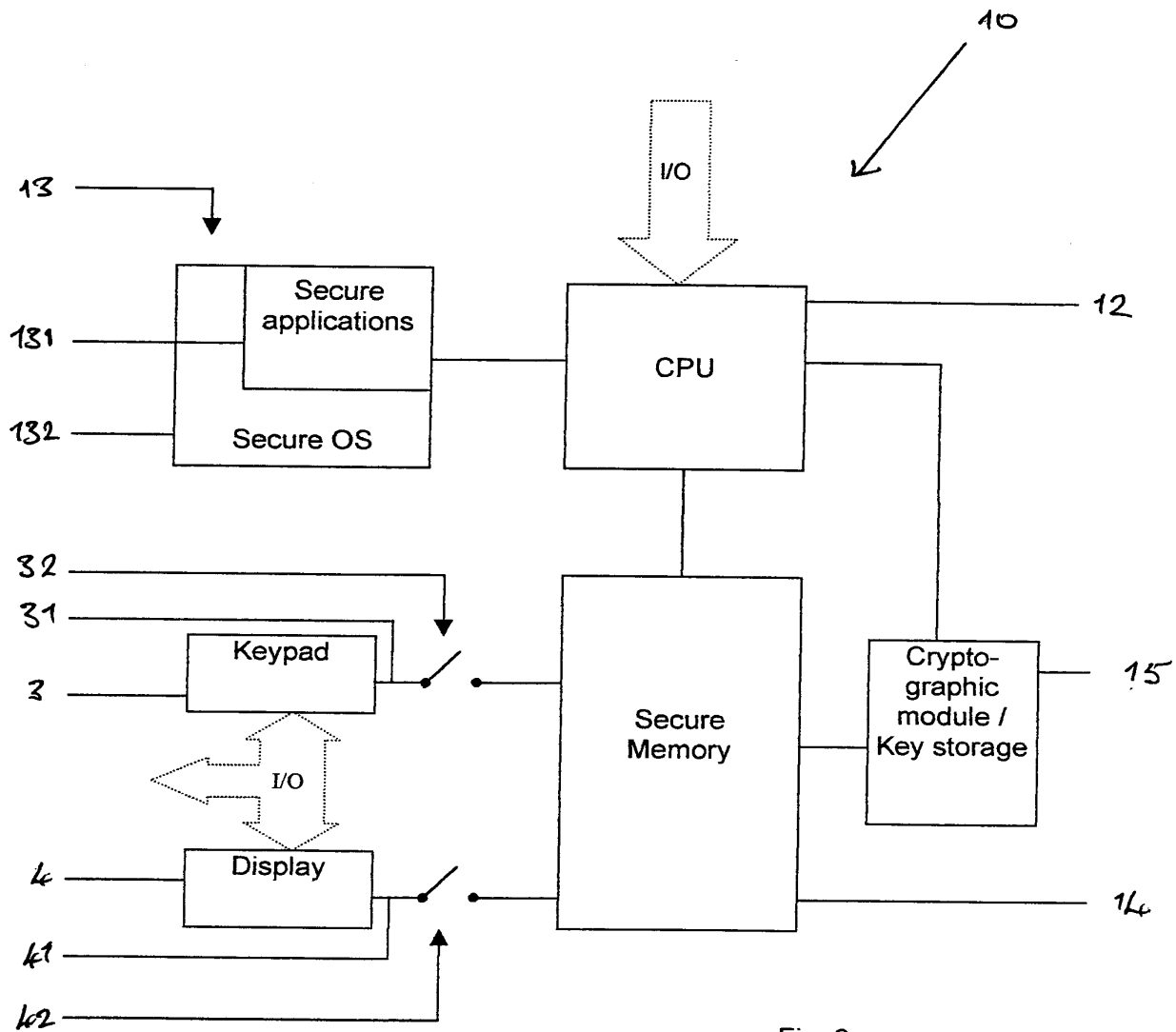


Fig. 2

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 00/01642

## A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04Q 7/32

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)

IPC7: H04Q

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

SE,DK,FI,NO classes as above

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

INSPEC, IEEE

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0436518 A2 (MOTOROLA, INC.), 10 July 1991 (10.07.91), column 3, line 11 - line 27; column 3, line 53 - column 4, line 27 --	1-20
X	US 5913175 A (F. PINAULT), 15 June 1999 (15.06.99), column 3, line 49 - column 4, line 13; column 11, line 3 - line 19 --	1,2,8-20
Y	--	3-7
Y	EP 0617528 A2 (KABUSHIKI KAISHA TOSHIBA), 28 Sept 1994 (28.09.94), column 3, line 57 - column 4, line 34; column 5, line 39 - line 41; column 7, line 9 - line 47, figure 2 --	3-7

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

\* Special categories of cited documents:

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

"E" earlier application or patent 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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

7 November 2000

Date of mailing of the international search report

06-12-2000

Name and mailing address of the ISA/

Swedish Patent Office

Box 5055, S-102 42 STOCKHOLM

Facsimile No. +46 8 666 02 86

Authorized officer

Peter Göransson / MRO

Telephone No. +46 8 782 25 00

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 00/01642

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4368357 A (R.J. GURAK), 11 January 1983 (11.01.83), column 4, line 8 - line 26  --	1-10
P,X	US 6084968 A (P.R. KENNEDY ET AL), 4 July 2000 (04.07.00), column 3, line 7 - line 36  -- -----	1,2,8-20

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

03/10/00

International application No.  
PCT/SE 00/01642

EP	0436518	A2	10/07/91	AT	146640	T	15/01/97
				CA	2033651	A,C	06/07/91
				DE	69123608	D,T	12/06/97
				FI	102928	B	00/00/00
				FI	906424	A	06/07/91
				NO	178360	B,C	27/11/95
				NO	910044	D	00/00/00
				US	5060264	A	22/10/91
-----							
US	5913175	A	15/06/99	AU	716887	B	09/03/00
				AU	7414796	A	26/06/97
				CA	2193712	A	22/06/97
				EP	0781065	A	25/06/97
				FR	2742959	A,B	27/06/97
				JP	9187081	A	15/07/97
-----							
EP	0617528	A2	28/09/94	AU	661228	B	13/07/95
				AU	5790694	A	06/10/94
				CA	2119823	A	26/09/94
				CN	1097533	A	18/01/95
				FI	941373	A	26/09/94
				JP	6284464	A	07/10/94
-----							
US	4368357	A	11/01/83	NONE			
-----							
US	6084968	A	04/07/00	NONE			
-----							

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

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
26 September 2002 (26.09.2002)

PCT

(10) International Publication Number  
**WO 02/076127 A1**

(51) International Patent Classification<sup>7</sup>: **H04Q 7/32**,  
G06F 1/00

(21) International Application Number: PCT/US02/07693

(22) International Filing Date: 15 March 2002 (15.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/276,380 16 March 2001 (16.03.2001) US  
09/826,742 5 April 2001 (05.04.2001) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US];  
5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventor: **MAURO, Anthony**; 12615 Darkwood Road,  
San Diego, CA 92129 (US).

(74) Agents: **WADSWORTH, Philip, R.** et al.; Qualcomm In-  
corporated, 5775 Morehouse Drive, San Diego, CA 92121-  
1714 (US).

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

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING SECURE PROCESSING AND DATA STORAGE FOR A WIRELESS COMMUNICATION DEVICE

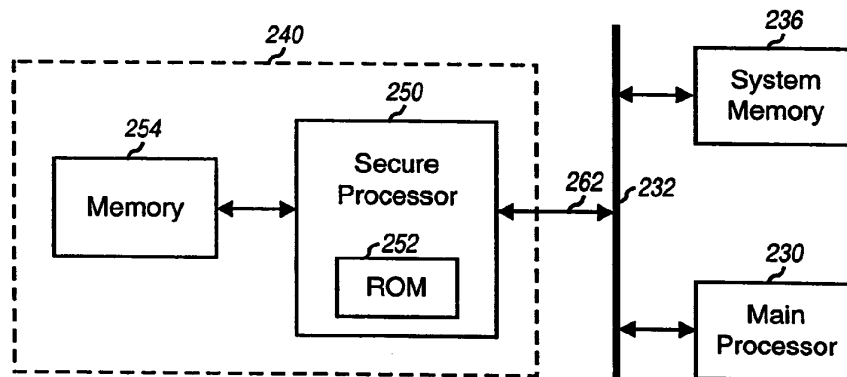


Exhibit E8

EP 3 229 099 (17000713.2)

BARDEHLE PAGENBERG  
Our ref.: S154820EPT1EIN

(57) Abstract: Techniques for providing secure processing and data storage for a wireless communication device. In one specific design, a remote terminal (110) includes a data processing unit (210, 224), a main processor (230), and a secure unit (240). The data processing unit processes data for a communication over a wireless link. The main processor provides control for the remote terminal. The secure unit includes a secure processor (250) that performs the secure processing for the remote terminal (e.g., using public-key cryptography) and a memory (254) that provides secure storage of data (e.g., electronics funds, personal data, certificates, and so on). The secure processor may include an embedded ROM (252) that stores program instructions and parameters used for the secure processing. For enhanced security, the secure processor and memory may be implemented within a single integrated circuit. Messaging and data may be exchanged with the secure unit via a single entry point provided by a bus (262).





---

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## **METHOD AND APPARATUS FOR PROVIDING SECURE PROCESSING AND DATA STORAGE FOR A WIRELESS COMMUNICATION DEVICE**

### **BACKGROUND**

#### **Field**

**[1001]** The present invention relates generally to data communication, and more specifically to techniques for providing secure processing and data storage for a wireless communication device.

#### **Background**

**[1002]** Wireless communication systems are widely deployed to provide various types of communication. These systems may be based on code division multiple access (CDMA), time division multiple access (TDMA), or some other modulation techniques. CDMA systems may provide certain advantages over other types of system, including increased system capacity.

**[1003]** Conventional wireless communication systems are typically designed to provide voice and packet data services. For these services, the data to be transmitted is processed (e.g., encoded, covered, and spread) and conditioned (e.g., amplified, filtered, and upconverted) to generate a modulated signal suitable for transmission over the wireless link. To provide a level of security for the transmission and deter eavesdropping, the data is also typically scrambled with a specific long pseudo-noise (PN) sequence assigned to the user terminal originating or receiving the transmission.

**[1004]** With the explosive growth of computer networks, such as the Internet, a user with a remote terminal is able to excess data and services from a large number of entities (e.g., websites). Via the wireless link and computer network, the remote terminal is able to retrieve and send data, purchase goods and services, and perform other transactions. For many applications, security is not necessary and data may be transmitted in the clear (i.e., without encryption). However, for certain other applications, "sensitive" data may be exchanged. Examples of such sensitive data include personal information, credit card information, account information, and so on.

## 2

For applications involving sensitive data, the scrambling with the long PN sequence only provides limited protection over the wireless portion of the transmission. This scrambling typically does not provide sufficient security for the communication.

**[1005]** For certain secure transactions, it is important to ascertain the true identities of the entities (e.g., the remote terminal) taking part in the transaction. Conventionally, a cellular system identifies a remote terminal by its mobile identification number (MIN) and electronic serial number (ESN). A shortcoming of this identification process is that the MIN/ESN is transmitted over-the-air on unsecured control channels. These channels may be easily monitored to obtain MIN/ESN information of active remote terminals. Once the MIN/ESN is known, it can be used to reprogram another remote terminal into a fraudulent clone of the original (legal) unit. Thus, the MIN/ESN is not sufficiently secure to be used for authentication of the remote terminal.

**[1006]** There is therefore a need in the art for techniques capable of supporting secure transaction for a wireless communication device.

### SUMMARY

**[1007]** Aspects of the invention provide techniques for providing secure processing and data storage for a wireless communication device. The secure processing and data storage can be achieved in various manners based on various designs and employing various cryptographic techniques. In one design, security can be achieved by designating a secure unit to perform all secure processing and to store all sensitive data.

**[1008]** A specific embodiment of the invention provides a remote terminal in a wireless communication system capable of providing secure processing and data storage. The remote terminal includes a data processing unit, a main processor, and a secure unit. The data processing unit processes data for a communication over a wireless link. The main processor provides control for the remote terminal (e.g., control of the data processing unit). The secure unit includes a secure processor that performs the secure processing for the remote terminal and a memory that provides secure storage of data (e.g., electronics funds, personal data, certificates used for authentication, and so on).

**[1009]** The secure processor can be designed to include an (embedded) read only memory (ROM) that stores program instructions and parameters used for the secure

processing. For enhanced security, the secure processor and memory can be implemented within a single integrated circuit (IC), which may also include the main processor. Messaging and data may be exchanged with the secure unit via a single entry point provided by a bus.

[1010] The secure unit can be designed to implement public-key cryptography for the secure processing. In this case, the private and public keys used for the secure processing may be generated based on various schemes and stored within the secure unit in various manners, as described below.

[1011] The secure processor may be designed with the capability to implement one or more security protocols such as, for example, the Secure Sockets Layer (SSL) protocol, Transport Layer Security (TLS) protocol, Internet Protocol Security (IPSec), and Wireless Application Protocol (WAP). For each secure transaction with a foreign entity, the secure unit can be configured to act in a role of a client or a server.

[1012] The invention further provides methods, apparatus, and elements that implement various aspects, embodiments, and features of the invention, as described in further detail below.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[1013] The features, nature, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

[1014] FIG. 1 is a block diagram of a system capable of supporting secure communication via a wireless link, in accordance with certain aspects of the invention;

[1015] FIG. 2 is a block diagram of an embodiment of a remote terminal capable of implementing various aspects of the invention;

[1016] FIG. 3 is a diagram of a specific embodiment of a secure unit within the remote terminal;

[1017] FIGS. 4A and 4B are simplified diagrams of the processing to authenticate and encrypt/decrypt a message, respectively;

[1018] FIG. 5A is a diagram of an ITU X.509 certificate that may be used for authentication;

[1019] FIG. 5B is a diagram of a specific embodiment of a memory within the secure unit;

[1020] FIGS. 6A-6C are diagrams respectively illustrating an initial loading, a successful subsequent loading, and an unsuccessful subsequent loading of a certificate into the secure unit, in accordance with embodiments of the invention; and

[1021] FIG. 7 is a diagram illustrating an SSL transaction between a client and a server.

### DETAILED DESCRIPTION

[1022] FIG. 1 is a block diagram of a system 100 capable of supporting secure communication via a wireless link, in accordance with certain aspects of the invention. In system 100, each remote terminal 110 may communicate with one or more base stations 120 over a wireless link at any particular moment, depending on whether or not the remote terminal is active and whether or not it is in soft handoff. Each base station 120 couples to and communicates with a base station controller (BSC) 130, which provides coordination and control for the base station. BSC 130 controls the routing of calls for remote terminals in communication with the base stations coupled to the BSC.

[1023] For data services, BSC 130 further couples to a packet data serving node (PDSN) 140, which performs various functions to support packet data service. PDSN 140 further couples to a network 150 (e.g., an Internet Protocol (IP) network such as the Internet) that further couples to a number of servers 160. Each server 160 may be operated to provide data and/or services.

[1024] For voice services, BSC 130 further couples a mobile switching center (MSC) 142 that further couples to a public switched telephone network (PSTN) 152. MSC 142 controls, via BSC 130 and base stations 120, the routing of telephone calls between remote terminals 110 and users coupled to PSTN 152 (e.g., conventional telephones).

[1025] The wireless portion of system 100 can be designed to support one or more CDMA standards such as the IS-95, IS-98, cdma2000, W-CDMA, or some other CDMA standard, or a combination thereof. These CDMA standards are known in the art and incorporated herein by reference.

[1026] FIG. 2 is a block diagram of an embodiment of remote terminal 110, which is capable of implementing various aspects of the invention. Remote terminal 110 includes a data processing unit used to process data for communication over the forward and reverse links with one or more base stations, a main processor used to provide control for the remote terminal, and a secure unit used to provide secure processing and data storage.

[1027] For the reverse link, data is provided (typically in blocks or packets) from a data source (e.g., a system memory 236) to a transmit (TX) data processor 210, which formats and encodes the data to provided encoded data. A modulator/transmitter unit (MOD/TMTR) 212 receives and further processes (e.g., covers, spreads, scrambles, filters, amplifies, modulates, and upconverts) the encoded data to generate a modulated signal suitable for transmission over the wireless link. The modulated signal is routed through a duplexer 214 and transmitted via an antenna 216 to one or more base stations 120. The encoding and processing at remote terminal 110 are dependent on the CDMA standard or system being implemented. The processing of the reverse link signal at a receiving base station is complementary to that performed at remote terminal 110.

[1028] For the forward link, the transmitted forward link signal from one or more base stations 120 is received by antenna 216, routed through duplexer 214, and provided to a receiver/demodulator (RCVR/DEMOM) 222. Within receiver/demodulator 222, the received signal is conditioned (e.g., amplified, filtered, downconverted, quadrature demodulated, and digitized) and further processed (e.g., descrambled, despread, and decoded) to provide symbols. A receive (RX) data processor 224 then decodes the symbols to recover the transmitted data, which is provided to a data sink (e.g., system memory 236). The processing and decoding for the forward link signal are performed complementary to the processing and coding performed at the transmitting base station.

[1029] In the embodiment shown in FIG. 2, remote terminal 110 further includes a main processor 230 acting as the central processing unit for the remote terminal. Main processor 230 performs various processing functions and further coordinates and controls the operations of various elements within remote terminal 110 to achieve the desired functionality. For example, main processor 230 typically directs the operation of TX and RX data processors 210 and 224 to process data for the reverse and forward links, respectively.

[1030] Main processor 230 further couples to a bus 232 that interconnects a number of other elements such as input/output (I/O) interfaces 234, system memory 236, and a secure unit 240. I/O interfaces 234 provide interface with the user and may comprise a keypad, a display unit, a speaker, a microphone, and possibly others. System memory 236 may comprise a random access memory (RAM) and a read only memory (ROM) used to store program instructions (e.g., for main processor 230) and data. Secure unit 240 performs secure processing and provides secure storage, as described in further detail below.

[1031] Main processor 230 can be designed to operate based on program instructions downloaded onto system memory 236 (e.g., onto a Flash memory that is part of memory 236). The download may be achieved via external I/O lines or over-the-air transmissions. Because of its easy accessibility, main processor 230 is vulnerable to attack from the external I/O lines as well as from over-the-air negotiation.

[1032] Secure processing and data storage can be achieved in various manners based on various designs and employing various cryptographic techniques. In one design, security can be achieved by designating secure unit 240 to perform all secure processing and to store all "sensitive" data. In general, sensitive data includes any data desired to be prevented from unauthorized access. In another design, security can be achieved by designating secure unit 240 to perform all secure processing (e.g., based on cryptographic keys stored within the secure unit) but the sensitive data can be made secure and stored outside secure unit 240 (e.g., in system memory 236). Some of these designs are described below, and others are also possible and within the scope of the invention.

[1033] FIG. 3 is a diagram of a specific embodiment of secure unit 240. In this embodiment, secure unit 240 effectively implements a secure digital "vault" that employs a secure processor 250 to access a non-volatile memory 254, which is isolated from other untrusted units (e.g., main processor 230). In an embodiment, to provide enhanced security, secure unit 240 interfaces with other elements within remote terminal 110 (e.g., main processor 230, system memory 236) via a single entry point provided by a bus 262 that couples directly to secure processor 250. This design ensures that all communication and data exchanges with secure unit 240 are channeled to a single trusted processor 250, which can be designated and designed to safeguard

against security attacks and spoofing entities (e.g., hackers, viruses, and so on) that may attempt to infiltrate the secure unit in order to retrieve secure data.

**[1034]** Secure processor 250 is a trusted processing unit that performs secure processing for remote terminal 110. The secure processing may be achieved based on program instructions and parameter values (e.g., cryptographic keys) stored in a ROM 252. Secure processor 250 receives external messages and data via bus 262, authenticates and/or processes the received messages and data, and may store the data to memory 254. When required and as instructed, secure processor 250 retrieves data stored within memory 254, processes and/or encrypts the retrieved data, and may send the data to external elements (e.g., main processor 230) via bus 262.

**[1035]** Memory 254 is a non-volatile memory that may be used to store sensitive data and (possibly) program instructions. Because of its placement behind secure processor 250, memory 254 is physically separated from other unsecured elements, which do not have direct access to memory 254. Memory 254 may be battery backed, and may be implemented as a Flash memory.

**[1036]** In the embodiment shown in FIG. 3, ROM 252 is implemented within secure processor 250 and stores program instructions and secure parameters used to perform the secure processing. This design allows secure processor 250 to be operated without dependency on other external elements, since such dependency may compromise security. The program instructions and parameters may be loaded (or burned) into the ROM 252 via a secure operation (e.g., during the manufacturing phase) and become available for use thereafter.

**[1037]** Various mechanisms may be used to deter against unauthorized access of memory 254 (i.e., without first going through secure processor 250). In one embodiment, secure processor 250 and memory 254 are implemented within a single integrated circuit (IC). This allows memory 254 to be physically secure with secure processor 250 and prevents tampering of memory 254. The IC may or may not include other elements of remote terminal 110 (e.g., main processor 230). In another embodiment, secure processor 250 and memory 254 are implemented as two separate units enclosed within a secure and/or tamper resistance/evident unit. Other mechanisms to prevent and deter unauthorized access of memory 254 may also be implemented and are within the scope of the invention.



[1038] Secure unit 240 may be designed to implement a number of secure functions for remote terminal 110, which may in turn be used for various applications. These secure functions may include any combination of the following: authentication, encryption, data storage/manipulation, and possibly others. Authentication comprises the processing needed to verify the true identity of an entity, and is used to allow remote terminal 110 to verify the identity of a foreign entity (e.g., server 160) or to allow the foreign entity to verify the remote terminal's identity. Encryption comprises the processing to secure data such that unauthorized entities are not able to intercept and recover the data. Secure data storage/manipulation entails safeguarding sensitive data against unauthorized accesses, and updating the data only when and as appropriate. These secure functions are described in further detail below.

[1039] Various schemes may be used to implement authentication and/or encryption. One popular scheme is based on public-key cryptography, which uses a pair of keys - a private key and a public key. The private key is kept in secrecy and the public key is provided as needed (e.g., for authentication, encryption, or decryption). Secret keys may also be generated for a particular secure transaction based on the private key. The generation and management of keys are described in further detail below. Different secure functions (i.e., authentication or encryption, or both) may be achieved based on how the keys are used to process data.

[1040] Other schemes may also be used for authentication and/or encryption and are within the scope of the invention. For example, secret key cryptography based on DES (Data Encryption Standard) may also be used. For secret key cryptography (which is also referred to as symmetric encryption), both transacting entities know a secret key *a priori* and keep the key secret from others.

[1041] FIG. 4A is a simplified diagram of the processing to authenticate a message. Authentication may be used by either remote terminal 110 or a foreign entity (e.g., server 160), or both, to verify that the source originating the message is what it claims to be. At the sending entity (A), a message (M) to be transmitted is hashed by a hash function (block 414) to provide a message digest (D). The hash function may be the SHA-1 (Secure Hash algorithm), MD-4 (Message Digest), MD-5, or some other hash algorithm known in the art.

[1042] The message digest is then either encrypted or signed (block 416) with the sending entity's private key to generate a signature (S). The encryption can be based on

the RSA (Rivest, Shamir, and Adleman), Diffie-Hellman, DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), CAVE (Cellular Authentication and Voice Encryption, which is defined by IS-54), or some other encryption algorithm known in the art. The "signing" can be based on the DSA (Digital Signature Algorithm) defined in the DSS (Digital Signature Standard), or some other algorithm. The private key is kept secret and is only known to the sending entity. The message and signature are both transmitted to the receiving entity (B).

[1043] At the receiving entity, the transmitted message and signature are received, and the received message (M') is hashed (block 424) with the same hash function used at the sending entity to generate a recovered message digest (D'). The received signature (S') is also decrypted or processed/verified (block 426) with the sending entity's public key to generate a value. Depending on the algorithm used, the decryption/verification can be based on an algorithm that is the same or complementary to the one used at the sending entity. The generated value is compared to the recovered message digest (block 428), and the sending entity is authenticated if the two match.

[1044] FIG. 4B is a simplified diagram of the processing to encrypt a message. Encryption may be used by either remote terminal 110 or a foreign entity, or both, to secure data prior to transmission. At the sending entity (A), a message (M) to be transmitted is encrypted (block 434) with the receiving entity's public key (or a secret key) to generate an encrypted message that only the receiving entity can recover. The encryption can be based on the RSA, Diffie-Hellman, DES, IDEA, or some other encryption algorithm. The encrypted message is transmitted to the receiving entity (B). The secret key may be generated for the communication (or transaction) based on the Diffie-Hellman or RSA algorithm.

[1045] At the receiving entity, the transmitted message is received and decrypted (block 444) with an algorithm that is the same or complementary to the one used at the sending entity. The decryption is performed with the receiving entity's private key (or the complementary secret key). Thus, only the receiving entity is able to recover the encrypted message since only it has the private key (or secret key) corresponding to the public key used to encrypt the message.

[1046] Based on the above descriptions for authentication and encryption, the generation and management of keys are important aspects of a secure system. For a public-key system, a private key and a public key are needed for the secure processing.

These keys can be generated and provided to the remote terminal in a secure manner based on various schemes.

**[1047]** In one key management scheme, the private and public keys are generated for the remote terminal (e.g., by a certificate authority), and the private key is permanently stored in the remote terminal. The permanent storage may be achieved, for example, by etching the private key in a metal layer of secure processor 250 during the manufacturing process. This effectively "stamps" each remote terminal with its own permanent private key, which ensures that the private key is safe from theft, attack, and alteration.

**[1048]** In another scheme, the private and public keys are generated for the remote terminal, and the private key is loaded into the secure storage (e.g., ROM 252 or memory 254) within secure processor 250. The loading of the private key can be achieved in a secure environment, e.g., during the manufacturing process or at a

subsequent time if it is detected that security may have been compromised. For example, the private key may be stored to ROM 252 by blowing fuses (e.g., electronically or with a laser). This scheme may provide flexibility in updating the keys. Since the secure storage is inaccessible by external elements, the private key is safe.

**[1049]** In yet another scheme, the private and public keys are generated by secure processor 250, if requested or as directed. The keys may be generated entirely by the remote terminal, or may be generated based on parameters provided by an external source (e.g., a certificate authority). This scheme provides flexibility in updating the keys. The private key can be stored into the secure storage (e.g., ROM 252 or memory 254).

**[1050]** For the above schemes, the public key is typically also stored in the secure memory (e.g., stored in memory 254). The public key may thereafter be provided to other entities whenever needed. The public key may be certified by a trusted certificate authority and encapsulated in a certificate, which may be stored to the secure storage, as described below.

**[1051]** With public-key cryptography, entities can negotiate the keys used for performing the secure processing. In many instances (e.g., e-commerce) it is not practical to know the key beforehand. With public-key cryptography, the two transacting entities can use different private keys and exchange public keys or secret

keys, as needed. Secret keys (e.g., for a particular transaction) may also be generated based on the Diffie-Hellman or RSA algorithm and exchanged as needed.

**[1052]** A public key may be encapsulated within a certificate, which may then be sent and used for authentication and/or encryption. Initially, the remote terminal is provided with a private key (e.g., based on any one of the techniques described above). A certificate, such as the one defined by ITU X.509, can be issued for the remote terminal. The certificate includes various types of information such as the remote terminal's public key, a signature, and the specific algorithm and parameters used to generate the signature. The certificate can be stored by the remote terminal and later used for authentication, which may be achieved based on various schemes.

**[1053]** In one authentication scheme, the certificate is issued and signed by a trusted certificate authority that certifies the remote terminal's identity. Authentication of the remote terminal may thereafter be achieved as follows. The remote terminal sends to a foreign entity (e.g., server 160) a message signed by the remote terminal along with the certificate, which includes the remote terminal's public key and a signature of the certificate authority. The foreign entity receives the remote terminal certificate, authenticates the signature of the certificate authority, and uses the remote terminal's public key to authenticate the signed message. The foreign entity is thus able to verify the identity of the remote terminal, as certified by the trusted certificate authority.

**[1054]** In another authentication scheme, the certificate is generated and signed by the remote terminal. Authentication of the remote terminal may thereafter be achieved as follows. The remote terminal sends to a foreign entity a message signed by the remote terminal along with the certificate, which includes the remote terminal's public key and the remote terminal's signature. The foreign entity receives the remote terminal certificate, authenticates the signature of the remote terminal, and uses the remote terminal's public key to authenticate the signed message. The foreign entity thus verifies the identity of the remote terminal as based on the remote terminal's signature.

**[1055]** Other schemes may also be used for authentication and are within the scope of the invention. Different levels of authentication may be achieved, for example, based on different uses of the certificate. The particular authentication scheme used for a secure transaction may be dependent on the type of transaction being performed.

**[1056]** FIG. 5A is a diagram of an ITU X.509 certificate 510 that may be used to encapsulate a public key. Certificate 510 includes a number of fields used to provide

various types of information related to the key. A version field 512 identifies the format of the certificate (e.g., X.509 version 3). A certificate serial number field 514 includes the specific serial number assigned to this certificate (by the issuer of the certificate, e.g., a certificate authority). A signature algorithm identification field 516 identifies the specific algorithm used by the issuer of the certificate to sign the certificate (e.g., MD-5 hash, RSA signature, or some other). This allows an entity receiving the certificate to process and authenticate the certificate. An issuer name field 518 identifies the specific trusted certificate authority (e.g., Verisign, Belsign, American Express, and so on), if any, which issues the certificate.

**[1057]** A validity period field 520 identifies the time period over which the certificate is valid. This period is typically determined by the issuer. A subject name field 522 includes the name of the entity (the "subject") for which the certificate was generated. A subject public key field 524 includes the public key of the subject (e.g., RSA, 0xabcd, 0x12345). An issuer unique ID field 526 and a subject unique ID field 528 include the IDs assigned to the issuer and subject, respectively. Extensions field 530 may be used to include keys, policy information, attributes, constraints, and other pertinent information. And a signature field 532 includes a signature generated by hashing fields 512 through 530 and encrypting/signing the hash digest with the issuer's private key.

**[1058]** FIG. 5B is a diagram of a specific embodiment of memory 254. The implementation of memory 254 is typically dependent on the overall design of secure unit 240. Moreover, the types of data to be stored in memory 254 may be dependent on the scheme used for the secure processing. In the embodiment shown in FIG. 5B, memory 254 includes a flags field 552, a certificate field 554, and a number of data fields 556a through 556n. Additional and/or different fields may also be supported and are within the scope of the invention.

**[1059]** Flags field 552 includes one or more flags indicative of the state of memory 254 and/or the state of the stored data. The flags within field 552 allow secure unit 240 to keep track of approved memory accesses, parameter and data updates, alarms, and so on, as described below. For example, a flag may be provided to indicate whether or not a certificate is stored within memory 254. Certificate field 554 stores one or more certificates, which may be used to authenticate remote terminal 110 and/or other entities. Certificates are typically loaded into memory 254 via secure transactions (as

described below), and typically include parameters (e.g., cryptographic keys) used to perform secure processing (as described above). Data fields 556 store sensitive data and possibly data needed for the operation of secure processor 250.

**[1060]** Various types of sensitive data may be stored within secure unit 240. Such sensitive data may comprise, for example, personal information, financial information (e.g., credit card number, electronic funds balance, account information, and so on), authentication information, and others. Some of these data types are described below. Generally, any data desired to be prevented from unauthorized accesses may be deemed as sensitive data and stored within secure unit 240.

**[1061] Certificates.** As a remote terminal (e.g., a cellular phone) becomes more of an e-commerce device, the need for it to act as a "server" inevitably arises. For example, before being entrusted with sensitive data or before a transaction can be initiated, the true identity of the remote terminal may need to be ascertained. In that case, the remote terminal is authenticated by and to the satisfaction of the foreign entity. This authentication may be achieved based on a certificate containing identity verification information for the remote terminal. If the remote terminal certificate is distributed from a trusted certificate authority and can be verified as such, then the remote terminal's identity can be authenticated, as verified by the certificate authority. The vault may be used to store one or more remote terminal certificates. For example, certificates for all members of a family or a team may be stored, with each member having individual "personal account" information stored with different pins to access.

**[1062] Electronic Wallet.** The remote terminal may serve as a "wallet" and store electronic currency in the secure digital vault (e.g., within memory 254). For example, a user may communicate with a bank and download funds into the vault. The funds may thereafter be used to purchase goods and services from stores or websites, to pay bills, or may be transferred to another device or entity. The user may also replenish the vault with additional funds as desired. For each transaction, the proper amount is deducted from or credited to the current balance. The transaction may be achieved via a wireless connection (e.g., via a bluetooth connection with a properly equipped cash register, over-the-air to a website, and so on).

**[1063] Cryptographic Information.** The remote terminal may store cryptographic parameters and keys used for the secure processing. These parameters may include, for example, those used for the CAVE algorithm defined by IS-54 to authenticate the

remote terminal. The remote terminal may also be designed to store session keys used to support secure sessions with websites. The session keys may be provided at the start of a session and may be discarded at the end of the session. The remote terminal may further store cryptographic keys used for signing and verifying messages, for encrypting and decrypting data, and so on.

**[1064]** FIG. 6A is a diagram illustrating an initial loading of a certificate into secure unit 240, in accordance with an embodiment of the invention. In an embodiment, the level of security to be exercised for a certificate load is dependent on the status of the secure storage (e.g., memory 254). If the secure storage is empty (e.g., as indicated by a flag) a certificate may be loaded with reduced security checks. Otherwise, if the secure storage already contains a certificate, a more complex transaction involving more secure checks may be performed. The process shown in FIG. 6A may be used to load the private key for the remote terminal, if one is not already embedded in the secure processor. The process may also be used to load the primary and secondary users' certificates into memory 254.

**[1065]** The certificate includes the remote terminal's public key. Depending on the particular scheme used, the certificate authority may (1) generate the remote terminal's private and public keys and provide them to the remote terminal, or (2) be provided with the public key generated by the remote terminal. The public key is then encapsulated into a certificate such as the one shown in FIG. 5A. If the encapsulation is performed by the certificate authority, the certificate authority's signature is included in the certificate and attests to the verification of the remote terminal and the validity of the public key by the certificate authority.

**[1066]** The initial certificate loading may be performed during the manufacturing process or at a subsequent time. As shown in FIG. 6A, the certificate is loaded via a transaction between a trusted certificate authority 600 and secure processor 250, via main processor 230. Initially, certificate authority 600 sends a message 612 requesting a certificate load. Main processor 230 receives and processes the message and, in response, sends a request 614 to check the status of the secure storage (e.g., memory 254). Secure processor 250 receives the request and determines the status of the secure storage, e.g., by checking a particular flag that indicates whether the certificate field in the memory is full or empty. If the secure storage is empty (i.e., does not contain a certificate), secure processor 250 sends a message 616 indicating this status. Main

processor 230 receives message 616 and reports the status of the secure storage to certificate authority 600 via a message 618.

**[1067]** In response to message 618, certificate authority 600 sends a certificate via a load certificate message 620. Main processor 230 then receives and forwards message 620 to secure processor 250, which loads the certificate (e.g., included in the message) to the secure storage and further sets the flag to full. Secure processor 250 then sends an acknowledgment message 624, which is received by main processor 230 and forwarded to certificate authority 600.

**[1068]** FIG. 6B is a diagram illustrating a subsequent loading of a certificate into secure unit 240, in accordance with an embodiment of the invention. A certificate may need to be updated via a subsequent certificate loading if it is determined that the user information has changed, the keys have been compromised, or for other reasons. Initially, certificate authority 600 sends a message 612 requesting a certificate load. Main processor 230 receives and processes the message and, in response, sends a request 614 to check the status of the secure storage. Secure processor 250 receives the request and determines the status of the secure storage. If the secure storage is full (i.e., already contains a certificate), secure processor 250 sends a message 636 indicating this status. Main processor 230 receives message 636 and sends to certificate authority 600 a first message 638 indicating that a certificate is already present in memory 254 and a second message 640 requesting authentication of certificate authority 600.

**[1069]** In response to message 640, certificate authority 600 sends a signed message 642. Main processor 230 receives and processes message 642 and sends a message 644 requesting secure processor 250 to authenticate the signed message. Secure processor 250 verifies the signed message (e.g., using the public key of certificate authority 600) and, if authenticated, sets the status of the secure storage to empty and sends an acknowledgment message 646 indicating that the authentication passed. Main processor 230 receives and forwards the acknowledgment to certificate authority 600.

**[1070]** In response to message 646, certificate authority 600 sends a certificate via a load certificate message 620, which is received by main processor 230 and forwarded to secure processor 250. Secure processor 250 loads the certificate to memory 254, sets the flag to full, and sends an acknowledgment message 624 back to certificate authority 600.



[1071] FIG. 6C is a diagram illustrating an unsuccessful attempt to load a certificate into secure unit 240. Initially, certificate authority 600 sends a certificate load request message 612. Main processor 230 receives and processes the message and, in response, sends a request 614 to check the status of the secure storage. Secure processor 250 receives the request, determines the status of the secure storage (which is full in this example), and sends a message 636 indicating this status. Since the secure storage is full, main processor 230 sends a first message 638 indicating that a certificate already exists in the secure storage and a second message 640 requesting authentication of certificate authority 600.

[1072] In response to message 640, certificate authority 600 sends a signed message 642. Main processor 230 receives and processes message 642 and sends a message 644 requesting secure processor 250 to authenticate the signed message. Secure processor 250 verifies the signed message and, if not authenticated, sends an error message 656 indicating that the authentication failed. Main processor 230 receives and forwards the error message to certificate authority 600. The error message terminates the transaction.

[1073] Secure processor 250 can be designed to perform various functions in support of the secure processing and data storage. These functions may include any combination of the following: signature generation and verification, encryption and decryption, database management, approval of updates of secure data in the secure storage, accounting, error message handling, and possibly others.

[1074] In an embodiment, the program instructions for some or all of the supported functions are stored within secure unit 240 (e.g., within ROM 252 or memory 254). This allows secure processor 250 to execute the functions based on instructions known to be reliable. This also prevents external elements (e.g., main processor 230) from spoofing secure processor 250 and maliciously accessing the secure storage. The securely stored program instructions may include those that implement hash functions, encryption, decryption, and signature algorithms, accounting functions, data management functions, and so on.

[1075] For authentication, secure processor 250 may be queried to generate and sign a message using the private key and provide the signed message to main processor 230, which then transmits the signed message to the intended recipient entity. The signature generation can be performed based on any one of the digital signature and encryption algorithms listed above. Secure processor 250 may further provide the certificate that

includes the remote terminal's public key, which can be used by the recipient entity to authenticate the remote terminal.

**[1076]** Secure processor 250 may also be designed to authenticate each entity that requests to retrieve, load, or update the data stored in the secure storage. In general, secure processor 250 authenticates each entity desiring access to the data stored within the secure storage. The authentication may be achieved by verifying a signed message from the requesting entity (e.g., based on either the requesting entity's certificate or the requesting entity's public key, which may be included in a certificate already be stored within the secure storage). The signature verification can be performed based on the same or complementary algorithm used to generate the signature.

**[1077]** A message is used to generate a signature, and the signature is authenticated by a receiving entity (e.g., the remote terminal or the foreign entity). Thus, the data in the message is also verified if the signature checks. However, the message is transmitted in the clear, and no protection is provided against eavesdroppers. For many applications, it may only be important to ascertain the true identity of an entity, and authentication via digital signature is sufficient.

**[1078]** Encryption may be used to protect sensitive data from eavesdroppers. For encryption, secure processor 250 may be queried to encrypt data using a secret key. The encrypted data can then be provided to main processor 230, which then transmit the encrypted data to the intended recipient entity. The secret key may be generated based on the recipient entity's private key using, for example, the Diffie-Hellman or RSA algorithm. Secure processor 250 may also be queried to decrypt encrypted data using the remote terminal's private key. The encryption and decryption of traffic may be performed based on any one of the encryption algorithms listed above (e.g., DES, IDEA, and so on). The key exchange and traffic encryption/decryption algorithms are independent of using symmetric key encryption.

**[1079]** For each secure transaction with secure processor 250, the foreign entity can be authenticated as described above. Once the foreign entity has been authenticated, secure processor 250 can process the received message. Depending on the transaction, data may be extracted from the received message and stored to the secure storage, or retrieved from the secure storage and provided via a signed or encrypted message. The transaction may alternatively request an update of the data within the secure storage.

For example, funds stored within the secure storage may be deducted for purchases or some other transactions, or may be increased for replenishments.

**[1080]** Sensitive data may be securely stored and/or updated based on various schemes. In one scheme, sensitive data is stored within secure unit 240 (e.g., in memory 254 or possibly in ROM 252). With this scheme, data is verified when received, before being stored to the secure storage. For this scheme, the data may be stored in plain form (i.e., unencrypted). Subsequent data manipulations and updates of the data are performed within secure unit 240. Secure unit 240 is assured of the integrity of the data since it is stored within, and is under control of the secure unit at all times.

**[1081]** In another scheme, sensitive data is stored outside secure unit 240 (e.g., in system memory 236) in secure form. Again, secure unit 240 verifies the data when received from external elements (e.g., main processor 230). Prior to storage of the data, secure unit 240 may sign or encrypt the data, depending on the desired implementation, using the private key stored within the secure unit. The secured data may then be stored outside secure unit 240. For subsequent accesses, manipulations, and/or updates, the secured data may be retrieved from the external storage, verified or decrypted, and processed by secure unit 240. Secure unit 240 is assured of the integrity of the data since it is secure prior to storage and verified prior to each use.

**[1082]** Other schemes to store and manipulate data to ensure integrity may also be used and are within the scope of the invention.

**[1083]** Secure processor 250 may be designed to generate error messages for incomplete transactions (e.g., if authentication fails) or alarms in response to unauthorized attempts to access the secure data. The error messages may indicate the failure level (e.g., a warning or a fatal error), the cause for the error (e.g., unexpected message received, bad record hash, signature does not check, compression failure, handshake failure, illegal parameters, certificate errors, insufficient funds for the transaction, unauthorized accessing entity, and so on), and possibly other information.

**[1084]** Secure processor 250 may also be designed to support any number of security protocols such as the Secure Sockets Layer (SSL) protocol, the Transport Layer Security (TLS) protocol, and others. These protocols are known in the art and not described herein. Each protocol may establish a handshake protocol used to establish a

secure communication and a messaging protocol used to establish security capabilities, exchange keys and certificates, and send secure data.

**[1085]** For each secure transaction, the remote terminal may act in the capacity of a server or a client. As a server, the remote terminal is requested by another entity to provide secure data. Before providing the requested data, the remote terminal typically authenticates the requesting entity. As a client, the remote terminal requests secure data from another entity and may be requested to provide information needed by the other entity to authenticate the remote terminal.

**[1086]** In a typical SSL transaction, the server does not authenticate the client via crypto protocols. However, the server may authenticate via other means such as credit card authorization. In an embodiment, the server authenticates the client for each secure transaction (e.g., cash register tries to take cash out of an e-wallet implemented by the secure processor). In a typical SSL transaction, the client always authenticates the server. In an embodiment, the server may authenticate the client for a secure transaction (e.g., if retrieving medical records, etc).

**[1087]** FIG. 7 is a diagram illustrating a secure transaction between a client and a server. The remote terminal may be acting in either capacity for this transaction, which is partitioned into four phases. In the first phase, hello messages 712a and 712b are exchanged between the client and server to establish the secure communication. In the second phase, the server sends the server certificate, exchanges server public keys, and requests for the client certificate via messages 722, 724, and 726, respectively. The server concludes with a server hello done message 728.

**[1088]** In response to the server messages, the client sends the client certificate, exchanges client public keys, and requests verification of the certificate via messages 732, 734, and 736, respectively. And in the fourth phase, if the certificates have been authenticated, the client and server enable cipher suite and finishes the handshake via messages 742 and 744. Enabling cipher suite simply means to 'enable' encryption from that point forward in time. Each entity knows that from then on, the received stream will be encrypted. Data can thereafter be securely exchanged between the client and server via the exchanged keys.

**[1089]** Secure processor 250 and main processor 230 may each be implemented with a digital signal processor (DSP), an application specific integrated circuit (ASIC), a processor, a microprocessor, a controller, a microcontroller, a field programmable gate

array (FPGA), a programmable logic device, other electronic unit, or any combination thereof designed to perform the functions described herein. The integrated circuit that implements secure processor 250 may further include other elements of remote terminal 110 such as, for example, main processor 230, TX and RX data processors 210 and 224, and so on.

**[1090]** Non-volatile memories (e.g., memory 254 and ROM 252) may be implemented with be a Flash memory, a programmable ROM (PROM), an erasable PROM (EPROM), an electronically erasable PROM (EEPROM), a battery backed-up RAM, some other memory technologies, or a combination thereof. Volatile memories (e.g., part of memory 236) may be implemented with a random access memory (RAM), a Flash memory, some other memory technologies, or a combination thereof.

**[1091]** The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

**[1092] WHAT IS CLAIMED IS:**

**CLAIMS**

1. A remote terminal in a wireless communication system, comprising:  
2 a data processing unit configured to process data for a communication over a wireless link;  
4 a main processor coupled to the data processing unit and configured to provide control for the remote terminal, wherein the data processing unit and main processor are  
6 unsecured units vulnerable to being spoofed by external entities; and  
a secure unit operatively coupled to the main processor and including  
8 a secure processor configured to perform secure processing for the remote terminal, and  
10 a secure memory configured to provide secure storage of data, and  
wherein the secure unit is physically encapsulated within a secure module and  
12 further configured to prevent unauthorized accesses to the secure memory via hardcoded protocols.

2. The remote terminal of claim 1, wherein the secure unit further includes  
2 a read only memory (ROM) configured to store program instructions and parameters used for the secure processing.

3. The remote terminal of claim 2, wherein the ROM is embedded within  
2 the secure processor.

4. The remote terminal of claim 1, wherein the secure processor and secure  
2 memory are implemented and physically encapsulated within a single integrated circuit (IC).

5. The remote terminal of claim 1, wherein the secure processor and secure  
2 memory are physically encapsulated within a tamper resistance or tamper evident unit.

6. The remote terminal of claim 1, wherein the secure processor and secure  
2 memory are permanently affixed within the remote terminal.

7. The remote terminal of claim 1, wherein messaging and data are  
2 exchanged with the secure unit via a single entry point provided by a bus.

8. The remote terminal of claim 1, wherein the secure unit is configured to  
2 implement public-key cryptography for the secure processing.

9. The remote terminal of claim 8, wherein a private key assigned to the  
2 remote terminal is embedded within the secure processor.

10. The remote terminal of claim 9, wherein the private key is permanently  
2 etched within the secure processor.

11. The remote terminal of claim 9, wherein the private key assigned to the  
2 remote terminal is stored in a ROM within the secure processor.

12. The remote terminal of claim 1, wherein the secure processor is  
2 configurable to implement one or more security protocols.

13. The remote terminal of claim 12, wherein the one or more security  
2 protocols include Secure Sockets Layer (SSL) protocol or Transport Layer Security (TLS) protocol, or both.

14. The remote terminal of claim 1, wherein the secure unit is configurable  
2 to act in a role of a client or a server for each secure transaction with a foreign entity.

15. The remote terminal of claim 1, wherein the secure memory is  
2 configured to store electronics funds.

16. The remote terminal of claim 1, wherein the secure memory is  
2 configured to store cryptographic parameters used for the secure processing.

23

17. The remote terminal of claim 1, wherein the secure memory is  
2 configured to store one or more certificates used for authentication.

18. The remote terminal of claim 17, wherein a certificate is loaded into the  
2 secure memory via a secure transaction with a certificate authority.

19. The remote terminal of claim 18, wherein different levels of security is  
2 implemented for a certificate loading transaction depending on whether or not a  
certificate has already been loaded to the remote terminal.

20. A remote terminal in a wireless communication system, comprising:  
2 a data processing unit configured to process data for a communication over a  
wireless link;  
4 a main processor coupled to the data processing unit and configured to provide  
control for the remote terminal, wherein the data processing unit and main processor are  
6 unsecured units vulnerable to being spoofed by external entities; and  
a secure unit embedded within the main processor and configured to perform  
8 secure processing for the remote terminal and provide secure storage of data, wherein  
the secure unit is further configured to implement public-key cryptography for the  
10 secure processing, and wherein the secure unit is further configured to prevent  
unauthorized accesses to securely stored data via hardcoded protocols.

21. A method for providing secure processing and data storage for a wireless  
2 communication device, comprising:  
defining a secure processor within the communication device for performing  
4 secure processing;  
defining a secure storage within the communication device for providing secure  
6 data storage;  
storing program instructions and parameters used for the secure processing  
8 within the secure processor or secure storage, wherein the stored program instructions  
implement hardcoded protocols; and  
10 physically encapsulating the secure processor and secure storage within a secure  
unit.



22. The method of claim 21, wherein the secure processor and secure storage  
2 are physically encapsulated within a single integrated circuit (IC).

23. The method of claim 21, further comprising:  
2 permanently affixing the encapsulated secure processor and secure storage  
within the communication device.

24. A method for providing secure processing and data storage for a wireless  
2 communication device, comprising:  
receiving a first message to initiate a secure transaction with a foreign entity;  
4 authenticating the foreign entity through a secure processor located within the  
communication device; and  
6 if the foreign entity is authenticated, performing securing processing for the  
secure transaction through the secure processor, and  
8 wherein the secure unit is physically encapsulated within a secure module and  
further configured to prevents unauthorized accesses to the secure memory via  
10 hardcoded protocols.

25. The method of claim 24, wherein the secure processing is performed  
2 based on program instructions stored within the secure processor.

26. The method of claim 24, wherein the authentication is achieved via  
2 exchanges of certificates.

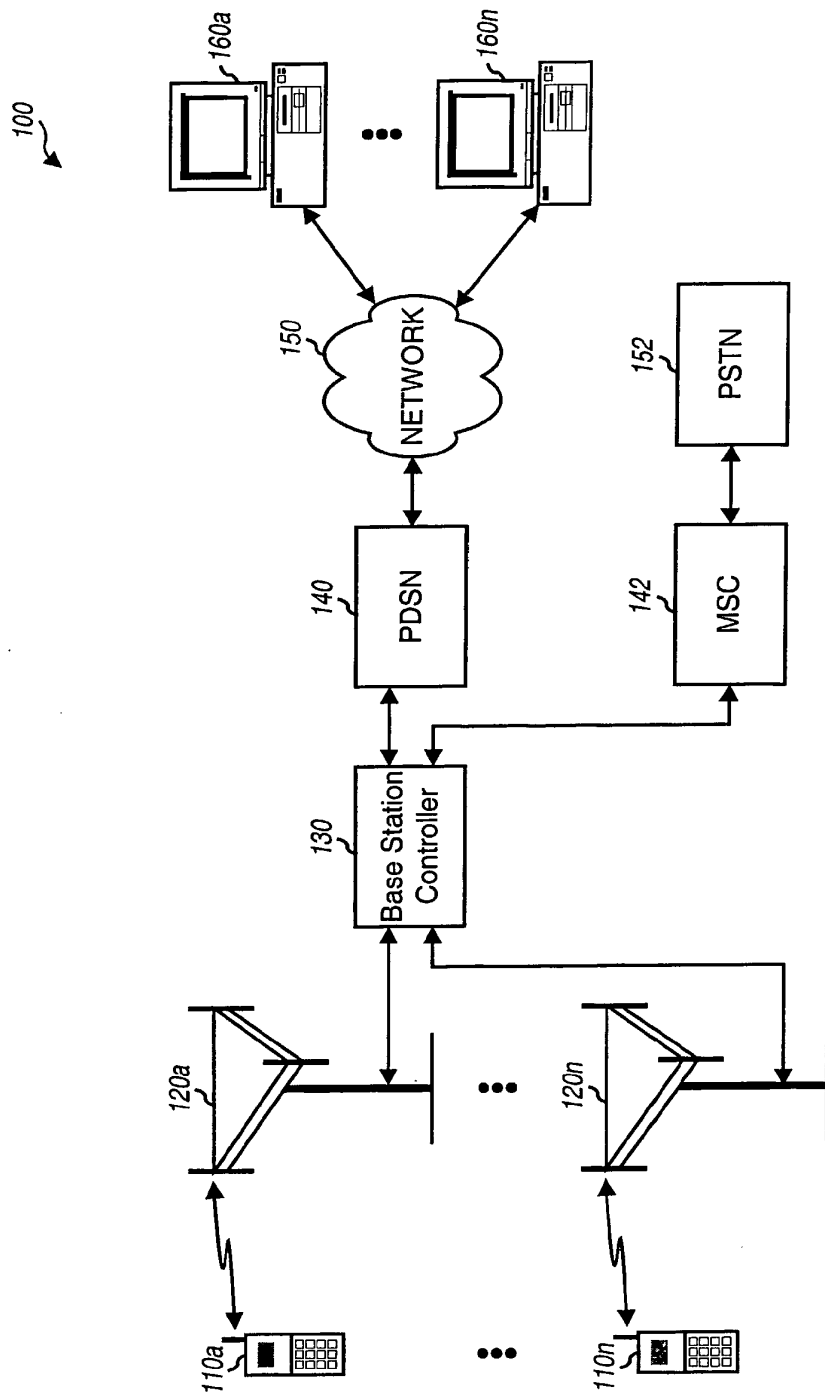


FIG. 1

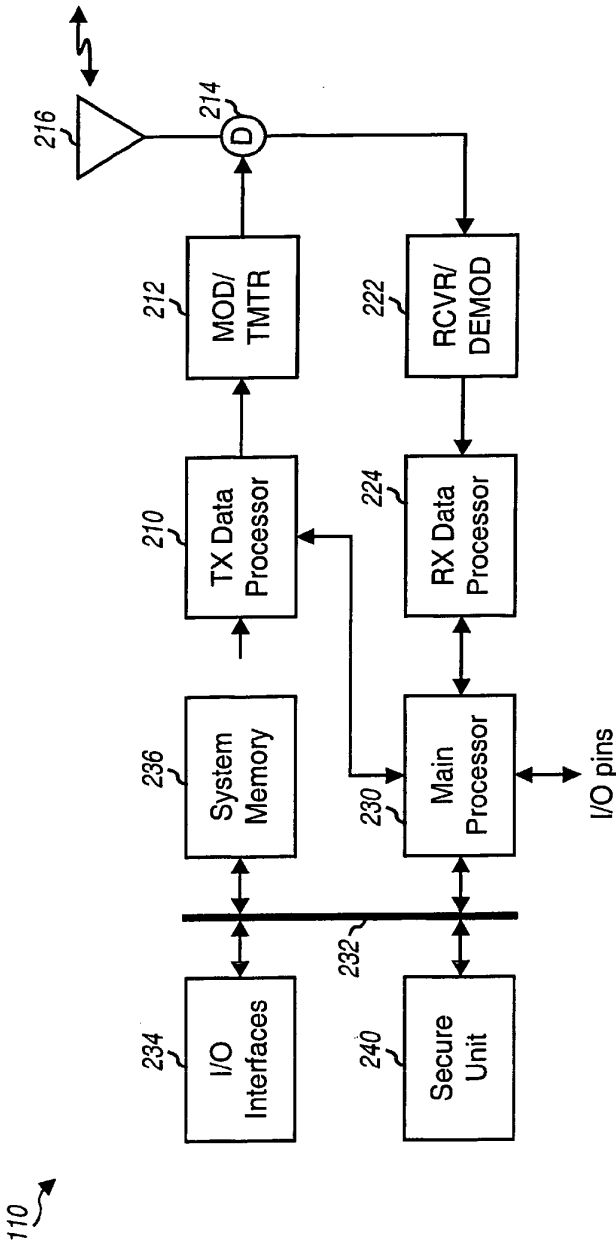


FIG. 2

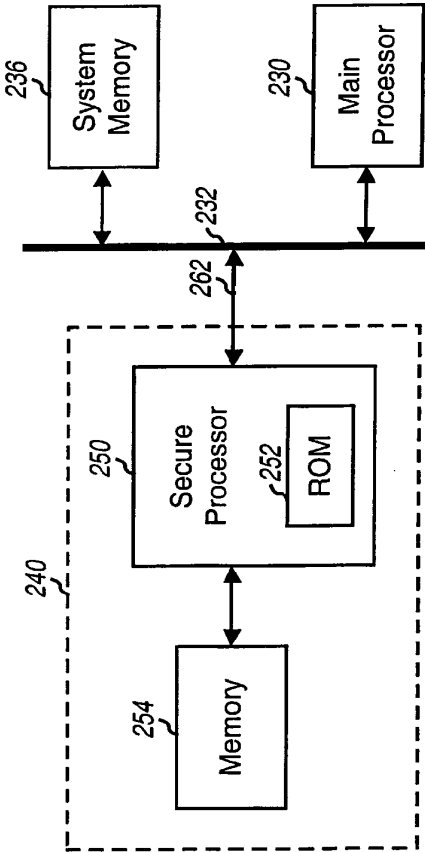


FIG. 3

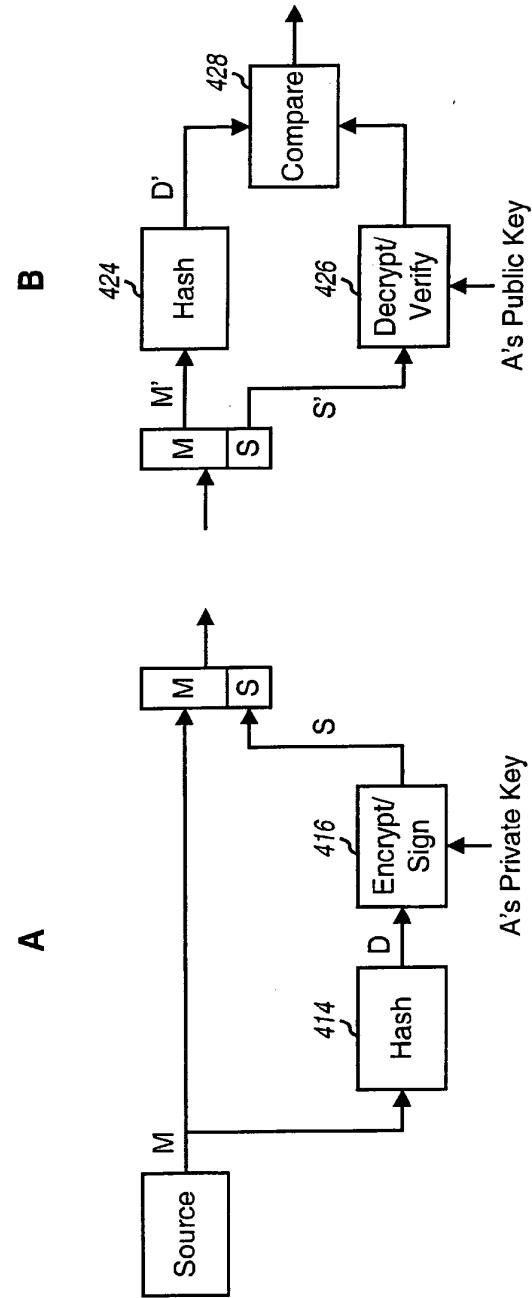


FIG. 4A

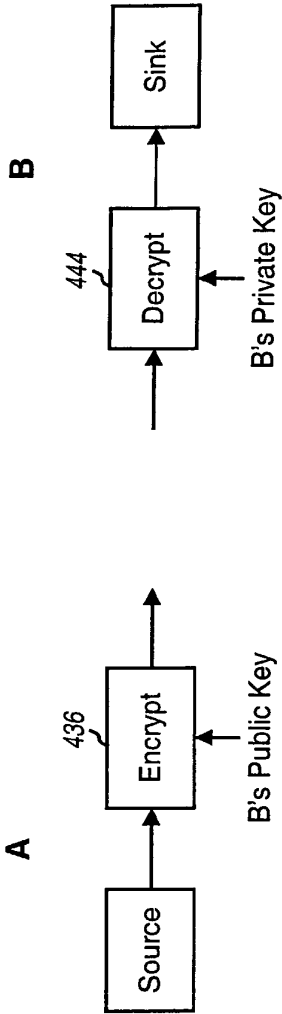


FIG. 4B

5/9

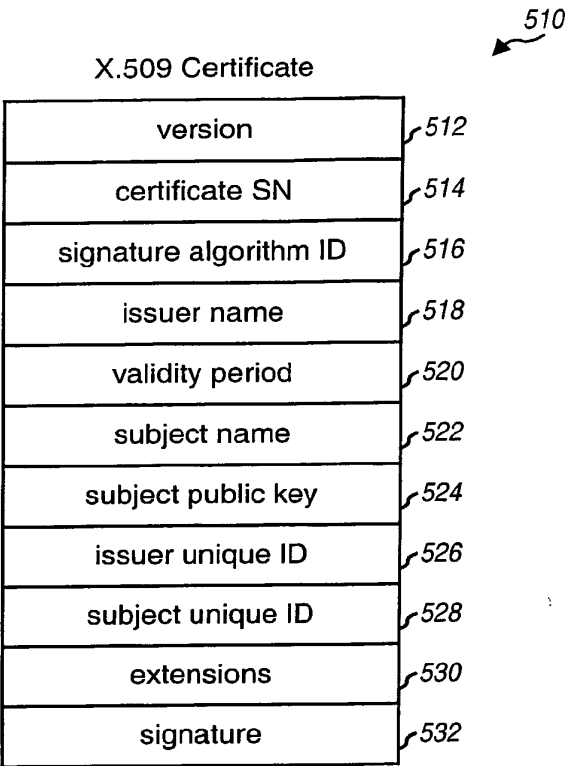


FIG. 5A

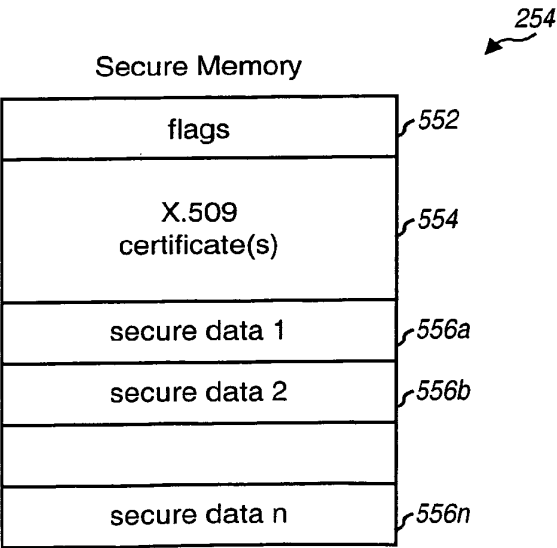


FIG. 5B

6/9

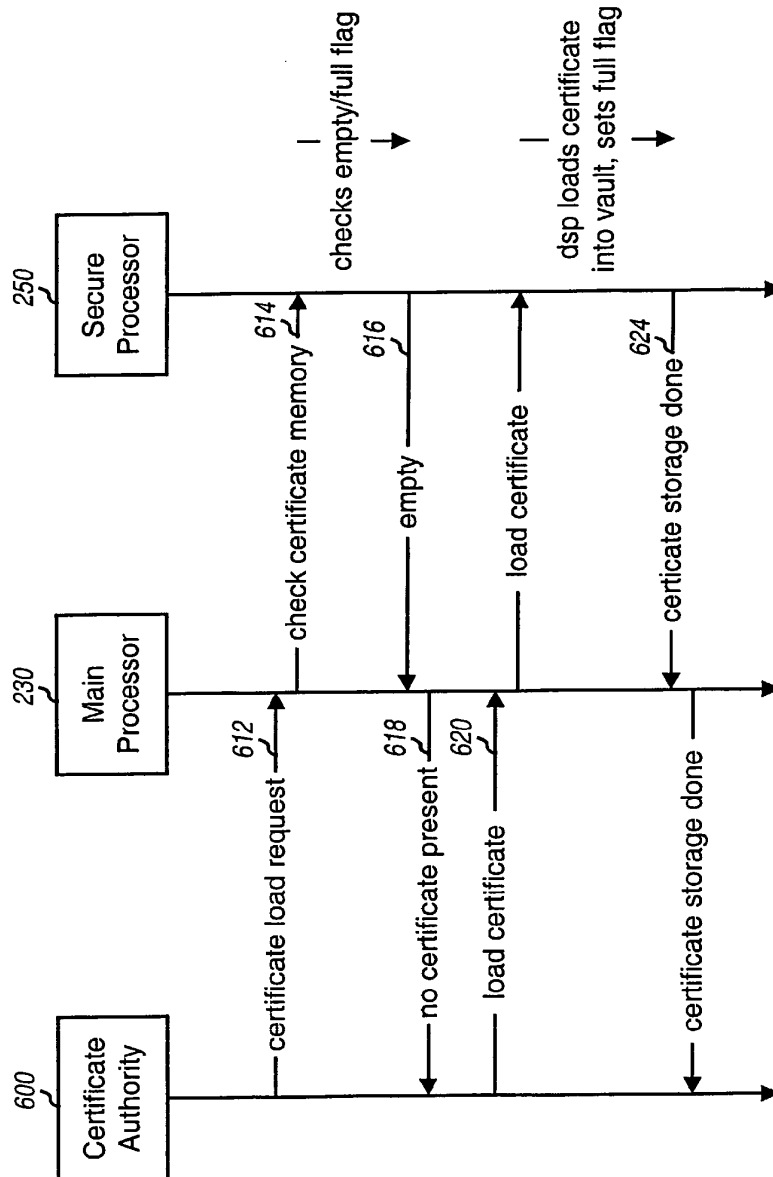


FIG. 6A

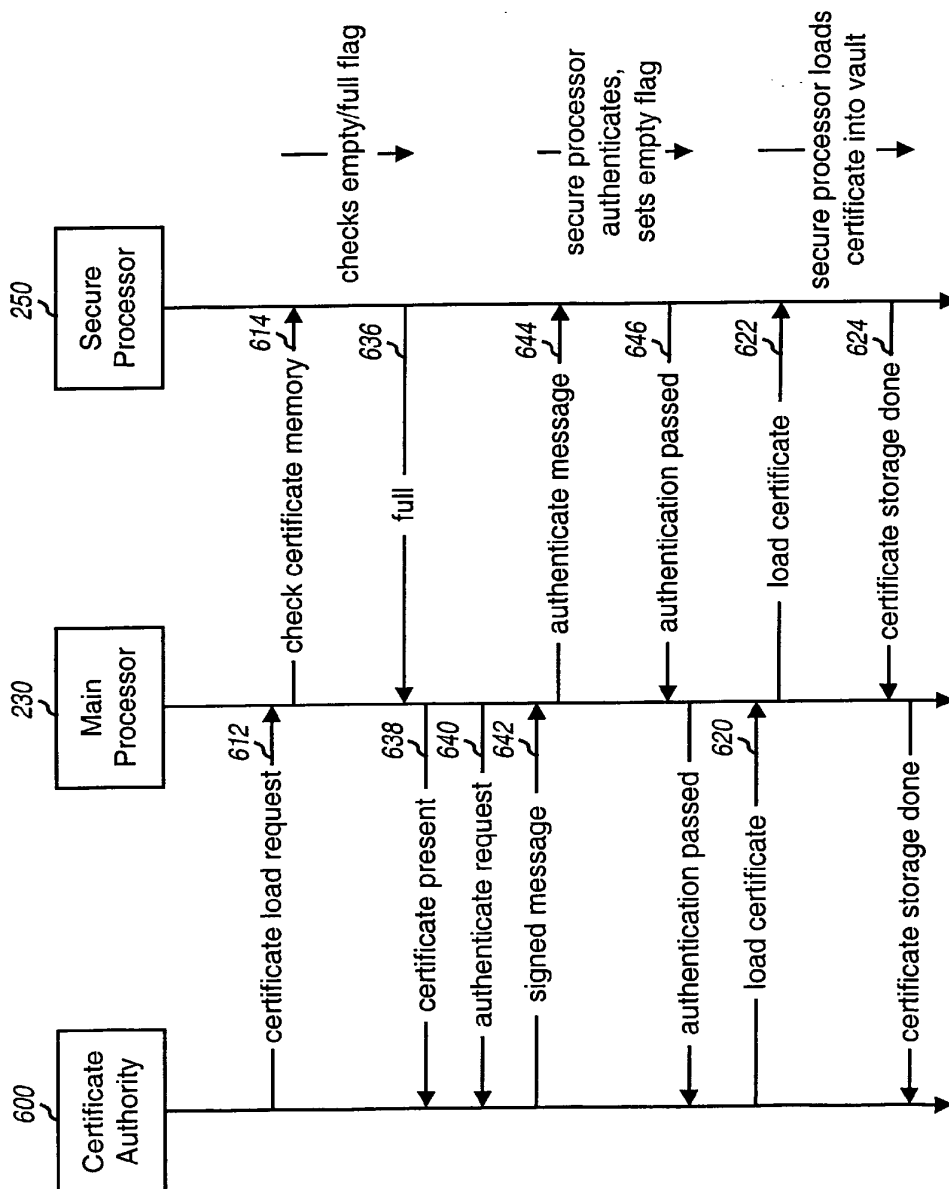


FIG. 6B



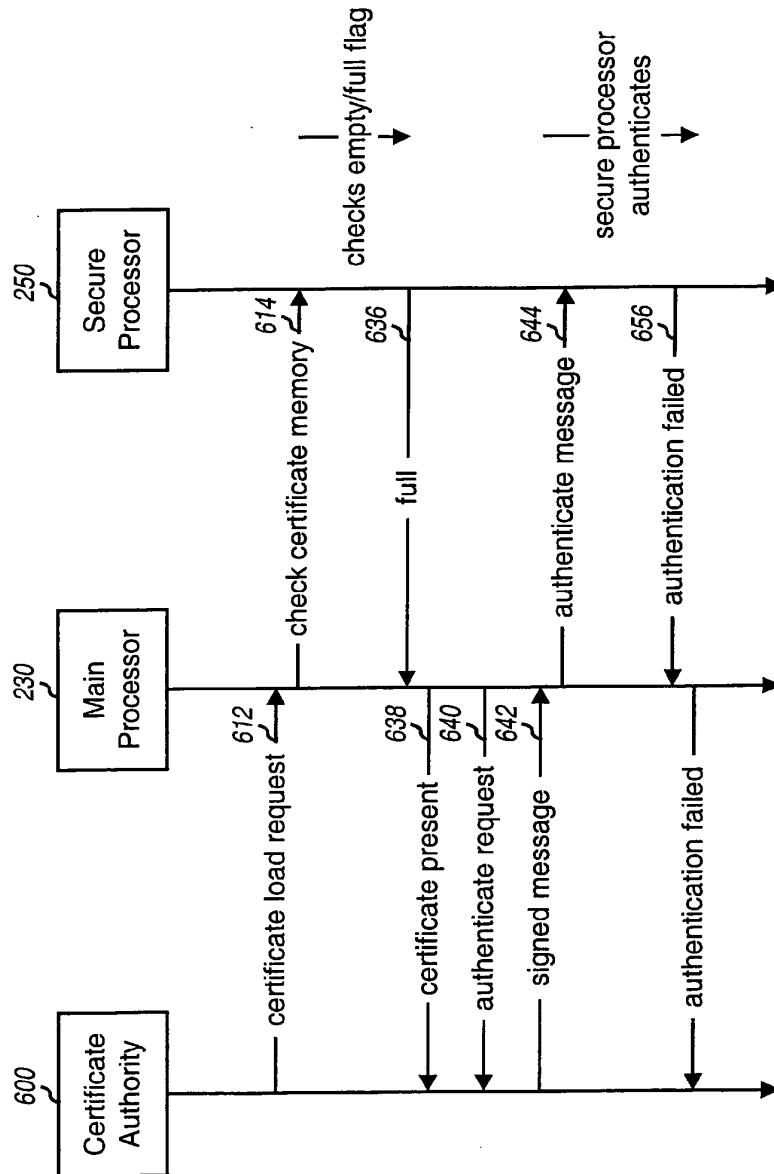


FIG. 6C

9/9

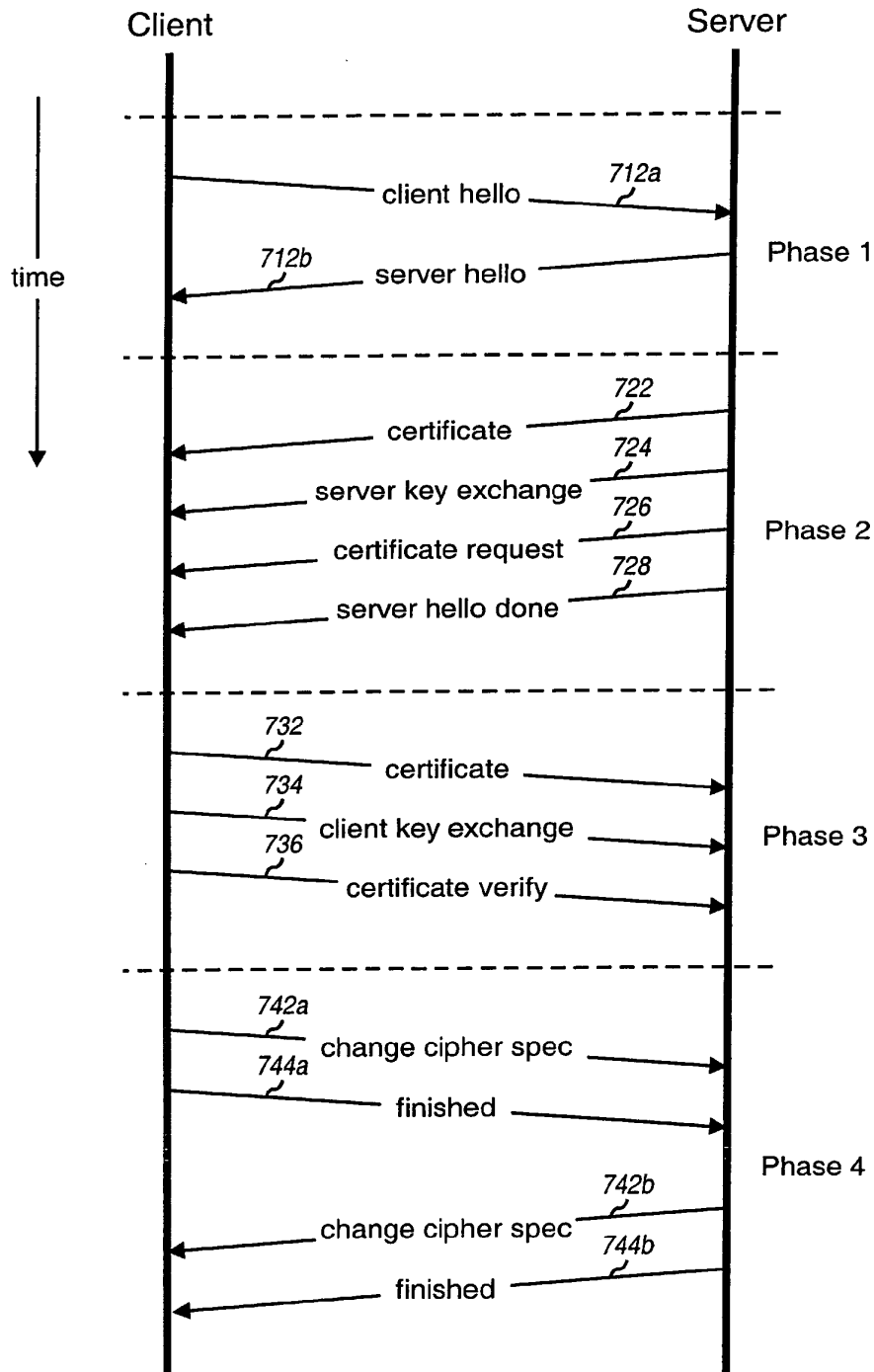


FIG. 7

## INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 02/07693

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 H04Q7/32 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04Q G06F

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

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

EPO-Internal, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 01 17296 A (ERICSSON TELEFON AB L M) 8 March 2001 (2001-03-08) abstract; figures 1,2 page 1, line 5 - line 9 page 4, line 13 - line 16 page 5, line 10 - line 14 page 6, line 1 -page 8, line 8 ---	1-26
A	US 6 026 293 A (OSBORN WILLIAM R) 15 February 2000 (2000-02-15) abstract column 1, line 4 - line 24 column 2, line 50 - line 61 column 3, line 16 - line 24 ---	1-26
A	US 6 084 968 A (HALL TIMOTHY GERARD ET AL) 4 July 2000 (2000-07-04) the whole document ---	1-26
	--- -/--	

☒ 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

17 July 2002

Date of mailing of the international search report

25/07/2002

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

Chêne, X

## INTERNATIONAL SEARCH REPORT

Inter Application No  
PCT/US 02/07693

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 224 160 A (PAULINI WERNER ET AL) 29 June 1993 (1993-06-29) column 6, line 63 -column 7, line 32 ---	1-26
A	EP 0 849 657 A (NCR INT INC) 24 June 1998 (1998-06-24) abstract; figure 1 -----	1-26

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

# INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. Application No

PCT/US 02/07693

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 0117296	A	08-03-2001	SE 515327 C2 AU 6887000 A WO 0117296 A1 SE 9903036 A	16-07-2001 26-03-2001 08-03-2001 28-02-2001
US 6026293	A	15-02-2000	AU 734212 B2 AU 4172297 A BR 9712007 A CN 1235743 A EE 9900084 A EP 0923842 A2 JP 2001500293 T PL 332050 A1 WO 9810611 A2	07-06-2001 26-03-1998 24-08-1999 17-11-1999 15-10-1999 23-06-1999 09-01-2001 16-08-1999 12-03-1998
US 6084968	A	04-07-2000	NONE	
US 5224160	A	29-06-1993	DE 3705736 A1 AT 103402 T DE 3888556 D1 EP 0280035 A2 ES 2050671 T3 JP 2564593 B2 JP 63240629 A	01-09-1988 15-04-1994 28-04-1994 31-08-1988 01-06-1994 18-12-1996 06-10-1988
EP 0849657	A	24-06-1998	EP 0849657 A1 JP 10282884 A US 6209099 B1 ZA 9710559 A	24-06-1998 23-10-1998 27-03-2001 24-05-1999

# (12) UK Patent Application (19) GB (11) 2 338 381 (13) A

(43) Date of A Publication 15.12.1999

(21) Application No 9812520.6

(22) Date of Filing 10.06.1998

(71) Applicant(s)  
**Barclays Bank Plc**  
(Incorporated in the United Kingdom)  
54 Lombard Street, LONDON, EC3P 3AH,  
United Kingdom

(72) Inventor(s)  
**David Alexander Taylor**  
**Mark Jonathan Stirland**

(74) Agent and/or Address for Service  
**R G C Jenkins & Co**  
26 Caxton Street, LONDON, SW1H 0RJ,  
United Kingdom

(51) INT CL<sup>6</sup>  
H04L 9/32

(52) UK CL (Edition Q )  
H4P PDCSA  
U1S S2209

(56) Documents Cited  
EP 0782296 A2 WO 97/41539 A1 WO 97/23972 A1  
US 5615268 A US 5602918 A US 5590197 A  
Internet Cryptography by R E Smith Pub Addison  
Wesley 1997 ISBN 0-201-92480-3 pp 113-116 262-265

(58) Field of Search  
UK CL (Edition P ) H4P PDCSA PDCSC  
INT CL<sup>6</sup> H04L 9/32  
Online:WPI,EPODOC,PAJ

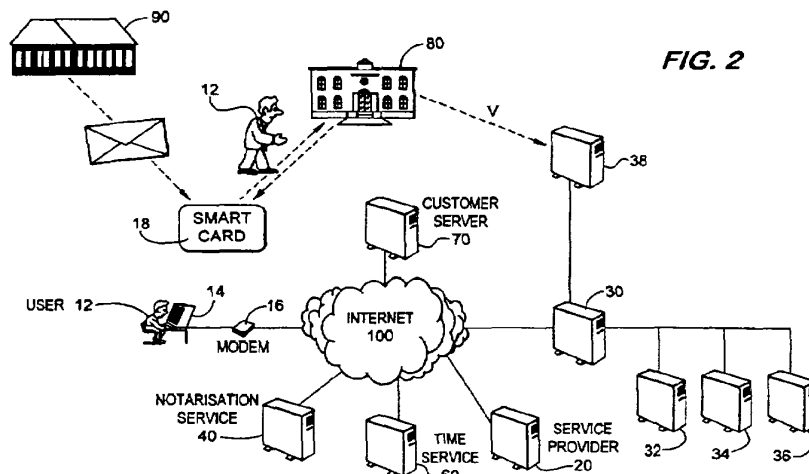
Exhibit E9

EP 3 229 099 (17000713.2)

BARDEHLE PAGENBERG  
Our ref.: S154820EPT1EIN

(54) Abstract Title  
**Cryptographic authentication for internet using two servers**

(57) In a system for the authentication of transactions over a public network (100), a terminal (14) sends digitally signed transaction data (SD) to a service provider (20) over the public network (100), together with card application data (CAD) generated by a smart card (18). The card application data (CAD) is sent to an authorization server (30) which checks that the smart card (18) is valid and that the card application data (CAD) must have been generated by that smart card (18) in the current transaction. User identification information (ID) is also sent from the terminal (14) to the service provider (20) and thence to the authorisation server (30), where this information (ID) is checked against the correct user details for the smart card (18). The results of these checks are indicated in a digitally signed authorisation response (ARES) from the authorization server (30) to the service provider (20), which then determines whether to proceed with the transaction by setting acceptance criteria for the current transaction and determining from the authorisation response (ARES) whether these criteria are met.



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

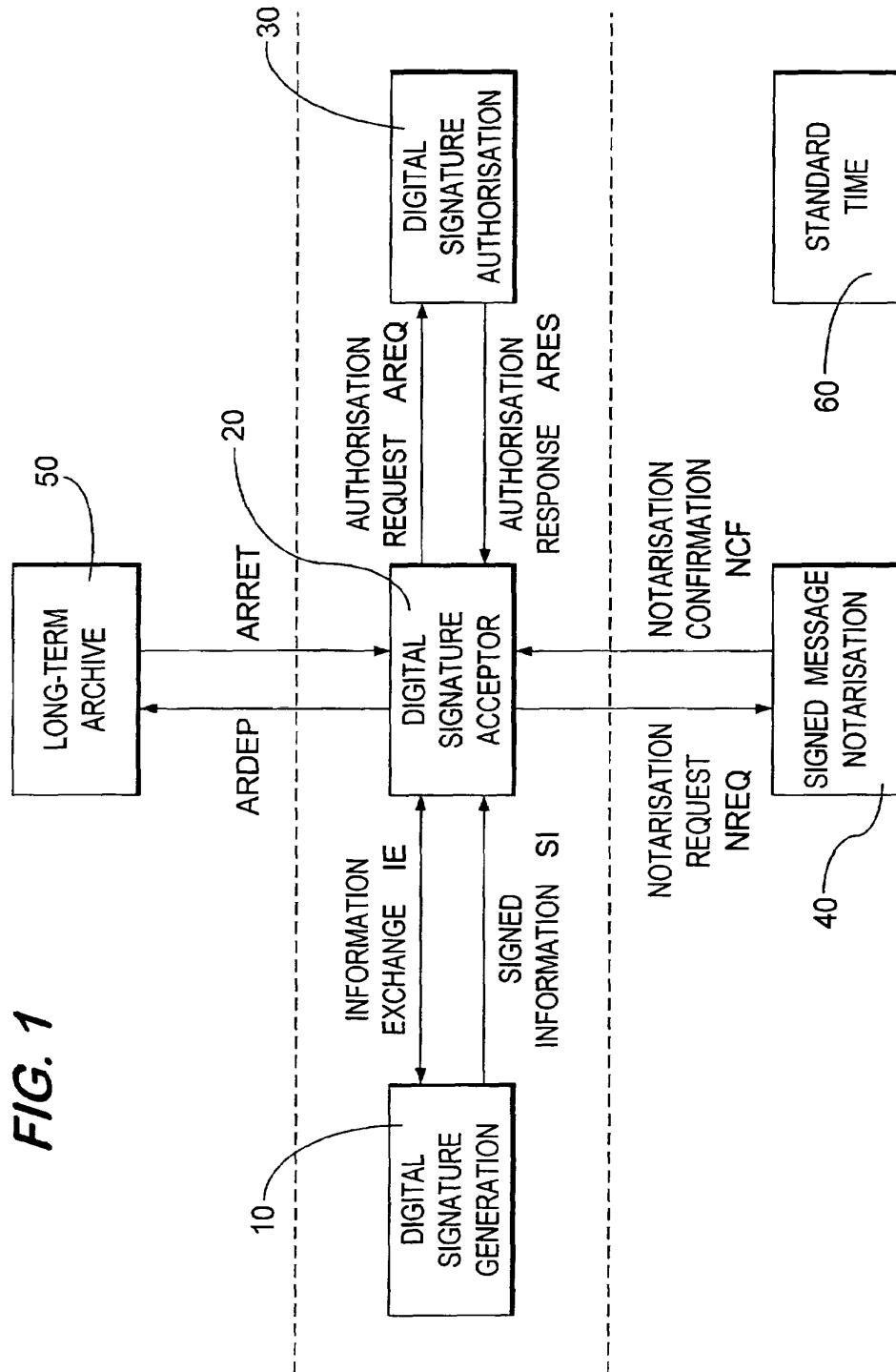
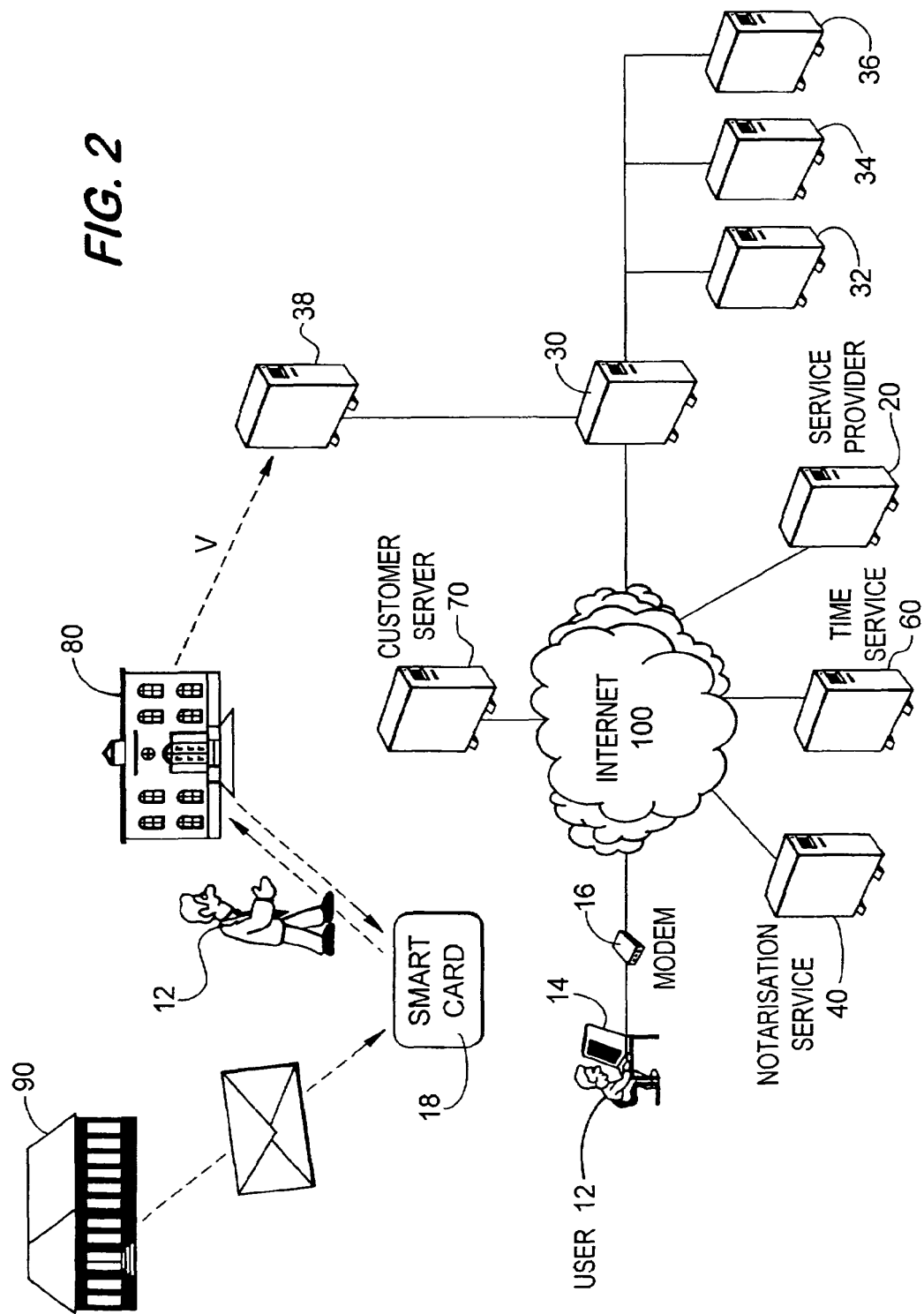
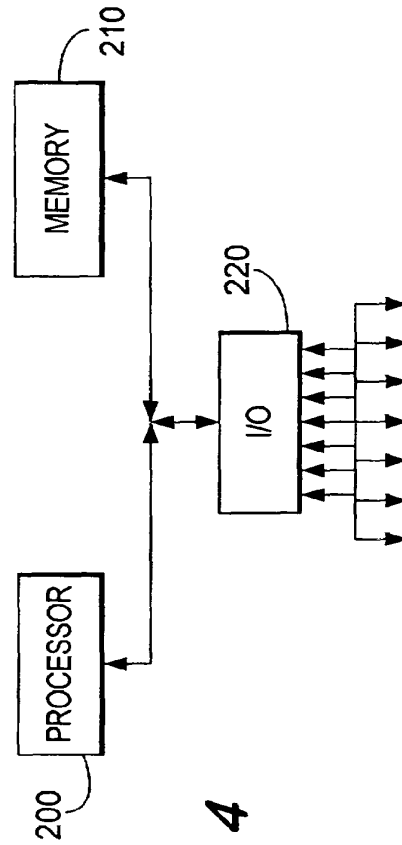
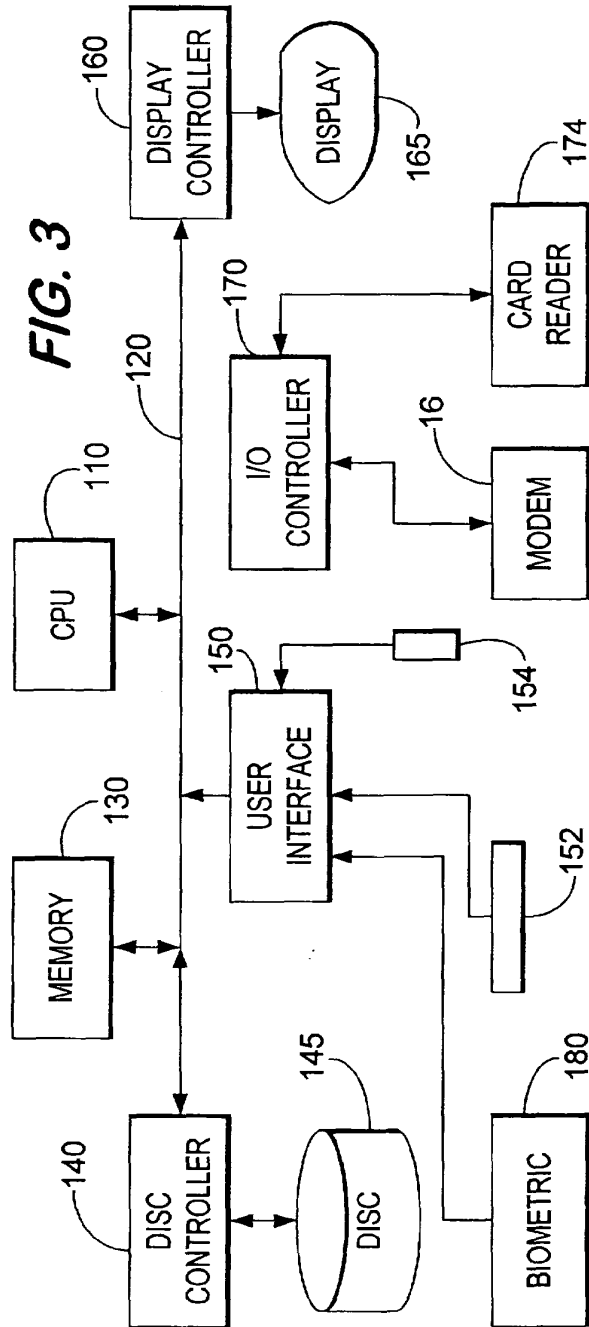


FIG. 2







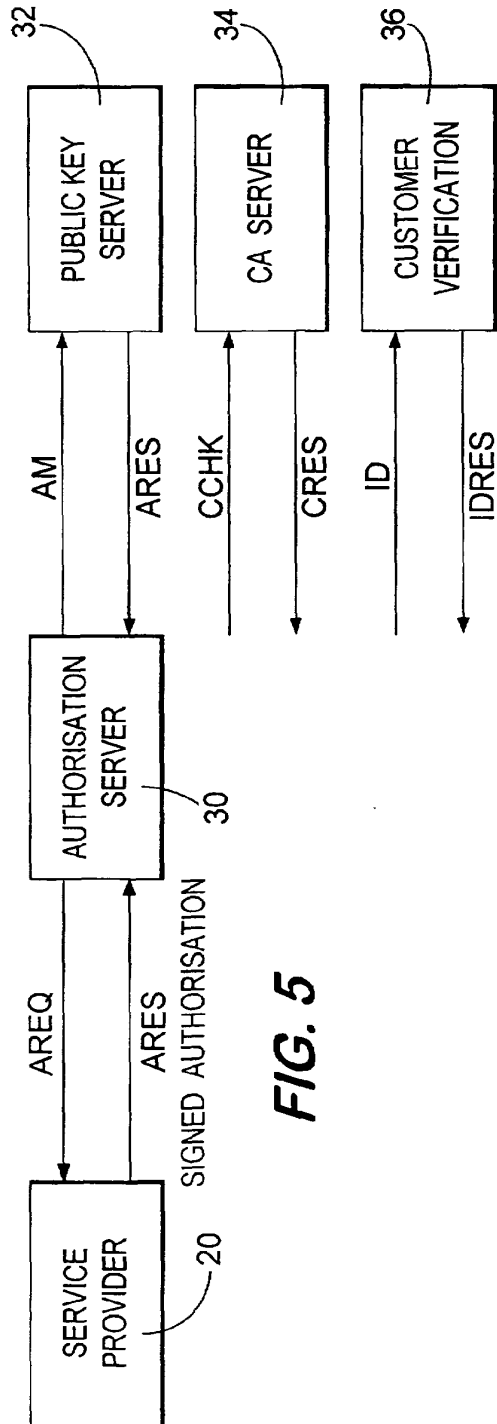


FIG. 5

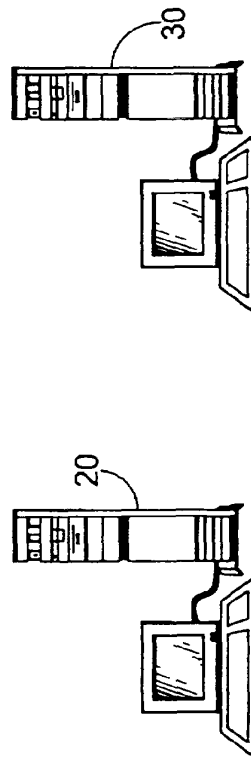
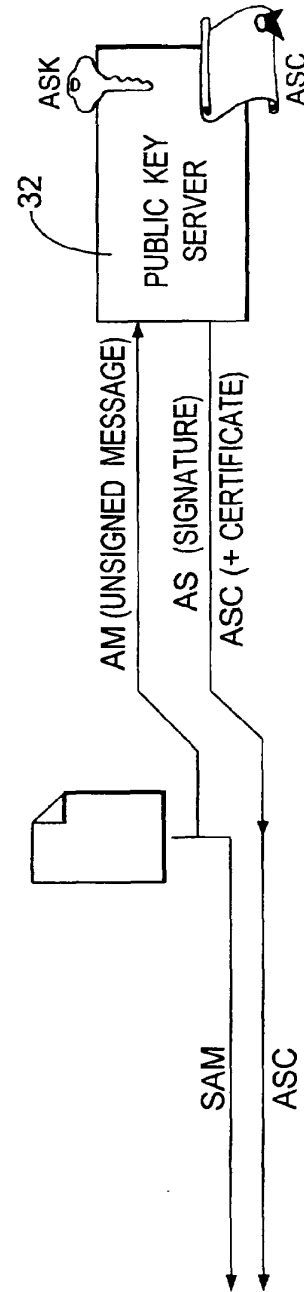
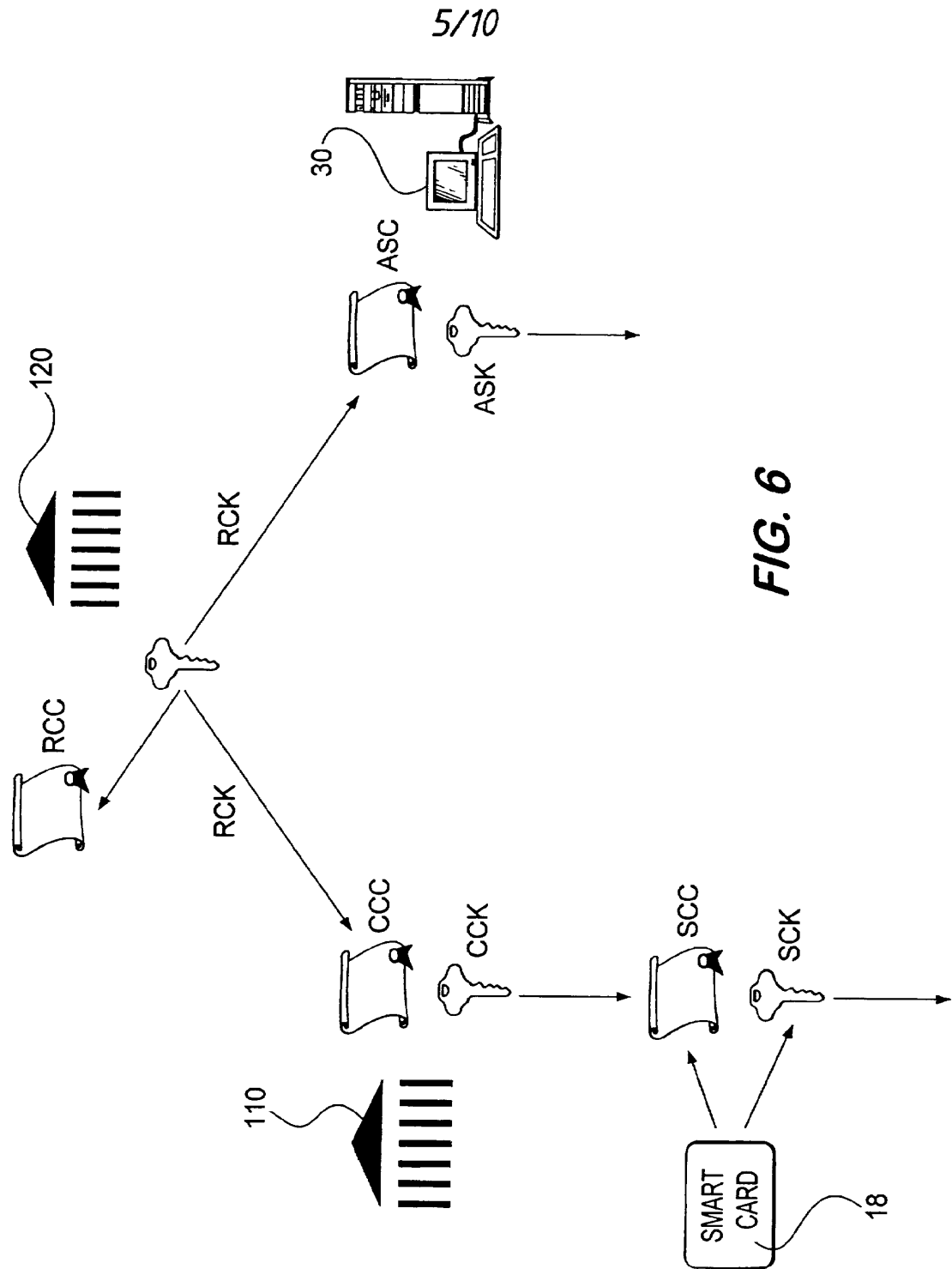


FIG. 13

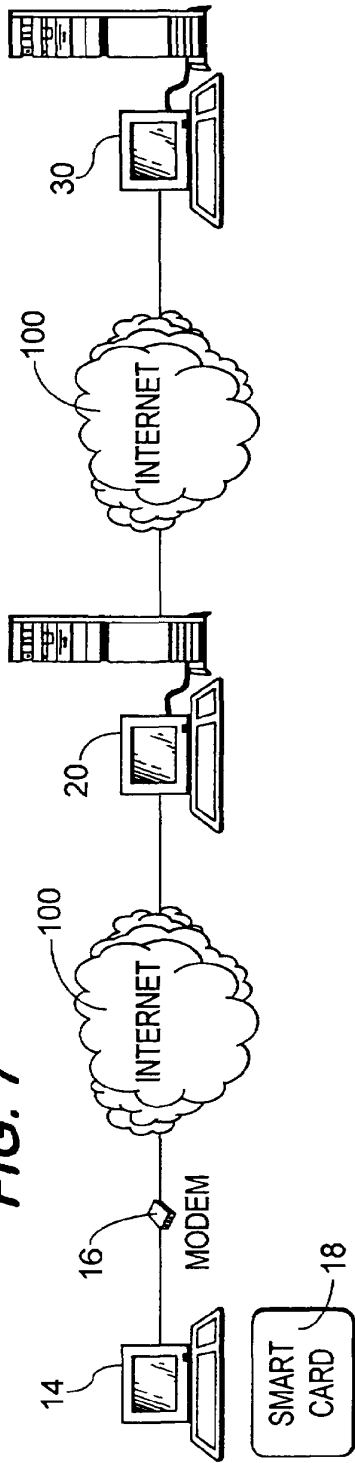




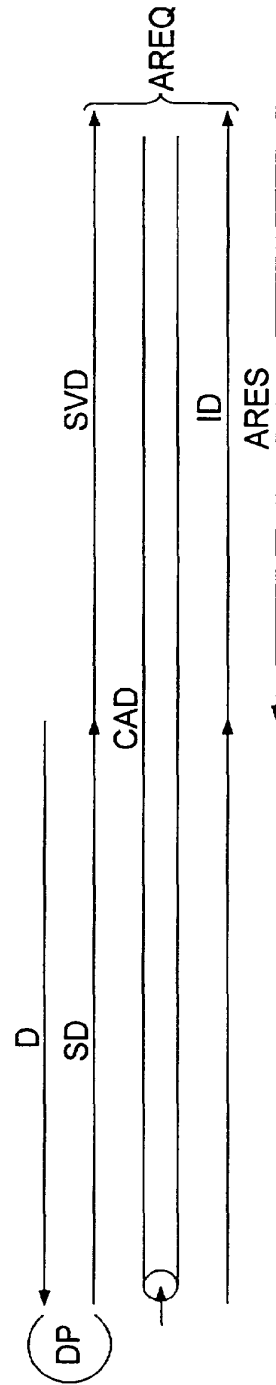
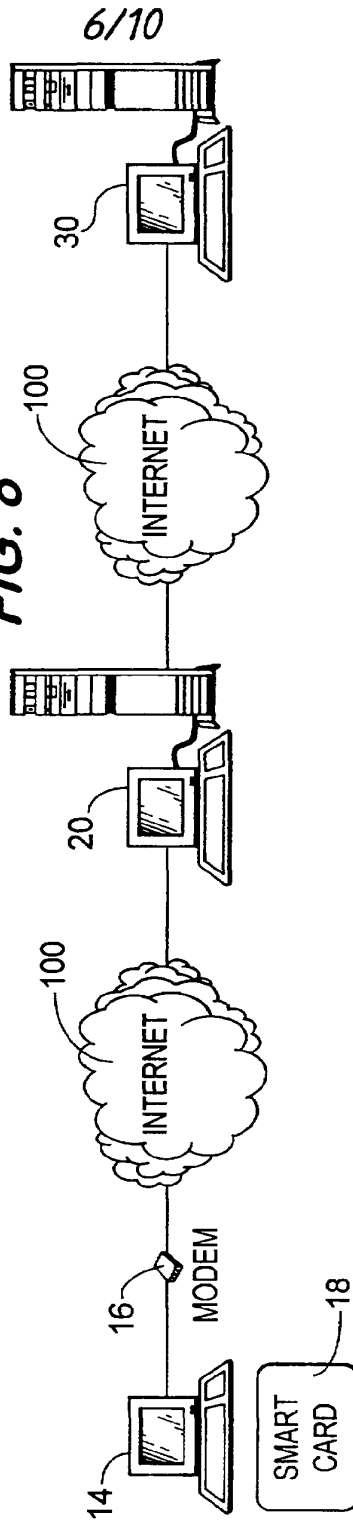
**FIG. 6**

5/10

**FIG. 7**



**FIG. 8**



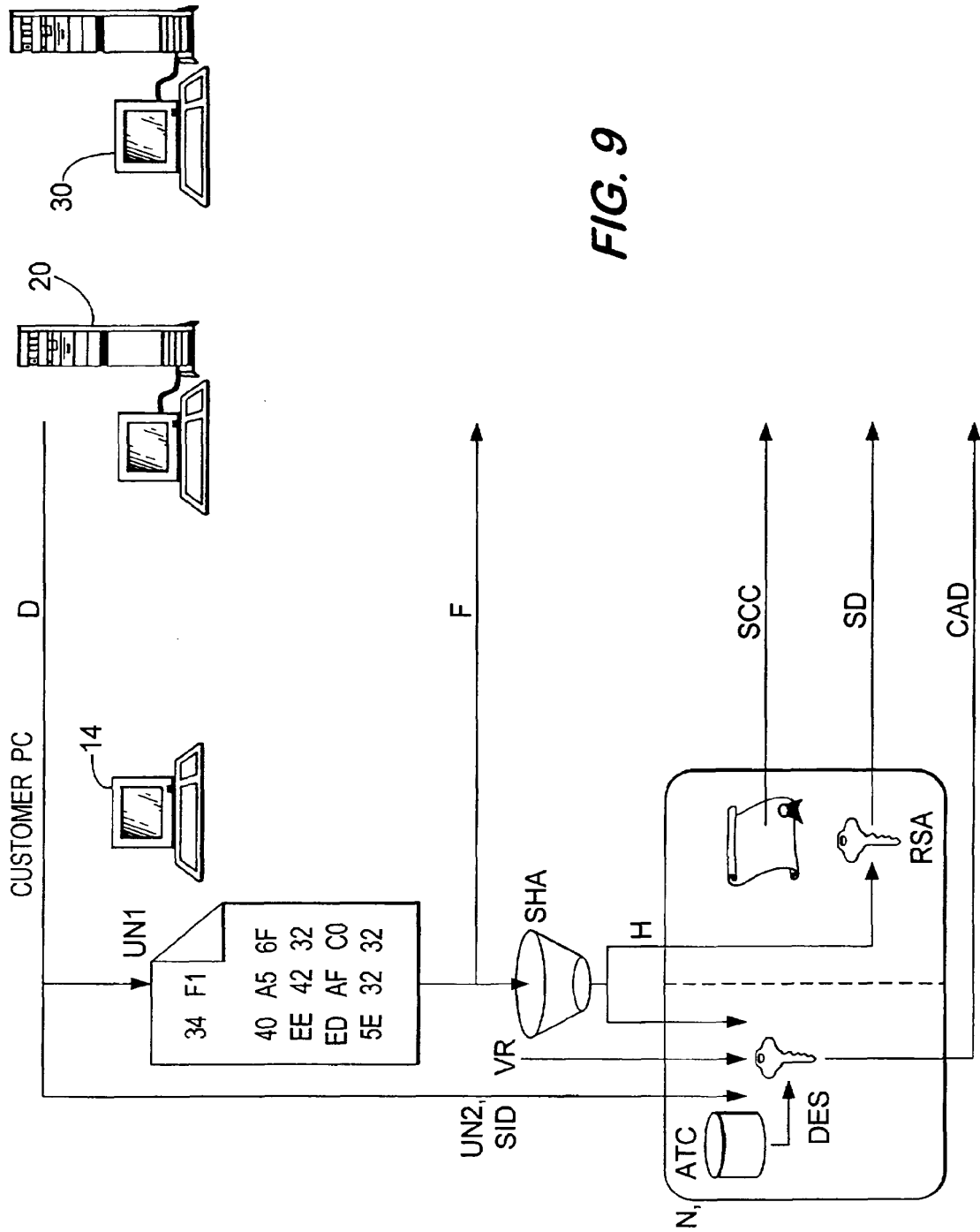


FIG. 9

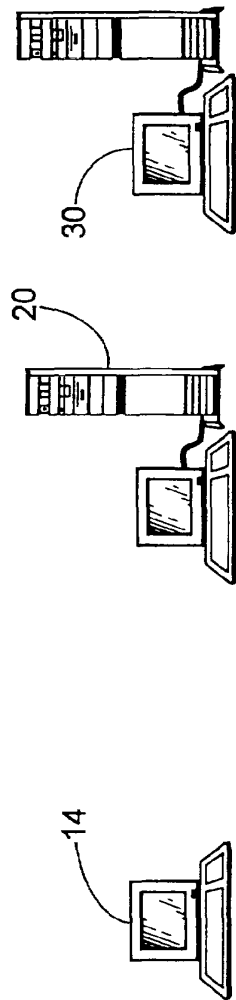


FIG. 10

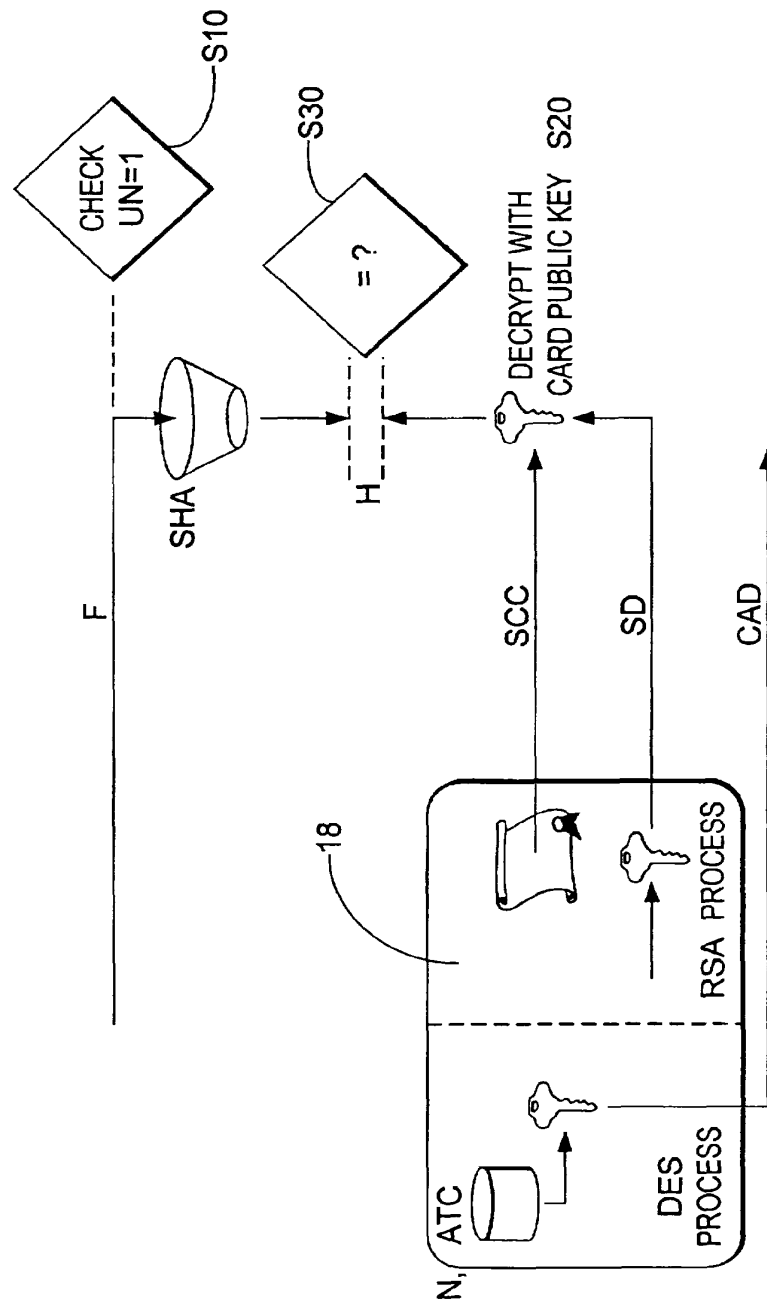


Diagram illustrating three computer systems:

- 14: A laptop computer.
- 20: A desktop computer system consisting of a monitor, a tower unit, and a keyboard.
- 30: A desktop computer system consisting of a monitor, a tower unit, and a keyboard.

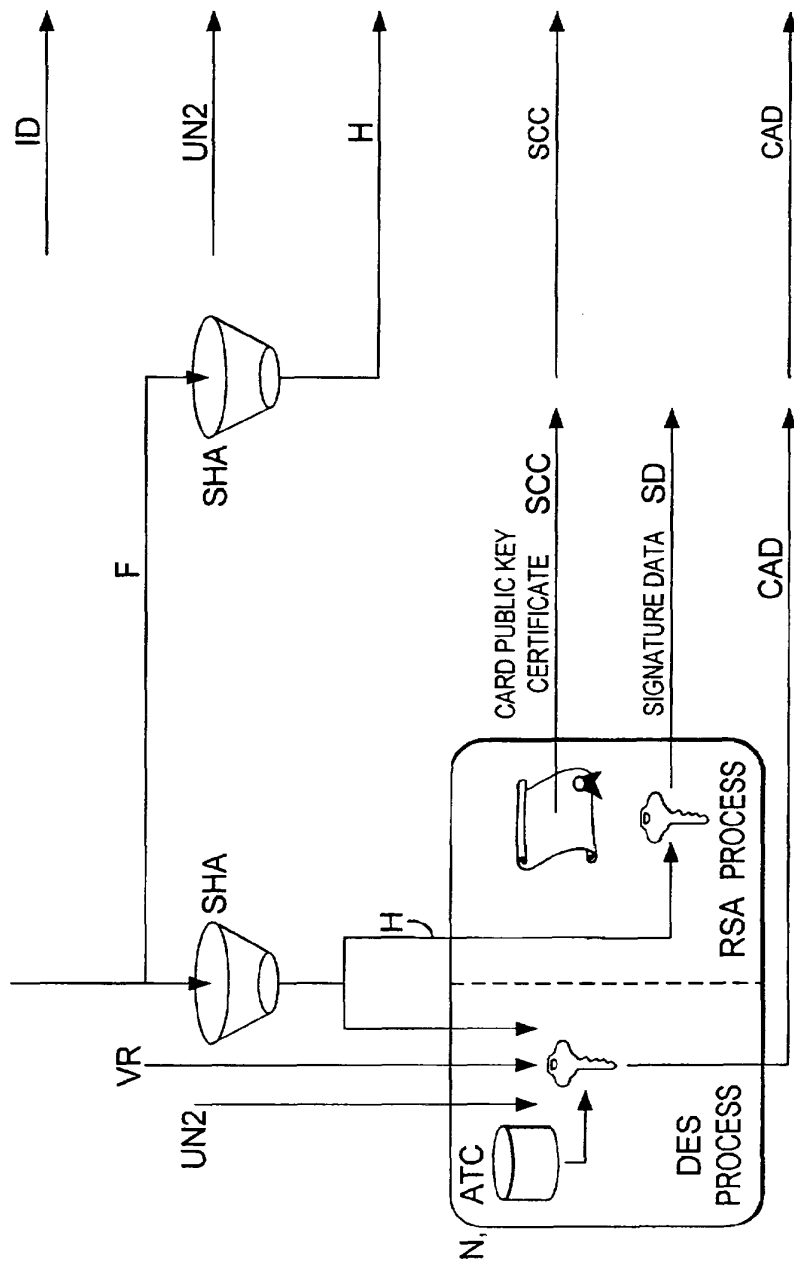
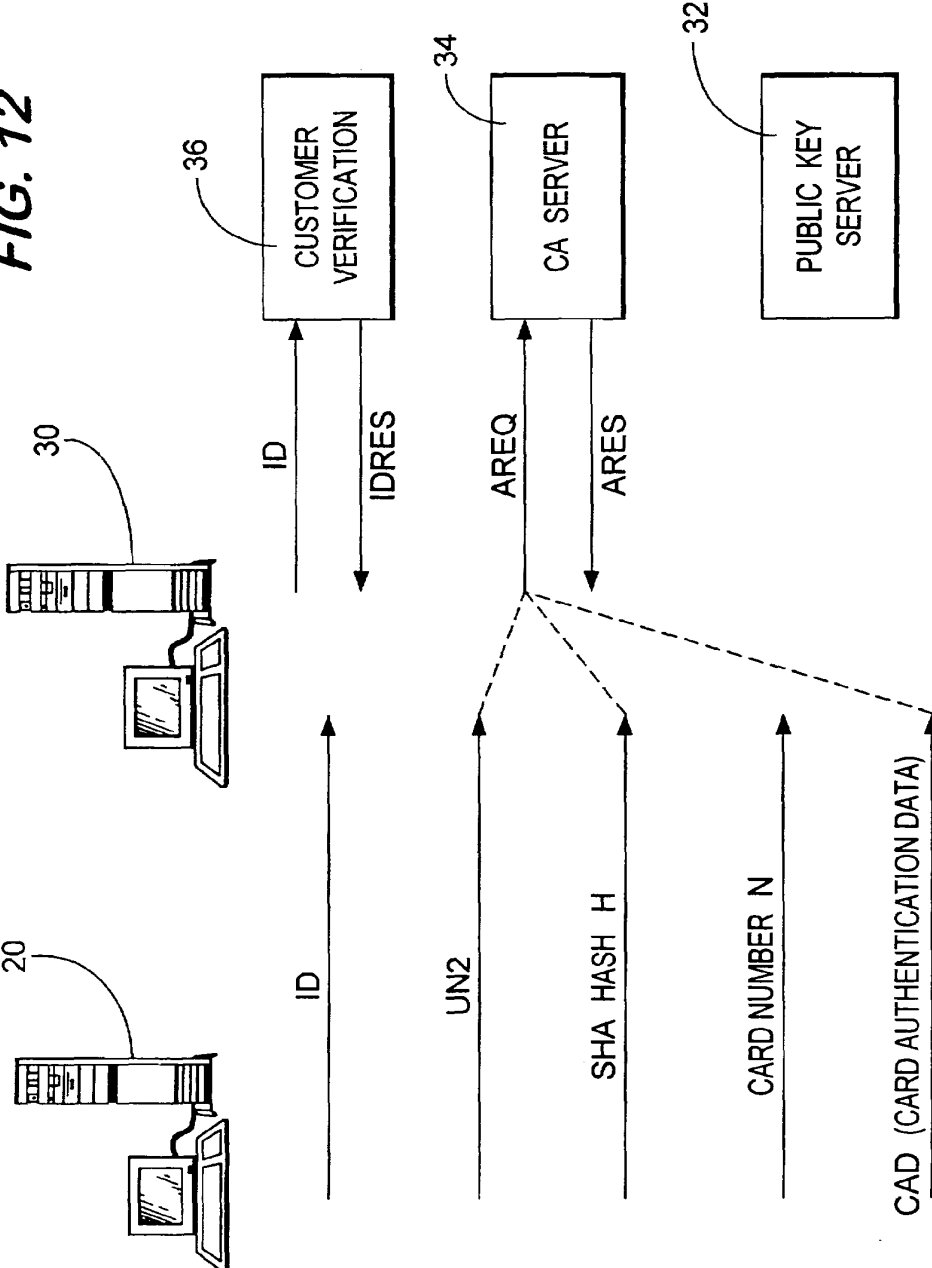


FIG. 12





SECURE TRANSACTION SYSTEM

The present invention relates to a secure transaction system,  
particularly for use over a public network.

The most common and well-known public data network is the Internet,  
5 which provides network access to members of the public at low cost. Many  
types of commercial transaction which were previously conducted via  
telephone, mail or private network may now be conducted more easily over  
the Internet. However, internet protocols were not designed for security, and it  
has therefore been necessary to provide additional protocols for secure  
10 transactions over the Internet, including transport-level protocols such as the  
Secure Sockets Layer (SSL), and application-level protocols such as the  
Secure Hypertext Transfer Protocol (SHTTP). Such protocols aim to prevent  
interception, decryption and forgery of transactions between a client and a  
server over the Internet, but they do not verify the identity of the user of the  
15 client terminal. For example, in a credit card transaction, only the user's name  
and address, and the card number and expiry date need be supplied to order  
goods or services over the Internet. It is comparatively easy to obtain the  
necessary information to carry out fraudulent transactions over the Internet.  
Some verification of the user's identity is usually implemented at the  
20 application level, such as the use of passwords, but these too can be obtained  
or guessed.

Nevertheless, various protocols have been proposed for allowing secure transactions, and particularly secure payment, over the Internet; one example is Secure Electronic Transaction (SET), a protocol for credit card transactions over the Internet, a modification of which is described in WO 97/41539. Typically, such transactions involve three parties: a client which  
5 supplies credit card details, a service provider server operated by a supplier of goods or services, and an authorisation server which checks the credit card details and informs the service provider server whether payment is authorised by the operator of the credit card system.

10 However, conventional electronic transaction systems do not support non-repudiation; in other words, they do not provide sufficient evidence to confirm that a specific transaction was authorised.

Moreover, conventional electronic transactions do not bind a specific user to the use of an authorisation card.

15 Furthermore, conventional electronic transaction systems are limited in their application, because the authorising server is designed merely to give an authorise or decline message on the basis of the details supplied.

According to one aspect of the present invention, there is provided an electronic transaction system in which a terminal combines transaction data  
20 which is unique to a current transaction with terminal data which is unique to that terminal to generate bound terminal/transaction information which is sent to the transaction server. The transaction server sends the transaction data to

an authorisation server which returns information which binds the transaction to the identity of the authorisation server. The transaction server then has available information which binds together the transaction, the terminal and the authorisation server in a form which cannot be fraudulently created by the transaction server, and therefore cannot be repudiated.

According to another aspect of the present invention, there is provided an electronic transaction system in which an authorisation token is issued to a user. Information confirming the identity of both the authorisation token and the user are presented to an authority which confirms the validity of this information and makes it available to an authorisation server. The authorisation token is then used in an electronic transaction with a transaction server, in which user identification information and authorisation token information are supplied to the transaction server and passed to the authorisation server. The authorisation server compares the user identification information and authorisation token information with information previously made available by the authority and indicates to the transaction server the result of the comparison. In this way, because the correspondence between the user and the token has been verified before the transaction, the use of the token by the user can be confirmed during the transaction.

According to a further aspect of the present invention, there is provided an electronic transaction system in which a user terminal transmits to a transaction server transaction data and identification data. The transaction

data is passed from the transaction server to an authorisation server, which compares the identification data with stored identification data relating to authorised users of the system. The authorisation server then transmits to the transaction server an authorisation message indicating no authorisation, partial  
5 authorisation or full authorisation. In response to a partial authorisation message, the transaction server determines whether to accept the transaction data on the basis of the data content of the transaction data.

Specific embodiments of the present invention will now be described with reference to the accompanying drawings, in which:

- 10        Figure 1 is a diagram of the service functions of an authorisation system in an embodiment of the present invention;
- Figure 2 is a diagram of the architecture of the system;
- Figure 3 is a diagram of the hardware components of a user terminal;
- Figure 4 is a diagram of the hardware components of a smartcard;
- 15        Figure 5 is a diagram of the sub-systems within the authorisation server in the embodiment;
- Figure 6 is a diagram of the cryptographic architecture of the system;
- Figure 7 is a diagram illustrating a more specific embodiment in which the system is used to authorise an electronic form;
- 20        Figure 8 shows the flow of data in the specific embodiment;
- Figure 9 shows the cryptographic processes applied by the user terminal;

Figure 10 shows the digital signature validation process applied by the electronic form server;

Figure 11 shows the transmission of an authorisation request message to the authorisation server;

5        Figure 12 shows the authorisation process performed by the authorisation server; and

Figure 13 shows the authorisation response message process from the authorisation server to the electronic form server.

#### 10        Authorisation Service

Figure 1 shows the service functions which provide a digital signature generation and authentication service in an embodiment of the present invention. A digital signature generator 10 generates digital signatures in the form of data which uniquely identifies a specific party and binds the signed data to that party. Digital signatures are a known technique for protecting data from modification and for identifying the signing party. The signing party is provided with a private/public key pair, which can be used to generate and verify digital signatures. The 'party' is usually assumed to be the user operating the computer which stores the private key and generates the digital signature. However, in the present embodiment, the private key is stored on a smart card and therefore the digital signature binds the signed data to the smart card. Hence, the 'party' is the smart card.

15

20

As is well-known, data encrypted with a private key can be decrypted with the corresponding public key, and vice versa. Suitable cryptographic algorithms are the RSA algorithm, described for example in US 4,405,829, and the Diffie-Helman algorithm described in US 4,200,770. In a typical digital signature process, the data to be 'signed' is subjected to a hash algorithm, such as the Secure Hash Algorithm (SHA). The result is a type of checksum, known as a 'hash', which depends both on the values of the bits making up the data and the positions of those bits. It is therefore very difficult to modify the data while giving the same hash. The hash is then encrypted using the private key, to produce a digital signature.

A digital signature can be verified by performing the same hash algorithm on the received data as was used to generate the hash encoded in the digital signature. The digital signature is then decrypted using the public key and the decrypted hash is then compared to the calculated hash. If the hashes match, the signature is valid.

Examples of digital signature algorithms are the RSA digital signature, in which the hash is encrypted together with an indication of the type of the hash algorithm using RSA encryption, and the digital signature algorithm (DSA), in which the hash algorithm is the SHA and the hash is encrypted together with a random number by a variant of the Diffie-Helman algorithm.

The digital signature generator 10 exchanges information IE with a digital signature acceptor 20, and sends 'signed' information SI to the digital

signature acceptor which includes a digital signature generated using a private key held by the digital signature generator. On receipt of the 'signed' data, the digital signature acceptor 20 sends an authorisation request AREQ to a digital signature authoriser 30. The authorisation request AREQ includes information  
 5 derived from the signed information SI. The digital signature authoriser 30 checks the information derived from the signed information SI, determines whether the digital signature should be accepted as genuine, and sends to the digital signature acceptor an authorisation response ARES indicating whether the signature is to be accepted as genuine. The authorisation response ARES  
 10 is signed with a digital signature generated by a private key held by the digital signature authoriser 30. Dependent on the authorisation response, the digital signature acceptor then accepts or rejects the signed information SI.

In order to provide evidence of the provision and acceptance or rejection of the signed information, the digital signature acceptor 20 may  
 15 further transmit a notarisation request NREQ to a signed message notariser 40, including information derived from the signed information SI and from the authorisation response ARES. In response, the signed message notariser 40 transmits to the digital signature acceptor 20 a notarisation confirmation NCF which includes information derived from the notarisation request NREQ,  
 20 digitally signed by the signed message notariser 40. The digital signature acceptor 20 transmits archive deposit information ARDEP, which may for example comprise the signed information SI, the authorisation response

ARES and the notarisation confirmation NCF, to a long-term archive 50. This information is later retrieved from the long-term archive 50 as archive retrieval information ARRET in the event of repudiation of the signed information SI.

5            Optionally, where an independent time reference is needed, for example to confirm the exact time of a transaction such as the sending and receipt of the signed information SI, information from a standard time signal source 60 is made available to each of the functions described above and is incorporated in each digitally signed message.

10

#### System Architecture

A high-level system architecture of the digital signature authorisation service is shown in Figure 2. Like parts to those of Figure 1 carry the same reference numerals.

15            A user 12 wishing to subscribe to the digital signature service must first apply for a smart card 18 from which the user's digital signatures are derived. The user 12 has a computer 14 connectable to the Internet 100 via a modem 16, using for example web browser software and TCP/IP protocols as is well-known in the art.

20            The components of the computer 14, which may be an IBM-compatible 'personal computer' or Apple Macintosh computer, are shown in Figure 3. A CPU 110 is connected through a bus 120 to main memory 130, a



disc controller 140 connected to a hard disc 145, a user interface controller 150 connected to a keyboard 152 and other input devices such as a mouse 154, a display controller 160 connected to a visual display 165, and an I/O controller 170. The I/O controller 170 controls the modem 16 and a card reader 174 into which a smart card can be docked. Optionally, a biometric device 180 is connected to the I/O controller 170 or to the user interface controller 150. The biometric device 180 may be a fingerprint scanner, an iris scanner, or another device which allows information derived uniquely from the user 12 to be input to the computer 14. The fingerprint scanner may be integrated with the keyboard 152.

The user accesses a customer server 70 through the Internet 100 and requests subscription to the digital signature service. The request is passed on to a card bureau 90 by a suitable form of secure communication. The card bureau 90 sends a smart card 18 to the user 12. An example of the components within the smart card 18 is shown in Figure 4.

The smart card 18 contains a processor 200 connected to a memory 210 and an external connector 220. The card 18 may include a power source such as a cell integrated within the card 18, or power may be supplied through the connector 220 by the card reader 174. At least part of the memory 210 is non-volatile so that a operating program and data are stored when the card 18 is removed from the reader 174.

A public/private key pair, a card identity code and a PIN are recorded in the non-volatile memory of the card 18 during manufacture. When a request is received from the customer server 70, the name of the user 12 is printed on the card 18, which is then sent to the user 12. A status message is sent to the authorisation server to indicate that the card 18 is issued but inactive.

The user 12 then takes the card 18 to the public premises 80 of an organisation which supports the digital signature service, where the user's identity is checked against the identity of the user to whom the card 18 has been issued. Once the user's identity has been verified, a status message is sent from the premises 80 to the authorisation server 30 identifying the card 18 and the user 12. The user 12 is also issued with the card reader 174 and application software for the digital signature service, if the user 12 does not already have these. Subsequently, the user 12 is notified of the PIN.

To use the authorisation service, the user 12 inserts the card 18 into the card reader 174. Application software running on the computer 14 then prompts the user 12 for a PIN. The user enters a PIN which is compared to that stored on the card 18, and the application generates a card validation result (VR) which indicates whether a PIN has been requested and whether the entered PIN matched that stored on the card 18.

Additionally or alternatively, the computer 14 obtains biometric data from the biometric device 180 if this is present. The smartcard 18 generates a

digital signature for the signed information SI transmitted by the computer 14 to the service provider 20, as will be described in a specific example below.

The service provider 20 may comprise a general purpose computer running web server software and connected to the Internet 100. The service provider 20 also runs authorisation software for communication with the  
5 authorisation server 30.

The authorisation server 30 comprises a general purpose computer running authorisation server software and connected to the Internet 100. As shown in Figures 2 and 5, the authorisation server 30 is also connected, for  
10 example over a local or private network, to a public key server 32, a card authentication server 34 and a customer verification server 36. The public key server 32 and the card authentication server 34 comprise dedicated hardware modules including encryption /decryption acceleration hardware. The customer verification server 36 stores a database containing the details of  
15 authorised users of the digital signature service.

The authorisation server 30 receives from the service provider 20 the authorisation request AREQ, which includes a hash of the signed information SI, a public key certificate, identification information relating to the card 18 and the user 12, and card authentication information. The authorisation server  
20 30 sends the card authentication information CCHK to the card authentication server 34, which checks the authenticity of the card 18 and returns a response CRES indicating whether the card is authentic or not. The authentication

server 30 sends the user identification information ID to the customer verification server 36, which returns a response IDRES indicating whether, or to what extent, the customer identity details are correct.

The authorisation server 30 generates from the responses CRES and IDRES an authorisation message AM, which is sent to the public key server 32. The public key server 32 signs the authorisation message AM to produce a signed authorisation response ARES which is transmitted to the service provider 20.

#### 10 Public Key Hierarchy

The digital signatures are generated, verified and authorised using public key cryptography, as explained above. The public keys are distributed by means of public key certificates, so that users of public keys can trust that the public keys belong to the parties with which the users wish to communicate. As is well known, a public key certificate consists of a name identifying a party who holds a private key (in this case, the card 18), the corresponding public key, and a digital signature comprising a hash of the name and the public key, encrypted using the private key of a certification authority trusted by the user. If the user does not have the certification authority's public key, this can be obtained from the certification authority's public key certificate, which is signed by a root certification authority. Thus, a

hierarchy of public key certificates may be used, ultimately administered by a root certification authority which is always trusted.

If a public/private key pair is changed, however, the public key certificate is no longer valid. The old public key certificate may be revoked by placing a code identifying that certificate on a Certificate Revocation List CRL, which is circulated periodically to users of the public key. Before a public key certificate is used, it may first be checked against the CRL.

Figure 6 shows the hierarchy of keys in the present embodiment. The smartcard 18 performs the digital signature process using a private key SCK stored within the card 18. The corresponding public key is contained within a smart card certificate SCC stored within the card and signed using a card certification private key CCK of a card certification authority 110. The corresponding public key is contained within a card certification authority certificate CCC which is signed by a root certification authority private key RCK of a root certification authority 120. The root certification authority private key RCK is also used to sign the public key certificate ASC of the authorisation server 30. The corresponding public key is distributed in a root certification authority public key certificate RCC, which is self-signed using the corresponding private key RCK. The authorisation server private key ASK is used to provide a digital signature on the authorisation response ARES.

#### Card Authentication

In addition to the public key transactions described above, the smart card 18 also performs a symmetric key card authentication function with the authentication server 30, using a separate private key stored on the card 18. The process uses a two-key triple data encryption standard (DES, encrypt-decrypt-encrypt) algorithm in Cipher Block Chaining Mode to produce an eight byte message authentication code (MAC).

The symmetric key authentication function is used for secure communications between the smartcard 18 and the authentication server 30, which the service provider 20 passes transparently but is unable to decrypt.

For each card authentication transaction, the MAC is calculated from a combination of: data stored internally in the card 18, including a variable application transaction counter value ATC which is incremented for each transaction, and a card identity number N which is included in the public key certificate SCC; data supplied by the service provider 20, including the time, date and the identity of the service provider, so as to bind the MAC to the specific transaction; a hash of the signed information SI, so as to bind the MAC to the data supplied and to the digital signature; and the validation result VR or information derived from the biometric data.

#### Electronic Forms Implementation

A more specific embodiment will now be described with reference to Figures 7 to 13, in which the digital signature system is used to authenticate a

completed form supplied electronically by the user 12 to the service provider 20, which is in this case operated by a government department. For convenience, Figures 7 to 13 show the topological connection between the computer 14 and the service provider 20, and between the service provider 20 and the authorisation server 30, as being through separate networks but in this example the connections are both through the Internet.

Figure 8 shows the data transmission which takes place during a transaction. The computer 14 has already established an Internet connection to the service provider 20, and is running web browser software. In response to a request initiated by the user 12, the service provider 20 sends data D comprising HTML pages representing a blank form (such as a form for registering a new business). The user 12 completes the form by entering data within the browser software and requesting submission of the completed form to the service provider 20. The browser software supports the digital signature service, for example by means of a 'plug-in' which modifies standard browser software, so that the user 12 is prompted to enter a PIN when requesting submission of the completed form. If the smartcard 18 is not located in the card reader 174, the software prompts the user 12 to do this.

At a data processing stage DP, the smartcard 18 generates a digital signature from the form data F of the completed form and the smart card private key SCK. The signed data SD is then sent to the service provider 20, which verifies the signature by recovering the card public key from the smart

card certificate SCC using the card certification public key recovered from the card certification authority certificate CCC, recalculating the hash function for the form data and comparing the recalculated hash function with the one signed with the smart card key SCK and included in the signed data SD.

- 5     Signature verification data SVD derived from the signed data SD is sent by the service provider 20 to the authorisation server 30.

- Card authentication data CAD, which comprises the MAC and the input data used to generate the MAC, and the user identification data ID are transmitted to the service provider 20 and passed to the authentication server
- 10    30. The user identification data ID comprises for example the name, address and date of birth of the user, entered by the user 12 and transmitted by the browser software in a separate field from the signed data SD. Optionally, biometric data from the biometric device 180 is included in the user identification information. Collectively, the signature verification data SVD,
- 15    the card authentication data CAD and the user identification data ID comprise the authorisation request AREQ submitted to the authorisation server 30.

- The authorisation server 30 checks the authorisation request AREQ for counterfeit or replayed cryptograms, checks whether the smartcard public key certificate SCC matches the card identity and checks the user identification
- 20    data ID against an entry in a database of details of known cardholders. The results of these checks are digitally signed using the authorisation server



private key ASK and sent as the authorisation response ARES to the service provider 20.

Some of the above processes will now be explained in greater detail.

## 5 User to Service Provider

Figure 9 shows how the data sent from the user's computer 14 to the service provider 20 is generated. The data D sent from the service provider 20 includes two random or otherwise unpredictable 32-bit numbers UN1 and UN2, the date and time of the transaction and a server identifier SID. The first  
10 unpredictable number is embedded in the HTML form as a readable serial number and is returned in the form data F.

The application software calculates a hash of the completed form data F using a secure hash algorithm SHA and sends the hash to the card 18,  
together with the second unpredictable number UN2, validation result VR, the  
15 date and time and the server identifier SID, for use in the DES encryption process to generate the MAC. The card generates an application transaction counter value ATC which is also input to the DES process. The hash is also supplied as an input to an RSA public key encryption process. The card public  
key certificate SCC is stored in the card 18 and is retrieved by the application  
20 software together with the MAC and signature data SD, for transmission to the service provider 20.

### Signature Validation

The process performed by the service provider 20 to validate the signature of the signed data SD is shown in Figure 10. The service provider 20 receives the form data F and checks (S10) that the serial number UN1 matches that of the blank form previously sent. A hash is calculated from the form data F using the same SHA as performed by the card 18. The card public key is retrieved from the card public key certificate SCC and is used to decrypt (S20) the signature data SD to extract the hash calculated by the user's computer 14. The two hashes are compared (S30) and the signature is validated if they match. However, this process only establishes that the form was digitally signed using the card 18. The service provider 20 must further check that the card 18 is valid and is being used by the authorised cardholder, as will be described below.

### Authorisation Request

The service provider 20 sends the following information to the authorisation server 30 in the authorisation request message ARES, as illustrated in Figure 11: the user identification data ID, including any biometric data, the second unpredictable number UN2, the hash H calculated from the received form data F, the card public key certificate SCC and the message authentication code MAC. The form of the authorisation request message is shown in detail in Table 1 below:

Table 1 - Authorisation Request Message

Field Name	Field Description
Version	Version number of the cryptogram
Service Provider Ref.	Message reference supplied by service provider
Authorisation Service Ref.	Authorisation service reference supplied by the service provider
Contract Info	A URL for a contract governing the use of the authorisation service
Request Sent Time	The time, as supplied by the service provider system clock, at which the message was transmitted
SCC	Public key certificate of the card 18
H	Hash of the form data F
User Time	The time, from the user's computer 14, at which the authorisation request was transmitted from the computer 14
Service Provider Identity	Service provider identity used for generation of the MAC
UN2	Second unpredictable number
Card Data	Data generated by the application and the card 18 and passed to the authorisation server

User Surname	User's surname
User First Name	User's first name(s)
User Title	User's title
User DOB	User's date of birth
User Address	First line of user's address
User Postcode	User's postcode

The Card Data described in Table 1 comprises the fields shown below in Table 2:

Table 2 - Card Data Structure

Field Name	Field Description
Cryptogram Info Data	Indicates the type of the cryptogram
Application Transaction Counter	A counter value, stored in the card 18, which is updated after each transaction
MAC	Cryptogram generated by the card 18
Issuer Application Data	Data determined by the card issuer

5

#### Authorisation Request Process

The authorisation server 30 determines the authorisation response ARES as illustrated in Figure 12. The user identity information ID is sent to

the customer verification server 36, which checks whether this information matches stored user identity information related to the card number, and returns a response IDRES indicating whether these details are correct and the status of the card. The card authentication data CAD are sent to the card authentication server 34 where the MAC is verified using the card number N, 5 extracted from the card public key certificate, the second unpredictable number UN2 and the date, time and server identity originally provided by the service provider 20. The card authentication server 34 also extracts the validation result VR, determines whether the cryptogram has been replayed or 10 the card counterfeited, and whether the card public key certificate SCC is valid and corresponds to the MAC.

Optionally, the customer verification server 36 stores a database of biometric information for each authorised user, and the response IDRES includes information on the confidence level with which biometric data 15 contained within the user identity information ID matches the stored profile for that user.

#### Authorisation Response

The authorisation server 30 formats the authorisation response 20 message ARES and transmits it to the service provider 20 by means of a process illustrated in Figure 13. First, an authorisation message AM is generated including the following information: the hash value H and the user

identification information ID copied from the authorisation request AREQ, and a response code indicating the card authentication and user verification server responses. This message AM is sent to the public key server 32 which generates a digital signature for the message using the authorisation server private key ASK and returns this signature AS together with the authorisation server public key certificate ASC. The authorisation server then sends the authorisation message AM, the signature AS and the authorisation server certificate ASC to the service provider 20, together with a reference code indicating the agreement under which the authorisation is made to the service provider 20.

The data content of the authorisation message is summarised below in Table 3:

Table 3 - Authorisation Message Contents

Field Name	Field Description
Service Provider Reference	Message reference number supplied by the service provider, copied from the authorisation request message
Hash	Hash of the authorisation request message
Request Received Time	The time, from the authorisation server, at which the authorisation request was received
Authorisation Response	Authorisation Response Data
Response Sent Time	Time, from the authorisation server, at which

	the response was sent
--	-----------------------

The Authorisation Response Data is coded as individual bits, as shown in Table 4:

Table 4 - Authorisation Response Data Bit Meaning

Index	Bit No.	Meaning
0	0	Card does not exist
	1	Card not activated
	2	Card expired
	3	Card reported lost
	4	Card reported stolen
	5	Could not verify
	6	Card registered as demo
	7	Verification error - see following bytes
1	0	Unspecified address match failure
	1	Surname did not match
	2	First name did not match
	3	Title did not match
	4	Date of birth did not match
	5	First line of address did not match
	6	Postcode did not match

	7	(Unused)
2	0	Unspecified authentication failure
	1	Card authentication could not be performed
	2	Cryptogram verification failure
	3	Application Transaction Counter value invalid
	4	PIN verification not performed
	5	PIN verification failed

Optionally, the authorisation response data may include an indication of the confidence level with which the biometric data included in the customer identification information matches that stored in the customer verification server 36; for example, pattern matching algorithms used in iris and fingerprint scans will return a suitable value based on the correlation between an input and a reference pattern.

On the basis of the authorisation response ARES, and the result of the signature verification process, the service provider 20 determines how the form data F is to be processed. For example, if the signature is verified and the transaction authorised by the authorisation server, the service provider may update a record corresponding to the user 12 to add the information included in the form data F. If either the signature is not verified, or the transaction not authorised, the form data F may be discarded and/or a message transmitted to the user's computer indicating that the form has not been accepted.



Since the authorisation response ARES is able to indicate the reason for a negative authorisation, the service provider may allow certain types of transaction to proceed if an insignificant authorisation failure is indicated. For example, if the title did not match, the service provider judges this as insignificant, since the user is unambiguously identified by other data, and the transaction is processed as if a positive authorisation were issued by the authorisation server 30.

In the option where the authorisation response ARES includes an indication of the confidence level with which the supplied user identification information ID matches the stored user identification information, the service provider 20 sets a minimum confidence level for the current transaction and allows the transaction to proceed if this confidence level is exceeded. Preferably, this minimum confidence level is determined according to the monetary value of the transaction, or if the transaction does not have a specified monetary value, the consequences of fraud in that transaction.

Preferably, the authorisation response ARES is used to determine the liability in the case of fraud between the operator of the service provider 20 and the operator of the authorisation server 30. For example, if any of the bits with index 0 are set, the card system operator will not accept any liability if the service provider accepts the transaction, but if only one of the bits with index 1 is set, the card system operator will accept liability only to a

prearranged limit, and if none of the bits is set, the card system operator will accept liability to the maximum value prearranged for the user 12.

The present invention is not limited to the use of the Internet but may also be applied to transactions over other networks which are not inherently  
5 secure. The network used to connect the user's computer 14 to the service provider 20 may be separate from that used to connect the service provider 20 to the authorisation server 30.

Although the above embodiment is described with reference to one specific user, it is evident that similar procedures are carried out for different  
10 users so that the system can perform transactions initiated by any one of a large number of users.

The present invention is not limited to any specific type of transaction such as the authentication of forms or the authorisation of payment.

In an alternative embodiment, the smart card 18 may sign the form  
15 data using the symmetric encryption key under the DES process and may dispense with the RSA encryption process. The service provider 20 will not then be able to verify the signature, but instead recalculates the hash and transmits this to the authorisation server 30 for verification.

In an alternative embodiment, the smart card 18 uses the private key  
20 SCK to produce the MAC and does not use any symmetric encryption key. In that case, the service provider 20 can verify the authenticity of the MAC.

The above embodiments are described by way of example and are not to be construed as limiting the scope of the invention. Instead, the invention extends to all variants which fall within the scope of the following claims.

CLAIMS

1. A method of processing a data transaction between a terminal and a first server over a public network and between the first server and a second server, comprising:
  - 5 generating, at the terminal, terminal encrypted data derived from transaction identification information uniquely identifying a current transaction, using a first key;
  - transmitting said terminal encrypted data from said terminal to said first server over said public network;
  - 10 transmitting said terminal encrypted data and said transaction identification information from said first server to said second server;
  - decrypting said terminal encrypted data using a second key;
  - comparing said decrypted data with said transaction identification information received from said first server by said second server;
  - 15 and transmitting a transaction authorisation message from said second server to said first server, the content of the authorisation message being dependent on the result of said comparison.
2. A method as claimed in claim 1, including inputting user identification
  - 20 information from a user at said terminal,

wherein the user identification information is transmitted together with said terminal encrypted data to said first server and from said first server to said second server, the method further comprising:

5       comparing the received user identification information at said second server with previously stored user identification information, the content of the authorisation message being dependent on said comparison of said user identification information.

10       3.     A method as claimed in claim 1 or claim 2, wherein said first and second keys are symmetric encryption keys.

4.     A method as claimed in any preceding claim, wherein said authorisation message is signed by the second server using a third key.

15       5.     A method as claimed in any preceding claim, further comprising:  
           inputting transaction message data at said terminal by said user;  
           signing said transaction message data at said terminal using a fourth key to generate transaction signature data;  
           transmitting said transaction message data and said transaction  
 20       signature data from said terminal to said first server; and  
           verifying said transaction signature data against said transaction message data using a fifth key at said first server.

6. A method as claimed in any preceding claim, including supplying said authorisation message and transaction identification information from said first server to data storage means.

5

7. A method of processing a data transaction between a terminal and a first server over a public network and between the first server and a second server, comprising:

transmitting transaction message data and identification data from said  
10 terminal to said first server;

transmitting said identification data from said first server to said  
second server;

comparing said received identification data at said second server with  
previously stored identification data and generating an authorisation message  
15 indicating either no authorisation, partial authorisation or full authorisation  
dependent on said comparison;

transmitting said authorisation message from said second server to  
said first server;

and processing said transaction data at said first server according to the  
20 received authorisation message.

8. A method as claimed in claim 7, wherein said transaction data is processed at said first server further according to the content of the transaction data.

5 9. A method as claimed in claim 8, wherein the processing step comprises:

determining a threshold value from the authorisation message;

determining a transaction value from the transaction data;

10 and processing the transaction data as valid if the transaction value does not exceed the threshold value.

10. A method as claimed in any one of claims 7 to 9, wherein the authorisation message is signed by the second server using a first key, and the signed authorisation message is verified by the first server using a second key,  
15 the processing of the transaction data being dependent on said verification by the first server.

11. A method of authenticating a data transaction between a user terminal and a first server over a public network, comprising:  
20 issuing an authentication token to a user, said authentication token storing authentication information;

verifying the identity of the user and of the token and transmitting a verification message to a second server;

storing status information at said second server in response to said verification message;

5           inputting user identification information and transaction data at a user terminal;

connecting said authentication token to said terminal and retrieving said authentication information therefrom;

10           transmitting said user identification information, authentication information and transaction data from the user terminal over a public network to the first server;

transmitting said user identification information and authentication information from said first server to said second server;

15           comparing said received user identification data with previously stored identification data;

retrieving said status information corresponding to said authentication information;

20           and transmitting an authorisation message dependent on the result of said comparison from the second server to the first server and on said status information.



12. A method as claimed in any preceding claim, wherein said transmissions between the first server and the second server are performed over said public network.

5 13. A method as claimed in any preceding claim, wherein said public network is an internet.

14. A method as performed by the first server in a method as claimed in any preceding claim.

10

15. A method as performed by the second server in a method as claimed in any one of claims 1 to 13.

16. Apparatus arranged to perform the method as claimed in claim 14.

15

17. Apparatus arranged to perform the method as claimed in claim 15.

18. A method substantially as herein described with reference to the accompanying drawings.



**Application No:** GB 9812520.6  
**Claims searched:** 1-18

**Examiner:** B.J.SPEAR  
**Date of search:** 10 December 1998

**Patents Act 1977**  
**Search Report under Section 17**

**Databases searched:**

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:  
 UK CI (Ed.P): H4P(PDCSA,PDCSC)  
 Int CI (Ed.6): H04L 9/32  
 Other: Online:WPI,EPODOC,PAJ

**Documents considered to be relevant:**

Category	Identity of document and relevant passage	Relevant to claims
X	EP0782296A2 (NCR) Whole document, eg Fig.8 and p 21 50 to p 31 45 p 51 37-48.	1-3,7,11-17 at least
X	WO97/41539A1 (Verifone) Whole document, eg Figs. 1B, 4,6A,6B, abstract and p 191 39 to p 241 17.	1-3,7,11-17 at least
X	WO97/23972A1 (V-ONE) Whole document, eg Fig. 1,abstract, and p 31 18 to p 41 8, p 71 2 to p 81 26, p 121 8-19	1-3,7,11-17 at least
X	US5615268 (Document Authentication) Whole document, eg Fig.3 and col. 61 11 to col. 71 33.	1-3,7,11-17 at least
X	US5602918 (Virtual Open) Whole document, eg Fig. 1 and col. 31 31 to col. 51 44	1-3,7,11-17 at least
X	US5590197 (V-One) Whole document, eg Fig. 1 and col. 41 42 to col. 71 8.	1-3,7,11-17 at least
X	Internet Cryptography by Richard E Smith Pub 1997 Addison Wesley ISBN 0-201-92480-3 eg pages 113-116, 262-265	1-3,7,11-17 at least

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



(11) **EP 1 237 324 A1**

(12) **EUROPEAN PATENT APPLICATION**  
published in accordance with Art. 158(3) EPC

Exhibit E10

EP 3 229 099  
(17000713.2)

BARDEHLE PAGENBERG  
Our ref.: S154820EPT1EIN

(43) Date of publication:  
**04.09.2002 Bulletin 2002/36**

(51) Int Cl.7: **H04L 9/10, G06F 12/14,  
G10K 15/02, G06F 13/00**

(21) Application number: **00978073.5**

(86) International application number:  
**PCT/JP00/08544**

(22) Date of filing: **01.12.2000**

(87) International publication number:  
**WO 01/041356 (07.06.2001 Gazette 2001/23)**

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

(30) Priority: **02.12.1999 JP 34338999**

(71) Applicants:  

- **Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka-fu 570-8677 (JP)**
- **PFU LIMITED**  
**Kahoku-gun, Ishikawa 929-1125 (JP)**
- **FUJITSU LIMITED**  
**Kawasaki-shi, Kanagawa 211-8588 (JP)**
- **Hitachi, Ltd.**  
**Chiyoda-ku, Tokyo 101-8010 (JP)**
- **Nippon Columbia Co., Ltd.**  
**Tokyo 107-8011 (JP)**

(72) Inventors:  

- **HORI, Yoshihiro, Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka 570-8677 (JP)**
- **HIOKI, Toshiaki, Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka 570-8677 (JP)**

- **KANAMORI, Miwa, Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka 570-8677 (JP)**
- **YOSHIKAWA, Takatoshi,**  
**Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka 570-8677 (JP)**
- **TAKEMURA, Hiroshi, Sanyo Electric Co., Ltd.**  
**Moriguchi-shi, Osaka 570-8677 (JP)**
- **TAKAHASHI, Masataka, PFU Limited**  
**Kahoku-gun, Ishikawa 929-1125 (JP)**
- **HASEBE, Takayuki, Fujitsu Limited**  
**Kawasaki-shi, Kanagawa 211-8588 (JP)**
- **FURUTA, Shigeki, Fujitsu Limited**  
**Kawasaki-shi, Kanagawa 211-8588 (JP)**
- **HATAKEYAMA, Takahisa, Fujitsu Limited**  
**Kawasaki-shi, Kanagawa 211-8588 (JP)**
- **TONEGAWA, Tadaaki,**  
**Semicond. & Integr. Circuits**  
**Kodaira-shi, Tokyo 187-8588 (JP)**
- **ANAZAWA, Takeaki, Nippon Columbia Co., Ltd.**  
**Tokyo 107-8011 (JP)**

(74) Representative: **Glawe, Delfs, Moll & Partner**  
**Patentanwälte**  
**Postfach 26 01 62**  
**80058 München (DE)**

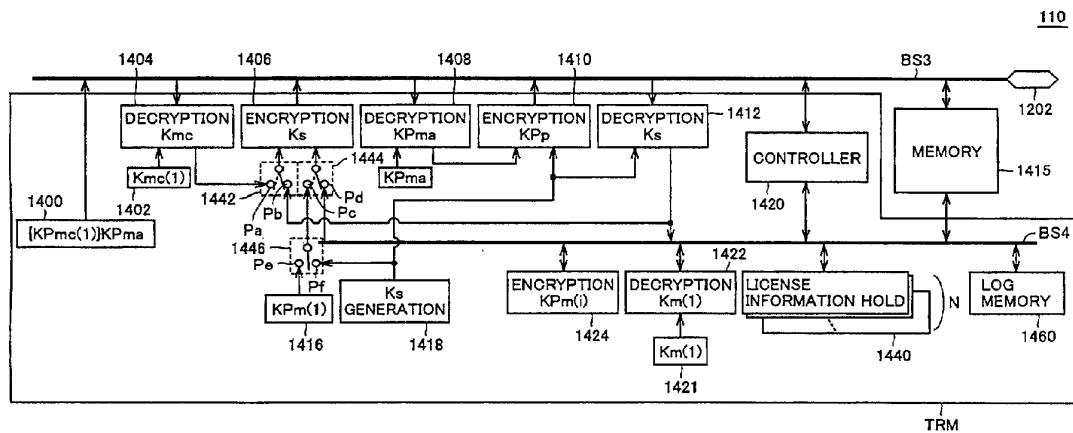
(54) **MEMORY CARD AND DATA DISTRIBUTION SYSTEM USING IT**

(57) A memory card (110) conducts an authentication process with a server based on data stored in an authentication data hold unit (1400). The memory card (110) extracts a first session key (Ks1) from a server by a decryption process and a transaction ID from the data applied on a data bus (BS3). The memory card (110) generates a second session key (Ks2) through a session key generation unit (1418), and transmits to the

server, as the keys to encrypt content data in receiving decryption of content data, the second session key (Ks2) and a key (KPm(1)) unique to the memory card (110) in an encrypted state with the first session key (Ks1). The transaction ID and the second session key (Ks2) stored in the log memory (1460) are used in the redistribution process.

EP 1 237 324 A1

496



## Description

Memory Card and Data Distribution System Using Such Memory Card

### Technical Field

**[0001]** The present invention relates to a memory card that allows protection on copyrights with respect to copied information in an information distribution system to distribute information to terminals such as cellular phones, and a distribution system using such a memory card.

### Background Art

**[0002]** By virtue of the progress in information communication networks and the like such as the Internet in these few years, each user can now easily access network information through individual-oriented terminals employing a cellular phone or the like.

**[0003]** In such information communication, information is transmitted through digital signals. It is now possible to obtain copied music and video information transmitted via the aforementioned information communication network without degradation in the audio quality and picture quality of the copy data, even in the case where the copy operation is performed by an individual user.

**[0004]** Thus, there is a possibility of the copyright of the copyright owner being significantly infringed unless some appropriate measures to protect copyrights are taken when any content data subject to copyright protection such as music and image information is to be transmitted on the information communication network.

**[0005]** However, if copyright protection is given top priority so that distribution of content data through the disseminating digital information communication network is suppressed, the copyright owner who can essentially collect a predetermined copyright royalty for copies of a copyrighted work will also incur some disadvantage.

**[0006]** In the case where content data such as music data is distributed through a digital information communication network as described above, each user will record the distributed data onto some recording apparatus, and then reproduce the data using a reproduction apparatus.

**[0007]** Such recording apparatuses include, for example, a medium that can have data written and erased electrically such as a memory card.

**[0008]** As the apparatus to reproduce distributed music data, the cellular phone per se used to receive such data distribution can be employed, or when the recording apparatus such as a memory card is detachable from the apparatus that receives distribution, a dedicated reproduction apparatus can be used.

**[0009]** In the case where distribution of content data such as music data is to be received through a digital

information communication network, particularly through a radio communication network, the communication may be cut off before the music data is completely distributed depending upon the state of the communication line. In the case where encrypted content data which is an encrypted version of content data is decrypted and reproduction information required for reproduction is to be distributed, any disruption in communication during distribution of the encrypted content data can be mended by establishing connection again and continuing data reception. Since the accounting process towards the user is carried out simultaneously in distributing reproduction information, the user will request retransmission of the reproduction information after connection is established again with respect to such disrupted communication. However, reproduction information should not be retransmitted incautiously in response to a request from the standpoint of protecting the rights of copyright owners. However, if retransmission is not conducted, the user will not be able to obtain the reproduction information even though the accounting process has been effected.

### Disclosure of the Invention

**[0010]** An object of the present invention is to provide a data distribution system that can complete distribution of reproduction information even in the case where communication is disrupted before complete distribution of reproduction information by resuming communication upon protecting the rights of copyright owners, and a memory card used in such a data distribution system.

**[0011]** A memory card of the present invention to achieve the above object receives and records reproduction information associated with reproduction of encrypted content data, including a content key to decrypt the encrypted content data into plaintext, through a communication path. The memory card includes a data communication unit, a first storage unit, an information extraction unit, a second storage unit and a control unit.

**[0012]** The data communication unit establishes a communication path with the transmission source of the reproduction information to receive the reproduction information transmitted in an encrypted state. The first storage unit stores data associated with the reproduction information applied from the data communication unit. The information extraction unit carries out the process of storing the data associated with the reproduction information from the data communication unit into the first storage unit, and extracting reproduction information based on data stored in the first storage unit. The second storage unit records a reception log indicating the processing status of the reproduction information transmission process. The control unit controls the operation of the memory card. The control unit controls transmission of the reception log to the transmission source in response to a request.

**[0013]** Preferably, the data communication unit in-

cludes a first key hold unit, a first decryption processing unit, a second key hold unit, a key generation unit, a first encryption processing unit, and a second decryption processing unit. The first key hold unit stores a first secret encryption key to decrypt data that is encrypted using a predetermined first public encryption key corresponding to a memory card. The first decryption processing unit receives and decrypts a first symmetric key that is updated and transmitted for each communication of the reproduction information, and encrypted using the first public encryption key. The second key hold unit stores a second public encryption key differing for each memory card. The key generation unit generates a second symmetric key updated for each communication of the reproduction information. The first encryption processing unit encrypts the second public encryption key and second symmetric key based on the first symmetric key for output. The second decryption processing unit receives reproduction information encrypted with the second public encryption key and further encrypted with a second symmetric key, and decrypts the reproduction information based on the second symmetric key. The first storage unit stores data based on the output of the second decryption processing unit. The information extraction unit includes a third key hold unit and a third decryption processing unit. The third key hold unit stores a second private decryption key to decrypt data encrypted by the second public encryption key. The third decryption processing unit carries out a decryption process for the second private decryption key in the procedure from the process of storing data associated with reproduction information into the first storage unit to the process of extracting reproduction information.

**[0014]** According to another aspect of the present invention, a data distribution system includes a content data supply apparatus and a plurality of terminals.

**[0015]** The content data supply apparatus supplies encrypted content data, and reproduction information including a content key which is a decryption key associated with reproduction of encrypted content data and used to decrypt the encrypted content data into plaintext. The content data supply apparatus includes a distribution information hold unit, a first interface unit, a first session key generation unit, a session key encryption unit, a session key decryption unit, a first license data encryption processing unit, a second license data encryption processing unit, and a distribution log information hold unit. The distribution information hold unit stores content data and reproduction information. The first interface unit transmits/receives data to/from an external source. The first session key generation unit generates a first symmetric key updated for each distribution of reproduction information to a terminal. The session key encryption unit encrypts and provides to the first interface unit a first symmetric key using a first public encryption key predefined corresponding to a user's terminal. The session key decryption unit decrypts the

second public encryption key and second symmetric key transmitted in an encrypted state by a first symmetric key. The first license data encryption processing unit encrypts reproduction information to reproduce encrypted content data using a second public encryption key encrypted by a session key decryption unit. The second license data encryption processing unit encrypts the output of the first license data encryption processing unit using a second symmetric key, and applies the encrypted output to the first interface unit for distribution. The distribution log information hold unit records a distribution log indicating the processing status of the current distribution process. The plurality of terminals receive distribution through a communication path from the content data supply apparatus, and correspond to a plurality of users, respectively. Each terminal includes a second interface unit, a reception control unit, and a data storage unit. The second interface unit transmits/receives data to/from an external source. The reception control unit controls the data transfer with an external source. The data storage unit receives and stores encrypted content data and reproduction information. The data storage unit includes a first key hold unit, a first decryption processing unit, a second key hold unit, a key generation unit, a first encryption processing unit, a second decryption processing unit, a first storage unit, a third key hold unit, a third decryption processing unit, and a second storage unit. The first key hold unit stores a first secret encryption key to decrypt data that is encrypted with a predetermined first public encryption key corresponding to the data storage unit. The first decryption processing unit receives a first symmetric key that is updated and distributed for each communication of the reproduction information, and encrypted using the first public encryption key, and applies a decryption process. The second key hold unit stores a second public encryption key differing for each data storage unit. The key generation unit generates a second symmetric key updated for each communication of the reproduction information. The first encryption processing unit encrypts and outputs the second public encryption key and second symmetric key based on the first symmetric key. The second decryption processing unit receives reproduction information encrypted with the second public encryption key and further encrypted with the second symmetric key, and decrypts the reproduction information based on the second symmetric key. The first storage unit stores data based on the output of the second decryption processing unit. The third key hold unit stores a second private decryption key to decrypt data encrypted with the second public encryption key. The third decryption processing unit applies a decryption process for the second private decryption key in the procedure of the process of storing data associated with reproduction information to the first storage unit to the process of extracting reproduction information. The second storage unit records a reception log indicating the processing status in the distribution process of encrypted content

data and reproduction information. The reception control unit controls the redistribution process based on the reception log when the communication path is cut off during a distribution process.

**[0016]** Both the server and memory card store the distribution history and distribution status in a distribution system using the data reproduction apparatus and a memory card employed in the distribution system of the present invention. Therefore, information can be retransmitted by resuming communication even in the case where communication is disrupted during distribution. The reliability of the distribution process can be improved.

#### Brief Description of the Drawings

#### **[0017]**

Fig. 1 is a diagram to schematically describe an entire structure of a data distribution system of the present invention.

Fig. 2 is a diagram to describe the characteristics of data and information used in communication in the data distribution system of Fig. 1.

Fig. 3 is a schematic block diagram showing a structure of a license server 10.

Fig. 4 is a schematic block diagram showing a structure of a cellular phone 100.

Fig. 5 is a schematic block diagram showing a structure of a memory card 110.

Fig. 6 is a first flow chart to describe a distribution operation in the data distribution system of a first embodiment.

Fig. 7 is a second flow chart to describe a distribution operation in the data distribution system of the first embodiment.

Fig. 8 is a third flow chart to describe a distribution operation in the data distribution system of the first embodiment.

Fig. 9 is a flow chart to describe a reconnection process.

Fig. 10 is a first flow chart to describe a second reconnection operation of the data distribution system according to the first embodiment.

Fig. 11 is a second flow chart to describe a second reconnection operation of the data distribution system according to the first embodiment.

Fig. 12 is a third flow chart to describe a second reconnection operation of the data distribution system according to the first embodiment.

Fig. 13 is a flow chart to describe a third reconnection operation of the data distribution system according to the first embodiment.

Fig. 14 is a flow chart to describe a reconnection process.

Fig. 15 is a first flow chart to describe a distribution operation in the event of purchasing content in the data distribution system according to a second em-

bodiment.

Fig. 16 is a second flow chart to describe a distribution operation in the event of purchasing content in the data distribution system according to the second embodiment.

Fig. 17 is a third flow chart to describe a distribution operation in the event of purchasing content in the data distribution system according to the second embodiment.

Fig. 18 is a first flow chart to describe a second reconnection operation of the data distribution system of the second embodiment.

Fig. 19 is a second flow chart to describe a second reconnection operation of the data distribution system of the second embodiment.

Fig. 20 is a third flow chart to describe a second reconnection operation of the data distribution system of the second embodiment.

Fig. 21 is a first flow chart to describe a second reconnection operation of the data distribution system according to a third embodiment of the present invention.

Fig. 22 is a second flow chart to describe a second reconnection operation of the data distribution system according to the third embodiment of the present invention.

Fig. 23 is a third flow chart to describe a second reconnection operation of the data distribution system according to the third embodiment of the present invention.

Fig. 24 is a fourth flow chart to describe a second reconnection operation of the data distribution system according to the third embodiment of the present invention.

#### Best Modes for Carrying Out the Invention

**[0018]** Embodiments of the present invention will be described hereinafter with reference to the drawings.

#### [First Embodiment]

**[0019]** Fig. 1 is a diagram to describe schematically an entire structure of the data distribution system of the present invention.

**[0020]** In the following, a data distribution system distributing music data to each user via a cellular phone network will be described as an example. However, as will become apparent from the following description, the present invention is not limited to such a case. The present invention is applicable to distribute content data corresponding to other copyrighted works such as book telling data, image data, video data, educational data, and the like, and further applicable to the case of distributing through other digital information communication networks.

**[0021]** Referring to Fig. 1, a license server 10 administering copyrighted music data encrypts music data

(also called "content data" hereinafter) according to a predetermined encryption scheme, and provides such encrypted content data to a cellular phone company which is a distribution carrier 20 to distribute information. An authentication server 12 challenges the authenticity of the user's apparatus establishing access for distribution of content data.

**[0022]** Cellular phone company 20 relays a distribution request from each user to license server 10 through its own cellular phone network. In response to a distribution request, license server 10 verifies the authenticity of the user's apparatus through authentication server 12, and distributes content data to respective user's cellular phone via the cellular phone network of cellular phone company 20 after the requested music data has been further encrypted.

**[0023]** Fig. 1 corresponds to a structure in which a detachable memory card 110 is loaded in a cellular phone 100 of a user 1. Memory card 110 receives the encrypted content data through cellular phone 100 and applies decryption on the above encryption, and then provides the decrypted data to the music reproduction unit (not shown) in cellular phone 100.

**[0024]** User 1, for example, can "reproduce" the music data to listen to the music via a headphone 130 or the like connected to cellular phone 100.

**[0025]** License server 10, authentication server 12, and distribution carrier (cellular phone company) 20 will generically be referred to as a distribution server 30 hereinafter.

**[0026]** The process of transmitting content data to each cellular phone or the like from distribution server 30 is called "distribution".

**[0027]** By such a structure, any user that has not purchased a memory card 110 cannot receive and reproduce distribution data from distribution server 30.

**[0028]** By taking count of the number of times content data of, for example, one song, is distributed in distribution carrier 20, the copyright royalty fee induced every time a user receives (downloads) content data can be collected by distribution carrier 20 in the form of telephone bills of respective cellular phones. Thus, the royalty fee of the copyright owner can be ensured.

**[0029]** Furthermore, since such content data distribution is conducted through a cellular phone network, which is a closed system, there is the advantage that measures to protect copyrights can be taken more easily than compared to an open system such as the Internet.

**[0030]** Here, a user 2 possessing a memory card 112, for example, can directly receive distribution of content data from distribution server 30 through his/her own cellular phone 102. However, direct reception of content data or the like from music server 30 is relatively time consuming for user 2 since the content data includes a large amount of information. In such a case, it will be convenient for the user if content data can be copied from user 1 that has already received distribution of that

content data.

**[0031]** However, from the standpoint of protecting the rights of copyright owners, unscrupulous copying of content data is not allowed on the basis of system configuration.

**[0032]** As shown in Fig. 1, the act of letting a user 2 copy the content data received by user 1, and transferring together the reproduction information to render the relevant content data reproducible to user 2 is called "transfer" of music data. In this case, the encrypted content and reproduction information required for reproduction are transferred between memory cards 110 and 112 through cellular phones 100 and 102. As will be described afterwards, "reproduction information" includes a license key that allows decryption of content data encrypted according to a predetermined encryption scheme, and license information such as information of restriction as to access reproduction and a license ID corresponding to information related to copyright protection.

**[0033]** In contrast, the act of copying only content data without transferring reproduction information is called "replicate". Since reproduction information is not transferred in replication, the user receiving this replication can render the data reproducible by requesting distribution of only the reproduction information. Accordingly, distribution of a significant amount of data can be eliminated in the distribution of content data.

**[0034]** By such a structure, the content data distributed by the distribution server can be used flexibly at the reception side.

**[0035]** In the case where cellular phones 100 and 102 are PHSs (Personal Handy Phones), information can be transferred between user 1 and user 2 taking advantage of conversation in the so-called available transceiver mode.

**[0036]** In the structure shown in Fig. 1, the system to render the content data distributed in an encrypted manner reproducible at the user side requires: 1) the scheme to distribute an encryption key in communication, 2) the scheme per se to encrypt distribution data, and 3) a configuration realizing data protection to prevent unauthorized copying of the distributed data.

**[0037]** In the embodiment of the present invention, a distribution system that records and stores the status and history of distribution at both the information transmission side and reception side, and that allows retransmission of information by resuming communication even when communication is disrupted during distribution to improve reliability of the distribution process will be described.

#### [System Key and Data Configuration]

**[0038]** Fig. 2 is a diagram to describe the characteristics of the keys associated with encryption used in communication and data to be distributed in the data distribution system of Fig. 1.



**[0039]** The data "Data" distributed by distribution server 30 is content data such as music data. The content data is distributed to a user from distribution server 30 in the form of encrypted content data {Data}Kc subject to encryption that can be decrypted using at least a license key Kc.

**[0040]** In the following, the expression of {Y}X implies information having data Y converted into encryption that can be decrypted using a key X.

**[0041]** From the distribution server are distributed additional information Data-inf in plaintext such as the information related to content data or related to server access and the like, together with the content data. Specifically, additional information Data-inf includes information to identify the content data such as the song title or the name of the artist and to identify distribution server 30.

**[0042]** Keys related to the encryption, decryption and reproduction process of content data as well as to authentication of a cellular phone which is the content reproduction circuit and a memory card which is a recording apparatus are set forth below.

**[0043]** As mentioned before, there are provided a license key Kc used to decrypt encrypted content data, a public encryption key K<sub>Pp</sub>(n) unique to the content reproduction circuit (cellular phone 100), and a public encryption key K<sub>Pmc</sub>(m) unique to a memory card.

**[0044]** Data encrypted using public encryption keys K<sub>Pp</sub>(n) and K<sub>Pmc</sub>(m) can be decrypted respectively using a secret encryption key K<sub>p</sub>(n) unique to the content reproduction circuit (cellular phone 100) and a private decryption key K<sub>mc</sub>(m) unique to the memory card. These unique private decryption keys having different contents for each type of cellular phone and each type of memory card. Here the type of cellular phone and memory card is defined based on the manufacturer thereof, the fabrication time (fabrication lot) and the like. The unit assigned to the public secret key and private decryption key is referred to as "class." Natural numbers m, n represent the numbers to discriminate the class of each memory card and content reproduction circuit (cellular phone).

**[0045]** Keys operated common to the entire distribution system include a secret common key Kcom used in obtaining license key Kc and restriction information for the reproduction circuit that will be described afterwards, and an authentication key KPma. Secret common key Kcom is stored at both the distribution sever and the cellular phone.

**[0046]** Public encryption keys K<sub>Pmc</sub>(m) and K<sub>Pp</sub>(n) specified for each memory card and content reproduction circuit can have their authenticity verified by decrypting with authentication key KPma. More specifically, they are recorded in respective memory cards and cellular phones at the time of shipment in the form of authentication data {K<sub>Pmc</sub>(n)}KPma and {K<sub>Pp</sub>(n)}KPma subject to the authentication process.

**[0047]** Secret common key Kcom is not restricted to

be in the symmetric key cryptosystem. It can be replaced with the private decryption key or public encryption key KPcom in the public key cryptosystem. In this case, private key Kcom and public key KPcom are held in cellular phone 100 and distribution server 30, respectively, as secret common key.

**[0048]** Information to control the operation of the apparatus constituting the system, i.e. cellular phone 100 which is a content reproduction circuit and memory card 110, includes purchase condition information AC transmitted from cellular phone 100 to distribution server 30 when a user purchases a license key or the like for the purpose of specifying the purchase condition, access restriction information AC1 distributed from distribution server 30 towards memory card 110 according to purchase condition information AC, indicating the number of times of accessing license key Kc for reproduction (reproduction permitted times), the number of replicates and transfer of license key Kc, and restriction as to copy and transfer, and reproduction circuit restriction information AC2 distributed from distribution server 30 to memory card 110, indicating restriction as to the reproduction condition of the content reproduction circuit, loaded in cellular phone 100. The reproduction condition of the reproduction circuit implies the condition, for example, of allowing reproduction of only the beginning of each content data for a predetermined time such as in the case where a sample is distributed at low price or freely to promote a new song, the reproduction period and the like.

**[0049]** The keys to administer data processing in memory card 100 includes a public encryption key K<sub>Pm</sub>(i) (i: natural number) specified for each memory card, and a private decryption key K<sub>m</sub>(i) unique to each memory card that can decrypt data encrypted with public encryption key K<sub>Pm</sub>(i). Here, natural number i represents a number to discriminate each memory card.

**[0050]** In the data distribution system of Fig. 1, keys used in data communication are set forth below.

**[0051]** The key to ensure security during data transfer with an external source to the memory card or between memory cards includes symmetric keys Ks1-Ks4 generated at server 30, cellular phone 100 or 102, and memory card 110 or 112 every time content data distribution, reproduction or transfer is carried out.

**[0052]** Here, symmetric keys Ks1-Ks4 are unique symmetric keys generated for each "session" which is the access unit or communication unit among the server, content reproduction circuit or memory card. In the following, these symmetric keys Ks1-Ks4 are also called "session keys".

**[0053]** These session keys Ks1-Ks4 have a unique value for each communication session, and is under control of the distribution server, content reproduction circuit and memory card.

**[0054]** More specifically, a session key Ks1 is generated for each distribution session by distribution server 30. A session key Ks2 is generated for each distribution

session and transfer (reception side) session of a memory card. Session key Ks3 is generated for each reproduction session and transfer (transmission side) session in a memory card. A session key Ks4 is generated for each reproduction session of the cellular phone. The level of security can be improved in each session by transferring the session keys and receiving a session key generated by another apparatus to perform encryption using the session keys and transmitting the license decryption key.

**[0055]** Data transferred with a distribution server includes a content ID for the system to identify each content data, and a transaction ID which is a code generated for each distribution session to identify each distribution session. It is to be noted that the license ID and transaction ID can be shared.

**[0056]** The license ID, content ID and access restriction information AC1 are generically referred to as license information. This license information, license key Kc and reproduction circuit restriction information AC2 are generically referred to as reproduction information.

[Configuration of License Server 10]

**[0057]** Fig. 4 is a schematic block diagram showing a structure of license server 10 of Fig. 1.

**[0058]** License server 10 includes an information database 304 to store content data encrypted according to a predetermined scheme as well as distribution information such as a license ID, an account database 302 to store accounting data according to the start of access to content data for each user, a log administration database 306 to store log information of the license server, a data processing unit 310 receiving data through a data bus BS1 from information database 304, accounting database 302 and log administration database 306 to apply a predetermined process, and a communication device 350 to transfer data between distribution carrier 20 and data processing unit 310 via the communication network.

**[0059]** "License distribution log" indicating the distribution history of the license information stored in log administration database 306 includes the transaction ID, content ID, public encryption key KPmc(n), KPP(n), access restriction information AC1, reproduction circuit restriction information AC2, public encryption key KPM(i), session key Ks2, and an accounting status flag. The accounting status flag indicates whether the accounting process for the currently-distributed content data has already ended or not.

**[0060]** Data processing unit 310 includes a distribution control unit 315 to control the operation of data processing unit 310 according to the data on data bus BS1, a session key generation unit 316 to generate a session key Ks1 in a distribution session, under control of distribution control unit 315, a decryption processing unit 312 receiving through communication device 350 and data bus BS1 authentication data {KPmc(n)}KPma

and {KPP(n)}KPma sent from a memory card and a cellular phone to apply a decryption process on authentication key KPma, an encryption processing unit 318 encrypting session key Ks1 generated by session key generation unit 316 using public encryption key KPmc(m) obtained by decryption processing unit 312 to provide the encrypted key onto data bus BS1, and a decryption processing unit 320 receiving through data bus BS1 the data encrypted with session key Ks1 and transmitted by each user.

**[0061]** Data processing unit 310 further includes a Kcom hold unit 322 storing secret common key Kcom, an encryption processing unit 324 encrypting license key Kc and reproduction circuit restriction information AC2 applied from distribution control unit 315 using secret common key Kcom, an encryption processing unit 326 to encrypt the data output from encryption processing unit 324 using a public encryption key KPM(i) unique to the memory card obtained from decryption processing unit 320, and an encryption processing unit 328 further encrypting the output of encryption processing unit 326 using a session key Ks2 applied from decryption processing unit 320 to provide the encrypted key onto data bus BS1.

**[0062]** In the case where secret common key Kcom is the key of an asymmetric public key cryptosystem, Kcom hold unit 322 stores public key Kpcom, which is the encryption key in the public key cryptosystem, instead of secret common key Kcom in the symmetric key cryptosystem.

[Configuration of Cellular Phone 100]

**[0063]** Fig. 4 is a schematic block diagram to describe a structure of a cellular phone 100 of Fig. 1.

**[0064]** In cellular phone 100, the natural number n representing the class is set to n = 1.

**[0065]** Cellular phone 100 includes an antenna 1102 to receive a signal transmitted through radio by a cellular phone network, a transmitter/receiver unit 1104 converting the signal received from antenna 1102 into a base band signal, or modulating and providing to antenna 1102 the data from a cellular phone, a data bus BS2 to transfer data between respective components of cellular phone 100, and a controller 1106 to control the operation of cellular phone 100 via data bus BS2.

**[0066]** Cellular phone 100 further includes a key board 1108 to apply designation to cellular phone 100 from an external source, a display 1110 to apply the information output from controller 1106 or the like to the user as visual information, an audio reproduction unit 1112 reproducing audio based on reception data provided via data bus BS2 in a general conversation operation, a connector 1120 to transfer data with an external source, and an external interface unit 1122 providing the data from connector 1120 to data bus BS2 for conversion, or to convert the data from data bus BS2 into a signal that can be applied to connector 1120.

[0067] Cellular phone further includes a detachable memory card 110 storing content data (music data) for a decryption process, a memory interface 1200 to control data transfer between memory card 110 and data bus BS2, and an authentication data hold unit 1500 storing a public encryption key  $KPp(1)$  set for each cellular phone class in an encrypted state that can be authenticated by decryption using authentication key  $KPma$ .

[0068] Cellular phone 100 further includes a  $Kp$  hold unit 1502 storing private decryption key  $Kp(n)$  ( $n=1$ ) which is a encryption key unique to the cellular phone (content reproduction circuit) class, a decryption processing unit 1504 decrypting the data received from data bus BS2 using private decryption key  $Kp(1)$ , and obtaining session key  $Ks3$  generated by the memory card, a session key generation unit 1508 generating using a random number a session key  $Ks4$  used to encrypt data transferred on data bus BS2 with memory card 110 in a session of reproducing content data stored in memory card 110, an encryption processing unit 1506 encrypting generated session key  $Ks4$  using a session key  $Ks3$  obtained by decryption processing unit 1504, and a decryption processing unit 1510 decrypting the data on data bus BS2 using session key  $Ks4$  to output data  $\{Kc//AC2\}Kcom$ .

[0069] Cellular phone 100 further includes a  $Kcom$  hold unit 1512 storing a secret common key  $Kcom$ , a decryption processing unit 1514 decrypting data  $\{Kc//AC2\}Kcom$  output from decryption processing unit 1510 using secret common key  $Kcom$  to output license key  $Kc$  and reproduction circuit restriction information  $AC2$ , a decryption processing unit 1516 receiving encrypted content data  $\{Data\}Kc$  from data bus BS2 to decrypt the data using license key  $Kc$  obtained by decryption processing unit 1510 to output content data  $Data$ , a music reproduction unit 1518 to receive content data  $Data$  which is the output of decryption processing unit 1516 to reproduce content data, a switch unit 1525 receiving the outputs of music reproduction unit 1518 and audio reproduction unit 1112 to selectively provide an output according to the operation mode, and a connection terminal 1530 receiving the output of mixer unit 1525 for connection to headphone 130.

[0070] Here, reproduction circuit restriction information  $AC2$  output from decryption processing unit 1514 is applied to controller 1106 via data bus BS2.

[0071] In Fig. 4, only the blocks associated with distribution and reproduction of music data among the blocks forming the cellular phone are illustrated for the sake of simplification. Blocks related to the general conversation function inherent to a cellular phone are left out.

[Configuration of Memory Card 110]

[0072] Fig. 5 is a schematic block diagram to describe a structure of memory card 110 of Fig. 1.

[0073] As described before, public encryption key

$KPm(i)$  and a corresponding private decryption key  $Km(i)$  take unique values for each memory card. In memory card 110, it is assumed that the natural number is set to  $i = 1$ . Also,  $KPmc(m)$  and  $Kmc(m)$  are set as the public encryption key and secret encryption key unique to the class of the memory card. In memory card 110, it is assumed that the natural number  $m$  is represented as  $m = 1$ .

[0074] Memory card 110 includes an authentication data hold unit 1400 to store authentication data  $\{KPmc(1)\}KPma$ , a  $Kmc$  hold unit 1402 storing a unique decryption key  $Kmc(1)$  set for each memory card class, a  $KPm(1)$  hold unit 1416 to store a unique public encryption key  $KPm(1)$  set for each memory card, and a  $Km(1)$  hold unit 1421 storing an asymmetric private decryption key  $Km(1)$  that can be decrypted using public encryption key  $KPm(1)$ . Here, authentication data hold unit 1400 encrypts and stores public encryption key  $KPmc(1)$  set for each memory card class using authentication key  $KPma$  in an authenticatable state. Authentication data hold unit 1400 encrypts and stores public encryption key  $KPmc(1)$  set for each memory card class in a state that can have the authenticity verified by decryption using authentication key  $KPma$ .

[0075] Memory card 110 further includes a data bus BS3 to transfer a signal with memory interface 1200 via a terminal 1202, a decryption processing unit 1404 receiving the data applied from memory interface 1200 onto data bus BS3, and receiving a private decryption key  $Kmc(1)$  unique to each memory card class from  $Kmc(1)$  hold unit 1402, and providing session key  $Ks3$  generated by the distribution server in a distribution session to contact  $Pa$ , a decryption processing unit 1408 receiving authentication key  $KPma$  from  $KPma$  hold unit 1443 to execute a decryption process using authentication key  $KPma$  from the data applied on data bus BS3 and providing the decrypted result to encryption processing unit 1410, and an encryption processing unit 1406 encrypting data selectively applied from switch 1444 using a key selectively applied by switch 1442, and providing the encrypted data onto data bus BS3.

[0076] Memory card 110 further includes a session key generation unit 1418 generating a session key at each distribution, reproduction and transfer session, an encryption processing unit 1410 encrypting the session key output from session key generation unit 1418 using public encryption key  $KPp(n)$  obtained by encryption processing unit 1408 to output the encrypted key onto data bus BS3, and a decryption processing unit 1412 receiving encrypted data on data bus BS3 to apply a decryption process using session key  $Ks3$  obtained by session key generation unit 1418, and providing the decrypted result to data bus BS4.

[0077] Memory card 110 further includes an encryption processing unit 1424 encrypting the data on data bus BS4 using a public encryption key  $KPm(i)$  ( $i$  is 1 or number  $j$  of another memory card) unique to the memory card, a decryption processing unit 1422 to decrypt the

data on data bus BS4 using a secret encryption key Km (1) unique to memory card 110 that is the companion to public encryption key KPm(1), and a memory 1415 receiving and storing from data bus BS4 a portion of the reproduction information encrypted with public encryption key KPm(1) (content decryption key Kc, content ID, license ID access control information AC1, reproduction circuit control information AC2), as well as receiving and storing encrypted content data {Data}Kc.

**[0078]** Memory card 110 further includes a license information hold unit 1440 storing license information obtained by decryption processing unit 1422 (transaction ID, content ID and access restriction information AC1), a log memory 1460 to store the log of the transmission/reception of the reproduction information in the memory card, and a controller 1420 transferring data with an external source via data bus BS3 to receive reproduction information and the like with data bus BS4 to control the operation of memory card 110.

**[0079]** "Reception log" indicating the reception status of the reproduction information stored in log memory 1460 includes the transaction ID, session key Ks2, and the like. In the first embodiment, the reception log information corresponds to data generated in the event of license reception, and is erased when reception and storage of the reproduction information to memory card 110 are completed.

**[0080]** It is assumed that the region TRM enclosed by the solid line in Fig. 5 is incorporated in a module TRM to disable readout of data and the like in the circuit located in that region by a third party by erasing the internal data or destroying the internal circuitry when an improper open process is conducted from an external source. Such a module is generally a tamper resistant module.

**[0081]** A structure may be implemented in which memory 1415 is also incorporated in module TRM. However, since the data stored in memory 1415 is completely encrypted according to the structure shown in Fig. 6, a third party will not be able to reproduce the music with just the data in memory 1415. Furthermore, it is not necessary to provide memory 1415 in the expensive tamper resistance module. Thus, there is the advantage that the fabrication cost is reduced.

#### [Distribution Operation]

**[0082]** The operation in each session of the data distribution system according to an embodiment of the present invention will be described in detail hereinafter with reference to the flow charts.

**[0083]** Figs. 6, 7 and 8 are the first, second and third flow charts, respectively, to describe a distribution operation in the event of purchasing content according to the data distribution system of the first embodiment (also called "distribution session" hereinafter).

**[0084]** Figs. 6-8 correspond to the operation of user 1 receiving content data distribution from distribution

server 30 via cellular phone 100 using memory card 110.

**[0085]** First, a distribution request is issued from cellular phone 100 of user 1 through the operation of the key buttons on touch key unit 1108 by user 1 (step S100).

**[0086]** At memory card 110, authentication data {KPmc(1)}KPma is output from authentication data hold unit 1400 in response to the distribution request (step S102).

**[0087]** Cellular phone 100 transmits to distribution server 30 authentication data {KPP(1)}KPma for authentication of cellular phone 100 per se, the content ID and license purchase condition AC in addition to authentication data {KPmc(1)}KPma accepted from memory card 110 for authentication (step S104).

**[0088]** Distribution server 30 receives the content ID, authentication data {KPmc(1)}KPma, {KPP(1)}KPma, license purchase condition data AC from cellular phone 100 (step S106). Decryption processing unit 312 executes a decryption process using authentication key KPma. Accordingly, distribution server 30 accepts public encryption key KPmc(1) of memory card 110 and KPP(1) which is the public encryption key of cellular phone 100 (step S108).

**[0089]** Distribution control unit 315 conducts authentication by authentication server 12 based on the accepted secret encryption keys KPmc(1) and KPP(1) (step S110). When these public encryption keys are valid, control proceeds to the next process (step S112). When these public secret keys are invalid, the process ends (step S170).

**[0090]** In verifying the authenticity of public encryption key KPP(1) or KPmc(1) in the decryption process by authentication key KPma, authentication server 12 performs the authentication. Since public encryption key KPP(1) or KPmc(1) is encrypted so that its authenticity can be determined by decrypting using authentication key KPma, a structure may be implemented in which distribution control unit 315 of license server 10 performs authentication from the decryption result using authentication key KPma.

**[0091]** When verification is made that the distribution is towards a proper memory card as a result of authentication, distribution control unit 315 generates a transaction ID to identify the distribution session (step S112).

**[0092]** When verification is made that the distribution is towards a proper memory card as a result of authentication, distribution control unit 315 also records the transaction ID, content ID, public encryption keys KPmc(1) and KPP(1) in administration database 306 together with the information indicating unsettled accounting (accounting status flag) as the license distribution log (step S113).

**[0093]** At distribution server 30, session key generation unit 316 generates a session key Ks1 for distribution. Session key Ks1 is encrypted by encryption processing unit 318 using a public encryption key KPmc(1) corresponding to memory card 110 obtained from decryption processing unit 312.

[0094] The transaction ID and encrypted session key  $\{Ks1\}Kmc(1)$  are output via data bus BS1 and communication device 350 (step S116).

[0095] Upon reception of the transaction ID and encrypted session key  $\{Ks1\}Kmc(1)$  at cellular phone 100, (step S118), the received data is applied onto data bus BS3 via memory interface 1200 in memory card 110. Decryption processing unit 1404 decrypts  $\{Ks1\}Kmc(1)$  using a private decryption key  $Kmc(1)$  unique to memory card 110 stored in hold unit 1402, whereby session key  $Ks1$  is decrypted and extracted. As a result, the transaction ID and session key  $Ks1$  are accepted (step S120).

[0096] The procedure up to step S120 is referred to as the "transaction ID obtain step".

[0097] Referring to Fig. 7, upon confirmation of the acceptance of session key  $Ks1$  generated at distribution server 30, controller 1420 designates session key generation unit 1418 to generate a session key  $Ks2$  generated in the distribution operation of the memory card. Controller 1420 also records in log memory 1460 session key  $Ks2$  together with the received transaction ID (step S121).

[0098] Encryption processing unit 1406 encrypts session key  $Ks2$  applied by sequential switching of the contact of switches 1444 and 1446 as well as public encryption key  $KPmc(1)$  using session key  $Ks1$  applied from decryption processing unit 1406 via contact Pa of switch 1442, whereby  $\{Ks2//KPmc(1)\}Ks1$  is output onto data bus BS3 (step S122).

[0099] Encrypted data  $\{Ks2//KPmc(1)\}Ks1$  output onto data bus BS3 is transmitted from data bus BS3 to cellular phone 100 via terminal 1202 and memory interface 1200, and then transmitted from cellular phone 100 to distribution server 30 (step S124).

[0100] Distribution server 30 receives encrypted data  $\{Ks2//KPmc(1)\}Ks1$  to execute a decryption process using session key  $Ks1$  by decryption processing unit 320. Session key  $Ks2$  generated at the memory card and public encryption key  $KPmc(1)$  unique to memory card 110 are accepted (step S126).

[0101] Distribution control unit 315 generates access restriction information AC1 and reproduction circuit restriction information AC2 according to the content ID and license purchase condition data AC obtained at step S106 (step S130). Also, license key Kc to decrypt the encrypted content data is obtained from information database 304 (step S132).

[0102] Distribution control unit 315 applies the obtained license key Kc and reproduction circuit restriction information AC2 to encryption processing unit 324. Encryption processing unit 324 encrypts license key Kc and reproduction circuit restriction information AC2 using secret common key Kcom obtained from Kcom hold unit 322 (step S134).

[0103] Encrypted data  $\{Kc//AC2\}Kcom$  output from encryption processing unit 324, and the transaction ID, content ID and access restriction information AC1 out-

put from distribution control unit 315 are encrypted by encryption processing unit 326 using a public encryption key  $KPm(1)$  unique to memory card 110 obtained by decryption processing unit 320 (step S136).

[0104] Encryption processing unit 328 receives the output of encryption processing unit 326 and applies encryption using session key  $Ks2$  generated by memory card 110 (step S137).

[0105] Distribution control unit 315 records access restriction information AC1, reproduction circuit restriction information AC2, public encryption key  $KPm(1)$ , session key  $Ks2$  in log data administration database 306 together with the information of settled accounting (accounting status flag) (step S138).

[0106] Encrypted data  $\{\{Kc//AC2\}Kcom//transaction\ ID//content\ ID//AC1\}Km(1)\}Ks2$  output from encryption processing unit 328 is transmitted to cellular phone 100 via data bus BS1 and communication device 350 (step S139).

[0107] By transferring respective session keys generated at the transmission server and memory card to each other to execute encryption using respective received encryption keys and transmitting the encrypted data to the other party, authentication of each other can be virtually conducted in the transmission/reception of respective encrypted data. Thus, security of the data distribution system can be improved. Furthermore, distribution server 30 will record and store the accounting status and information associated with the distribution history.

[0108] Cellular phone 100 receives the transmitted encrypted data  $\{\{Kc//AC2\}Kcom//transaction\ ID//content\ ID//AC1\}Km(1)\}Ks2$  (step S140). At memory card 110, the received data applied onto data bus BS3 via memory interface 1200 is decrypted by decryption processing unit 1412. Specifically, decryption processing unit 1412 decrypts the reception data on data bus BS3 using session key  $Ks2$  applied from session key generation unit 1418 and provides the decrypted data onto data bus BS4 (step S144).

[0109] Referring to Fig. 8, data  $\{Kc//AC2\}Kcom//license\ ID//content\ ID//AC1\}Km(1)$  decryptable with private decryption key  $Km(1)$  stored in  $Km(1)$  store unit 1421 is output onto data bus BS4 at the stage of step S144. This data  $\{Kc//AC2\}Kcom//transaction\ ID//content\ ID//AC1\}Km(1)$  is first decrypted by a private decryption key  $Km(1)$ , whereby data  $\{Kc//AC2\}Kcom$ , the transaction ID, content ID, and access control information AC1 which are the reproduction information are accepted (step S146).

[0110] The transaction ID, content ID and access restriction information AC1 are recorded in license information hold unit 1440. Data  $\{Kc//AC2\}Kcom$  is encrypted again with public encryption key  $KPm(1)$  and stored in memory 1415 as data  $\{\{Kc//AC2\}Kcom\}Km(1)$  (step S148).

[0111] The reception log in log memory 1460 is erased (step S150).

[0112] The process from step S121 to step S150 is referred to as the "reproduction information obtain step". In this "reproduction information obtain step", the accounting subject process is carried out.

[0113] At the stage of proper completion of the process up to step S150, a content data distribution request is issued from cellular phone 100 to distribution server 30 (step S152).

[0114] In response to reception of a content data distribution request, distribution server 30 obtains encrypted content data {Data}Kc and additional information Data-inf from information database 304 and outputs the same via data bus BS1 and communication device 350 (step S154).

[0115] Cellular phone 100 receives {Data}Kc/Data-inf, and accepts encrypted content data {Data}Kc and additional information Data-inf (step S156). Encrypted content data {Data}Kc and additional information Data-inf are transmitted onto data bus BS3 of memory card 110 via memory interface 1200 and terminal 1202. At memory card 110, the received encrypted content data {Data}Kc and additional information Data-inf are directly stored in memory 1415 (step S158).

[0116] The process from step S152 to step S158 is referred to as the "content data obtain step". In this "content data obtain step", a process not subject to accounting is carried out.

[0117] A distribution acceptance notification is transmitted from memory card 110 to distribution server 30 (step S160). Upon reception of the distribution acceptance at distribution server 30 (step S162), the distribution end process is executed accompanying storage of the accounting data into account database 302 (step S164). Thus, the distribution server process ends (step S170).

#### [Reconnection Operation]

[0118] The process when reconnection is to be established to receive distribution again when the communication line is disrupted during the stage of the above-described process of the distribution operation will be described hereinafter. Fig. 9 is a flow chart to describe a reconnection process.

[0119] User 1, for example, requests reconnection through the key button or the like on keyboard 1108 of cellular phone 100, whereby the reconnection process is initiated (step S200).

[0120] Controller 1106 of cellular phone 100 determines the processing step where communication was disrupted (step S202). If disruption has occurred in the transaction ID obtain step, the basic distribution process of Figs. 6-8 (first reconnection process) is effected since it is not relevant to accounting (step S204). Then, the reconnection process ends (step S206).

[0121] When determination is made that the step where communication has been disrupted is the license obtain step (step S202), controller 1106 carries out a

second reconnection process based on a reception log that will be described afterwards (step S206). When communication has been disrupted in the content data obtain step (step S202), a third reconnection process to continue communication corresponding to communication disruption that will be described afterwards is effected (step S206). Then, the reconnection process ends (step S210).

#### [Second Reconnection Process]

[0122] Figs. 10, 11 and 12 are the first, second and third flow charts, respectively, to describe a second reconnection process in the data distribution system of the first embodiment. By comparing the license distribution log of license server 10 and the reception log of memory card 110, the reproduction information distribution status when communication has been disrupted is confirmed to realize reliability for the user while protecting the rights of copyright owners.

[0123] Referring to Fig. 10, user 1 operates the key button of keyboard 1108 of cellular phone 100 to issue a reconnection request. In response, the second reconnection process is initiated (step S300).

[0124] In response to this reconnection request, the transaction ID stored in log memory 1460 is output at memory card 110 (step S302).

[0125] Cellular phone 100 transmits the transaction ID accepted from memory card 110 towards distribution server 30 (step S304).

[0126] At distribution server 30, the transaction ID is received (step S306). Distribution control unit 315 retrieves the license distribution log from log administration database 306 (step S308).

[0127] When an accounting process has been already performed for the terminal that has requested reconnection (cellular phone 100 and memory card 110) from the received transaction ID (step S308), distribution control unit 315 obtains public encryption key KPmc(1) from the license distribution log (step S310).

[0128] Session key generation unit 316 generates a session key Ks1 for distribution. Session key Ks1 is encrypted by encryption processing unit 318 using public encryption key KPmc(1) (step S312).

[0129] The transaction ID and encrypted session key {Ks1}Kmc(1) are output via data bus BS1 and communication device 350 (step S314).

[0130] In response to reception of the transaction ID and encrypted session key {Ks1}Kmc(1) at cellular phone 100 (step S316), decryption processing unit 1404 of memory card 110 decrypts the received data applied onto data bus BS3 via memory interface 1200 using a private decryption key Kmc(1) unique to memory card 110 stored in hold unit 1402, whereby session key Ks1 is decrypted and extracted (step S318).

[0131] The subsequent steps are similar to the process after step S121 of Fig. 7, i.e., the process following the license obtain step.

[0132] When determination is made that the accounting process has not been completed as a result of looking in the license distribution log from log administration database 306 by distribution control unit 315 at step S308, public encryption key KPmc(1) is obtained from the license distribution log (step S330).

[0133] Then, session key generation unit 316 at distribution server 30 generates a session key Ks1 for distribution. Session key Ks1 is encrypted by encryption processing unit 318 using public encryption key KPmc(1) (step S332).

[0134] The transaction ID and encrypted session key {Ks1}Kmc(1) are output via data bus BS1 and communication device 350 (step S334).

[0135] In response to reception of the transaction ID and encrypted session key {Ks1}Kmc(1) at cellular phone 100 (step S336), decryption processing unit 1404 decrypts the reception data applied onto data bus BS3 via memory interface 1200 using private decryption key Kmc(1) unique to memory card 110 stored in hold unit 1402, whereby session key Ks1 is decrypted and extracted (step S338).

[0136] Encryption processing unit 1406 encrypts the received log with session key Ks1 to generate {reception log}Ks1 (step S340).

[0137] Referring to Fig. 11, controller 1420 designates session key generation unit 1418 to generate a session key Ks2' generated in the distribution operation of the memory card (step S342).

[0138] Encryption processing unit 1406 encrypts session key Ks2' applied via the contacts of switches 1444 and 1446 using session key Ks1 applied from decryption processing unit 1404 via contact Pa of switch 1442 to generate {Ks2'}Ks1. The generated data {reception log}Ks1 and {Ks2'}Ks1 are output from memory card 110 (step S344).

[0139] Encrypted data {reception log}Ks1 and {Ks2'}Ks1 output onto data bus BS3 are transmitted from data bus BS3 to cellular phone 100 via terminal 1202 and memory interface 1200, and transmitted from cellular phone 100 to distribution server 30 (step S346).

[0140] Distribution server 30 receives encrypted data {reception log}Ks1 and {Ks2'}Ks1. Decryption processing unit 320 executes a decryption process using session key Ks1, whereby session key Ks2' generated by the reception log and memory card is accepted (step S348).

[0141] Then, distribution control unit 316 verifies the authenticity of the received reception log (step S350).

[0142] When authenticity of the reception log is not verified, the second reconnection process ends (step S390).

[0143] In contrast, when the authenticity of the reception log is verified, distribution control unit 315 obtains the content ID, access restriction information AC1, reproduction circuit restriction information AC2 and public encryption key KPm(1) from the license distribution log (step S352). Then, license key Kc to decrypt the en-

rypted content data is obtained from information database 304 (step S354).

[0144] Distribution control unit 315 applies the obtained license key Kc and reproduction circuit restriction information AC2 to encryption processing unit 324. Encryption processing unit 324 encrypts license key Kc and reproduction circuit restriction information AC2 using secret common key Kcom obtained from Kcom hold unit 322 (step S356).

[0145] Encrypted data {Kc//AC2}Kcom output from encryption processing unit 324 and the transaction ID, content ID and access restriction information AC1 output from distribution control unit 315 are encrypted by encryption processing unit 326 using public encryption key KPm(1) unique to memory card 110 obtained at step S352 (step S358).

[0146] Encryption processing unit 328 receives the output of encryption processing unit 326 to encrypt the output using session key Ks2' generated at memory card 110 (step S360).

[0147] Encrypted data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' output from encryption processing unit 328 is transmitted to cellular phone 100 via data bus BS1 and communication device 350 (step S362).

[0148] Cellular phone 100 receives the transmitted encryption data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' (step S364).

[0149] Referring to Fig. 12, memory card 110 has the reception data applied onto data bus BS3 via memory interface 1200 decrypted by decryption processing unit 1412. Specifically, decryption processing unit 1412 uses session key Ks2' applied from session key generation unit 1418 to decrypt the reception data on data bus BS3, and provides the decrypted data onto data bus BS4 (step S366).

[0150] At this stage, data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' is decryptable with private decryption key Km(1) stored in Km(1) hold unit 1421 is output onto data bus BS4. This data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' is decrypted with private decryption key Km(1), whereby data {Kc//AC2}Kcom, the transaction ID, the content ID, and access restriction information AC1 corresponding to the reproduction information are accepted (step S368).

[0151] The transaction ID, content ID, access restriction information AC1 are stored in license information hold unit 1440. Data {Kc//AC2}Kcom is encrypted again using a private decryption key KPm(1) and stored in memory 1415 as data {{{Kc//AC2}Kcom}Km(1)} (step S370).

[0152] Also, the reception log is erased from log memory 1460 (step S372).

[0153] At the stage of proper completion of the process up to step S372, a content data distribution request is issued from cellular phone 100 to distribution server 30 (step S374).

[0154] In response to this content data distribution re-

quest, distribution server 30 obtains encrypted content data {Data}Kc and additional information Data-inf from information database 304. These data are output via data bus BS1 and communication device 350 (step S376).

**[0155]** Cellular phone 100 receives {Data}Kc/Data-inf, and accepts encrypted content data {Data}Kc and additional information Data-inf (step S378). Encrypted content data {Data}Kc and additional information Data-inf are transmitted onto data bus BS3 of memory card 110 via memory interface 1200 and terminal 1202. At memory card 110, the received encrypted content data {Data}Kc and additional information Data-inf are directly stored in memory 1415 (step S380).

**[0156]** A distribution reception notification is transmitted from memory card 110 to distribution server 30 (step S382). When the distribution acceptance is received at distribution server 30 (step S384), the distribution end process is executed (step S386). The process of the distribution server ends (step S390).

[Third Reconnection Process]

**[0157]** Fig. 13 is a flow chart to describe a third reconnection operation in the data distribution system of the first embodiment.

**[0158]** Referring to Fig. 13, user 1 sends a reconnection request through the key button on keyboard 1108 of cellular phone 100. In response, a third reconnection process is initiated (step S400).

**[0159]** In response to this reconnection request, cellular phone 100 sends a content data distribution request to distribution server 30 (step S402).

**[0160]** In response to this content data distribution request, distribution server 30 obtains encrypted content data {Data}Kc and additional information Data-inf from information database 304. These data are output via data bus BS1 and communication device 350 (step S404).

**[0161]** Cellular phone 100 receives {Data}Kc/Data-inf, and accepts encrypted content data {Data}Kc and additional information Data-inf (step S406). Encrypted content data {Data}Kc and additional information Data-inf are transmitted onto data bus BS3 of memory card 110 via memory interface 1200 and terminal 1202. At memory card 110, the received encrypted content data {Data}Kc and additional information Data-inf are directly stored in memory 1415 (step S408).

**[0162]** Then, a distribution acceptance notification is transmitted from memory card 110 to distribution server 30 (step S410). When distribution server 30 receives this distribution acceptance (step S412), a distribution end process is executed (step S414). The process of the distribution server ends (step S416).

[Reconnection Operation When Line Is Cut During Reconnection Operation]

**[0163]** The process of establishing reconnection to receive distribution again in the case where the commu-

nication line is cut off in the stage of the processing step of the above-described reconnection operation will be described here. Fig. 14 is a flow chart to describe such a reconnection process.

**[0164]** User 1, for example, operates the key button on keyboard 1108 of cellular phone 100 to send a reconnection request. The reconnection process is initiated (step S500).

**[0165]** Based on the license reception standby log stored in memory card 110, controller 1106 determines the step where communication has been disrupted (step S502). When communication has been disrupted at the license obtain step or license reobtain step, the second reconnection process is performed again (step S504). Then, the reconnection process ends (step S508).

**[0166]** When determination is made that the step where communication has been disrupted is the content data obtain step by controller 1106 (step S502), a third reconnection process that will be described afterwards is carried out (step S506). Then, the reconnection process ends (step S508).

**[0167]** By virtue of such a structure, reconnection can be established even in the case where the communication line has been disrupted in the processing step. Thus, the reliability of the system is further improved.

[Second Embodiment]

**[0168]** The data distribution system of the second embodiment differs in the data distribution system of the first embodiment in that the license reception standby log stored in log memory 1460 in memory card 110 is not erased, as will be described hereinafter. Corresponding to this modification, the reception log includes, in addition to the structure of the first embodiment, a reception status flag.

**[0169]** The data distribution system of the second embodiment differs from the first embodiment in the operation of controller 1420 in memory card 110 and the data stored in log memory 1460.

**[0170]** Figs. 15, 16 and 17 are the first, second and third flow charts, respectively, to describe a distribution operation in the event of purchasing content in the data distribution system of the second embodiment, and is comparable to Figs. 6-8 of the first embodiment.

**[0171]** Figs. 15-17 correspond to the operation of user 1 receiving music data distribution from distribution server 30 via cellular phone 100 by using memory card 110.

**[0172]** The difference from the flow of the first embodiment is that, at step S121' of Fig. 16 following the transaction ID obtain step, controller 1420 designates session key generation unit 1418 to generate a session key Ks2 generated during the distribution operation of the memory card upon confirming acceptance of session key Ks1 generated at distribution server 30. Furthermore, controller 1420 records a reception status flag attaining an ON status indicating a reception wait state as



the reception log in log memory 1460 together with session key Ks2 and the received transaction ID (step S121').

**[0173]** Referring to Fig. 17, at step S148, the transaction ID, content ID and access restriction information AC1 are recorded in license information hold unit 1440. Data {Kc//AC2}Kcom is encrypted by public encryption key KPm(1), and stored in memory 1415 as data {{Kc//AC2}Kcom}Km(1). Then, the reception status flag in the reception log in log memory 1460 attains an OFF status indicating that reception has ended (step S150').

**[0174]** The remaining process is similar to that of the first embodiment. The same steps have the same reference characters allotted, and description thereof will not be repeated.

#### [Reconnection Operation]

**[0175]** Similar to Fig. 9 of the first embodiment, the second embodiment carries out a reconnection process to receive distribution again when the communication line has been disrupted at the stage of the processing step of the distribution operation.

**[0176]** It is to be noted that the second reconnection process is partially modified from that of the first embodiment.

#### [Second Reconnection Process]

**[0177]** Figs. 18, 19 and 20 are the first, second and third flow charts, respectively, to describe a second reconnection operation in the data distribution system of the second embodiment, and are comparable to Figs. 10-12 of the first embodiment.

**[0178]** Difference from the process of the first embodiment is that control proceeds to step S121' of Fig. 16 after accepting session key Ks1 at step S318, and the transaction ID, content ID and access restriction information AC1 are recorded in license information hold unit 1440 at step S370 shown in Fig. 20. Data {Kc//AC2}Kcom is encrypted using public encryption key KPm(1), and stored in memory 1415 as data {{Kc//AC2}Kcom}Km(1). Then at step S372', a process of rendering the reception status flag of the reception log OFF indicating that reception has ended is carried out.

**[0179]** The remain process is similar to that of first embodiment. Corresponding steps have the same reference characters allotted, and description thereof will not be repeated.

**[0180]** The third reconnection process as well as the reconnection operation when the line is cut off during a reconnection operation are similar to the process of Fig. 1.

**[0181]** By such a structure, reconnection can be established even in the case where the communication line is disrupted in the processing step. Thus, the reliability of the system is further improved.

[Third Embodiment]

**[0182]** The distribution system of the third embodiment differs from the data distribution system of the second embodiment in that status information with a status flag is transmitted to the server in the reception log stored in log memory 1460 in memory card 110.

**[0183]** The status information includes the transaction ID, session key Ks2, reception status flag and status flag corresponding to the reception log.

**[0184]** Here, the license status flag is a flag variable of 3 states. The license status flag takes the value of "01h" when the transaction ID recorded in the reception log is present in license information hold unit 1440 of memory card 110, corresponding reproduction information is present, and reproduction is not inhibited by the access restriction information stored in license information hold unit 1440, i.e. when in a reproducible state; takes the value of "00h" when there is the transaction ID in the license information hold unit, and there is no corresponding reproduction information or when reproduction is inhibited by the access restriction information stored in license information hold unit 1440 so that reproduction cannot be performed; and takes the value of "FFh" when there is no transaction ID.

**[0185]** The structure of the data distribution system of the third embodiment differs in the operation of controller 1420 of memory card 110 and the data stored in log memory 1460 as will be described hereinafter.

**[0186]** The distribution operation and reconnection operation of the third embodiment are similar to those of the second embodiment except for the second reconnection process set forth below.

#### [Second Reconnection Process]

**[0187]** Figs. 21, 22, 23 and 24 are the first, second, third and fourth flow charts, respectively, to describe the second reconnection operation of the data distribution system of the third embodiment.

**[0188]** Referring to Fig. 21, the process from step S300 to step S338 is similar to the second reconnection operation of the second embodiment.

**[0189]** At step S338, the reception data applied onto data bus BS3 via memory interface 1200 in memory card 110 is decrypted by decryption processing unit 1404 using private decryption key Kmc(1) unique to memory card 110 stored in hold unit 1402, whereby session key Ks1 is decrypted and extracted. Then, controller 1420 in memory card 110 retrieves data stored in license information hold unit 1440 according to the transaction ID in the reception log stored in log memory 1460 (step S640).

**[0190]** Controller 1420 checks whether there is a transaction ID in license information hold unit 1440 (step S642).

**[0191]** When there is no transaction ID, the license status flag is set to "FFh" (step S644), and control pro-

ceeds to step S652.

**[0192]** When there is the transaction ID at step S642, controller 1420 confirms the status of access restriction information AC1 stored in license information hold unit 1440 and whether a corresponding license key Kc is recorded in memory 1415 (step S646). When reproduction is allowed, the license status flag is set to "01h" (step S648). When reproduction is not allowed, the license status flag is set to "00h" (step S650). Then, control proceeds to step S652).

**[0193]** The status information with the status flag added to the reception log stored in log memory 1460 is generated (step S652).

**[0194]** Controller 1462 designates session key generation unit 1418 to generate a session key Ks2' generated in the distribution operation of the memory card (step S654).

**[0195]** Decryption processing circuit 1406 decrypts the status information and session key Ks2' using session key Ks1 (step S656).

**[0196]** Controller 1420 obtains the hash value according to the hash function corresponding to encrypted data {status information//Ks2'}Ks1 to generate signature data "hash" for encrypted data {status information//Ks2'}Ks1 (step S658).

**[0197]** Encryption processing unit 1406 encrypts the signature data hash applied under control of controller 1420 using session key Ks1 applied from decryption processing unit 1402 via contact Pa of switch 1442 to generate encrypted signature data {hash}Ks1 (step S660).

**[0198]** The generated data {status information//Ks2'}Ks1 and encrypted signature data {hash}Ks1 are output from memory card 110 (step S662).

**[0199]** Encrypted data {status information//Ks2'}Ks1 and encrypted signature data {hash}Ks1 output onto data bus BS3 are transmitted from data bus BS3 to cellular phone 100 via terminal 1202 and memory interface 1200, and transmitted from cellular phone 100 to distribution server 30 (step S644).

**[0200]** Distribution server 30 receives encrypted data {status information//Ks2'}Ks1 and encrypted signature data {hash}Ks1 (step S666).

**[0201]** Referring to Fig. 23, decryption processing unit 320 of distribution server 30 executes a decryption process on encrypted signature data {hash}Ks1 using session key Ks1 to obtain signature data hash corresponding to encrypted data {status information//Ks2'}Ks1. Then, the authenticity of the status information is checked based on encrypted data {status information//Ks2'}Ks1 and the signature data (step S668).

**[0202]** The process ends if the status information is not proper (step S712). When the authenticity of the status information is verified, a decryption process is executed using session key Ks1. The status information and session key Ks2' generated by the memory card are accepted (step S670).

**[0203]** Distribution control unit 315 verifies the au-

thenticity of the reproduction information retransmission request based on the received status information and license distribution log (step S672).

**[0204]** When the authenticity of the reproduction information retransmission request is not verified, the second reconnection process ends (step S712).

**[0205]** In contrast, if the authenticity of the reproduction information transmission request is verified, distribution control unit 315 obtains the content ID, access restriction information AC1, reproduction circuit restriction information AC2 and public encryption key Kpm(1) from the license distribution log (step S674). Then, license key Kc to decrypt the encrypted content data is obtained from information database 304 (step S676).

**[0206]** Distribution control unit 315 applies the obtained license key Kc and reproduction circuit restriction information AC2 to encryption processing unit 324. Encryption processing unit 324 encrypts license key Kc and reproduction circuit restriction information AC2 using secret common key Kcom obtained from Kcom hold unit 322 (step S678).

**[0207]** Encrypted data {Kc//AC2}Kcom output from encryption processing unit 324, and the transaction ID, content ID and access restriction information AC1 output from distribution control unit 315 are encrypted by encryption processing unit 326 using public encryption key Kpm(1) unique to memory card 110 obtained at step S674 (step S680).

**[0208]** Encryption processing unit 328 receives the output of encryption processing unit 326 to encrypt the same using session key Ks2' generated at memory card 110 (step S682).

**[0209]** The encrypted data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' output from encryption processing unit 328 is transmitted to cellular phone 100 via data bus BS1 and communication device 350 (step S684).

**[0210]** Cellular phone 100 receives the transmitted encrypted data {{{Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1)}Ks2' (step S686).

**[0211]** Referring to Fig. 24, memory card 110 has the reception data applied onto data bus BS3 via memory interface 1200 decrypted by decryption processing unit 1412. Decryption processing unit 1412 uses session key Ks2' applied from session key generation unit 1418 to decrypt the reception data on data bus BS3. The decrypted data is output onto data bus BS4 (step S690).

**[0212]** At this stage, data {Kc//AC2}Kcom//license ID//content ID//AC1}Km(1) that can be decrypted with private decryption key Km(1) stored in Km(1) hold unit 1421 is output. This data {Kc//AC2}Kcom//transaction ID//content ID//AC1}Km(1) is decrypted by public encryption key Km(1), whereby data {Kc//AC2}Kcom, the transaction ID, content ID and access restriction information AC1 are accepted (step S692).

**[0213]** The transaction ID, content ID, access restriction information AC1 are recorded in license information hold unit 1440. Data {Kc//AC2}Kcom is encrypted with

public encryption key  $K_{Pm}(1)$ , and stored in memory 1415 as data  $\{Kc//AC2\}Kcom\}Km(1)$  (step S694).

[0214] Then, the reception status flag in the reception log in log memory 1460 is altered to the off state indicating that reception has ended (step S696).

[0215] At the stage of proper completion of the process up to step S372, a content data distribution request is issued from cellular phone 100 to distribution server 30 (step S698).

[0216] In response to this content data distribution request, distribution server 30 obtains encrypted content data  $\{Data\}Kc$  and additional information Data-inf from information database 304. These data are output via data bus BS1 and communication device 350 (step S700).

[0217] Cellular phone 100 receives  $\{Data\}Kc//Data-inf$ , and accepts encrypted content data  $\{Data\}Kc$  and additional information Data-inf (step S702). Encrypted content data  $\{Data\}Kc$  and additional information Data-inf are transmitted onto data bus BS3 of memory card 110 via memory interface 1200 and terminal 1202. At memory card 110, the received encrypted content data  $\{Data\}Kc$  and additional information Data-inf are directly stored in memory 1415 (step S704).

[0218] A distribution acceptance notification is transmitted from memory card 110 to distribution server 30 (step S706). When the distribution acceptance is received at distribution server 30 (step S708), the distribution end process is executed (step S710). The process of the distribution server ends (step S712).

[0219] The above description is based on a structure in which all the status information is encrypted using session key  $Ks1$  at step S654, and encrypted data  $\{status\ information//Ks2\}Ks1$  is transmitted to distribution server 30 at steps S622 and S624.

[0220] The transaction ID in the status information is required only to identify its source so that its security is not so important. Since the source becomes apparent by encrypted signature data  $\{hash\}Ks1$ , the transaction ID does not have to be encrypted and can be transmitted to distribution server 30 in plaintext. In this case, the status information will be transmitted as transaction ID// $\{status\ information\ excluding\ transaction\ ID//Ks2\}Ks1$ , and signature data hash will be generated correspondingly.

[0221] By such a structure, reconnection can be established even when the communication line has been cut off in the processing step. Thus, the reliability of the system is further improved.

[0222] The data distribution system of the first to third embodiments was described in which encryption and decryption are carried out using secret common key  $Kcom$  at distribution server 30 and cellular phone 100. A structure implementing encryption and decryption without this secret common key  $Kcom$  is allowed.

[0223] In other words, a structure can be implemented in which distribution server 30 corresponding to the data distribution system of the first embodiment described with reference to Fig. 3 is absent of  $Kcom$  hold unit 322

and encryption processing unit 324. More specifically, license key  $Kc$  and reproduction circuit restriction information AC2 output from distribution control unit 315 can be directly transmitted to encryption processing unit 326 in distribution server 30.

[0224] Furthermore in comparison to the structure of cellular phone 100 described with reference to Fig. 4 in the first embodiment, a structure can be implemented absent of  $Kcom$  hold unit 1512 storing a secret common key  $Kcom$  and a decryption processing unit 1514 using secret common key  $Kcom$ .

[0225] In cellular phone 101 of such a structure, license key  $Kc$  is directly obtained by decryption processing unit 1510 that executes a decryption process using session key  $Ks4$  in view that an encryption process is not performed with a secret symmetric key as a symmetric encryption key in distribution server 30. Therefore, license key  $Kc$  is directly applied to decryption processing unit 1510.

[0226] In a structure where encryption and decryption is not effected using secret common key  $Kcom$ , memory card 110 can be used intact.

[0227] In a distribution process of such a case, content key  $Kc$  and reproduction circuit restriction information AC2 are transmitted and stored without being encrypted with secret common key  $Kcom$ . The encryption process and corresponding decryption process by secret common key  $Kcom$  are no longer required. The remaining elements are similar to those of the operation of the first to third embodiments.

[0228] By such a structure, a data distribution system that enjoys advantages similar to those of the data distribution system of the first to third embodiments can be developed with a structure that does not effect an encryption process associated with secret common key  $Kcom$ .

[0229] The above-described first to third embodiments may be subject to modifications set forth below.

[0230] The first to third embodiments had data  $\{Kc//AC2\}Kcom$  (or data  $Kc//AC2$  in the structure without key  $Kcom$  as mentioned above) encrypted by public encryption key  $K_{Pm}(1)$ , and recorded in license information storage unit 1440. However, the second encryption using public encryption key  $K_{Pm}(1)$  is not necessary if stored in license information hold unit 1440 provided in the TRM. Advantages similar to those of the first to third embodiments can be provided even if the entire reproduction information is stored in license information hold unit 1440. In this case, step S148 of Fig. 8 and step S370 of Fig. 12 in the first embodiment are to be modified to "record the transaction ID, content ID, AC1,  $\{Kc//AC2\}Kcom$  in the license information hold unit". Also, step S148 of Fig. 17 and step S370 of Fig. 20 in the second embodiment and step S694 of Fig. 24 in the third embodiment are to be modified similarly to "record the transaction ID, content ID, AC1,  $\{Kc//AC2\}Kcom$  in the license information hold unit." If a structure without key  $Kcom$  is to be implemented corresponding to modifica-

tions of first to third embodiments, the process is to be modified to "record the transaction ID, content ID, AC1, Kc//AC2 in the license information hold unit."

**[0231]** The data distribution system of the first to third embodiments to receive reproduction information distribution from the distribution server are described so that authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> and {K<sub>Pp</sub>(1)}K<sub>Pma</sub> of the memory card and cellular phone (content reproduction circuit) are transmitted to the distribution server (step S104), and received at the distribution server (step S106), decrypted using authentication key K<sub>Pma</sub> (step S108), and then conducting an authentication process with respect to both the memory card and cellular phone (content reproduction circuit) according to the decryption result. However, based on the fact that i) the content reproduction circuit to reproduce music does not necessarily have to be the cellular phone receiving distribution since the memory card is detachable, and ii) in reproduction, an authentication process of authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> of the content reproduction circuit of the output destination is carried out in providing a portion of the reproduction information (license key K<sub>c</sub> and reproduction circuit restriction information AC) from the memory card so that the security will not be degraded even if an authentication process of authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> of the content reproduction circuit in the distribution server does not have to be carried out, a structure can be implemented in which the authentication process by authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> of the content reproduction circuit of the distribution server is not carried out.

**[0232]** In this case, the cellular phone transmits the content ID, memory card authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> and license purchase condition data AC at step S104. The distribution server transmits the content ID, memory card authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> and license purchase condition data Ac at step S106, and authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> is decrypted using authentication key K<sub>Pma</sub> to accept public encryption key K<sub>Pm</sub>(1) at step S108. Then, at step S110, an authentication process to determine whether public encryption key K<sub>Pm</sub>(1) has been output from a proper apparatus is conducted by authentication of the authentication server based on the decrypted result. The subsequent process is to be carried out according to the authentication result based on authentication data {K<sub>Pm</sub>(1)}K<sub>Pma</sub> of the memory card. There is no change in the reproduction process.

**[0233]** In the above description, storage of the distribution information is effected by a memory card. However, the present invention is not limited to such a case. More specifically, the present invention is applicable to a more general recording apparatus as long as the function of recording and encryption or the like similar to that of a memory card as described above is possessed. Here, the recording apparatus is not limited to a structure such as a memory card that is detachable from a communication device such as the cellular phone, and

may be incorporated into a communication device.

**[0234]** Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

## Claims

1. A memory card (110) to receive and record reproduction information associated with reproduction of encrypted content data, including a content key to decrypt said encrypted content data into plaintext, through a communication path, said memory card comprising:

a data communication unit to establish a communication path with a transmission source of said reproduction information to receive said reproduction information transmitted in an encrypted state,

a first storage unit (1415, 1440) to store data associated with said reproduction information applied from said data communication unit, an information extraction unit performing a process of storing data associated with said reproduction information from said data communication unit into said first storage unit, and extracting said reproduction information based on data stored in said first storage unit,

a second storage unit (1460) to record reception log information indicating a processing status of a transmission process of said reproduction information, and

a control unit (1420) to control operation of said memory card,

wherein said control unit controls transmission of said reception log information to said transmission source according to a request.

2. The memory card according to claim 1, wherein said data communication unit comprises

a first key hold unit (1402) storing a first private decryption key to decrypt data preencrypted by a first public encryption key predefined corresponding to said memory card,

a first decryption processing unit (1404) to apply a decryption process, receiving a first symmetric key updated and transmitted for each communication of said reproduction information, and encrypted with said first public encryption key,

a second key hold unit (1416) to store a second public encryption key differing for each said

memory card,  
 a key generation unit (1418) generating a second symmetric key updated for each communication of said reproduction information,  
 a first encryption processing unit (1406) encrypting said second public encryption key and said second symmetric key based on said first symmetric key for output, and  
 a second decryption processing unit (1412) receiving said reproduction information encrypted with said second public encryption key and further encrypted with said second symmetric key to decrypt said reproduction information based on said second symmetric key,  
 said first storage unit storing data based on an output of said second decryption processing unit,

wherein said information extraction unit comprises

a third key hold unit (1421) storing a second private decryption key to decrypt data encrypted with said second public encryption key, and  
 a third decryption processing unit (1422) carrying out a decryption process for said second private decryption key in a procedure of a process of storing data associated with said reproduction information into said first storage unit to a process of extracting said reproduction information.

3. The memory card according to claim 2, wherein said first storage unit comprises

a third storage unit (1415) to store first data which is a portion of said reproduction information including said content key in an encrypted state, and  
 a fourth storage unit (1440) to store second data excluding said portion of data of said reproduction information in a plaintext state,

wherein said information extraction unit stores in said fourth storage unit said second data from a result of a decryption process on an output of said second decryption processing unit by said third decryption processing unit, and

comprises a re-encryption processing unit encrypting a portion of said result of a decryption process on the output of said second decryption processing unit by said third decryption processing unit using said second public encryption key to generate said first data to be stored in said third storage unit.

4. The memory card according to claim 3, wherein said

third storage unit receives and stores said encrypted content data that can be decrypted based on said content key.

5. The memory card according to claim 2, wherein said information extraction unit stores in said first storage unit in plaintext a result of a decryption process on an output of said second decryption processing unit by said third decryption processing unit.

6. The memory card according to claim 5, wherein said first storage unit comprises

a third storage unit (1415) to receive and store said encrypted content data that can be decrypted based on said content key, and  
 a fourth storage unit (1440) to store said reproduction information in plaintext state.

7. The memory card according to claim 2, wherein said memory card further comprises a fifth storage unit (1400) storing authentication data to conduct an authentication process at a transmission source of said reproduction information prior to transmission of said reproduction information,

wherein said reception log information is generated at said transmission source at every transmission of said reproduction information from said transmission source when authenticity of said memory card is verified in said authentication process, and includes communication identify information to identify said transmission and said second symmetric key.

8. The memory card according to claim 2, further comprising a fifth storage unit (1400) storing authentication data to conduct an authentication process at a transmission source of said reproduction information prior to transmission of said reproduction information,

wherein said reception log information includes

communication identify information generated at said communication source at every transmission of said reproduction information transmitted from said transmission source when authenticity of said memory card is verified in said authentication process to identify said transmission,  
 status information indicating status of said reproduction information already received, and  
 said second symmetric key,  
 said memory card further comprising means for generating and providing signature information based on at least said status information and said second symmetric key.

9. The memory card according to claim 8, wherein said first encryption processing unit encrypts said reception log information and said signature information based on said first symmetric key, and

said memory card transmitting to said transmission source said reception log information and said signature information encrypted individually at said first encryption processing unit.

10. The memory card according to claim 1, wherein said reception log information is erased from said second storage unit every time said content key is stored in said first storage unit.

11. The memory card according to claim 1, wherein said reception log information further includes a reception status flag rendered on every time transmission of said content key is requested towards said transmission source, and rendered off every time said content key is stored in said first storage unit.

12. A data distribution system comprising a content data supply apparatus to supply encrypted content data and reproduction information associated with reproduction of encrypted content data, including a content key which is a decryption key to decrypt said encrypted content data into plaintext,

wherein said content data supply apparatus (10) comprises

distribution information hold unit (304) to store said content data and said reproduction information,

a first interface unit (350) to transfer data with an external source,

a first session key generation unit (316) generating a first symmetric key updated for each distribution of said reproduction information to said terminal,

a session key encryption unit (318) encrypting said first symmetric key using a first public encryption key predefined corresponding to a terminal of said user, and applying the encrypted key to said first interface unit,

a session key decryption unit (320) to decrypt a second public encryption key and a second symmetric key transmitted in an encrypted state by said first symmetric key,

a first license data encryption processing unit (326) to encrypt reproduction information to reproduce said encrypted content data using said second public encryption key decrypted by said session key decryption unit,

a second license data encryption processing unit (328) encrypting an output of said first license data encryption processing unit with said second symmetric key, and applying the encrypted output to said first interface unit for distribution,

and

a distribution log information hold unit (306) to record distribution log information indicating a processing status during said distribution process, said distribution system further comprising a plurality of terminals (100) corresponding to a plurality of users, respectively, to receive distribution from said content data supply apparatus via a communication path,

wherein each said terminal comprises

a second interface unit (1104) to transfer data with an external source, and

a data storage unit (110) receiving and storing said encrypted content data and said reproduction information,

said data storage unit including

a first key hold unit (1402) to store a first private decryption key to decrypt data encrypted with a first public encryption key predefined corresponding to said data storage unit,

a first decryption processing unit (1404) to apply a decryption process, receiving a first symmetric key updated and transmitted for each communication of said reproduction information, and encrypted with said first public encryption key,

a second key hold unit (1416) to store a second public encryption key differing for each said data storage unit,

a key generation unit (1418) generating a second symmetric key updated for each communication of said reproduction information,

a first encryption processing unit (1406) encrypting said second public encryption key and said second symmetric key based on said first symmetric key for output, and

a second decryption processing unit (1412) receiving said reproduction information encrypted with said second public encryption key and further encrypted with said second symmetric key to decrypt said reproduction information based on said second symmetric key,

a first storage unit (1415, 1440) to store data based on an output of said second decryption processing unit,

a third key hold unit (1421) storing a second private decryption key to decrypt data encrypted with said second public encryption key,

a third decryption processing unit (1422) performing a decryption process for said second private decryption key in a procedure of a process of storing data associated with said reproduction information into said first storage unit to a process of extracting said reproduction information,

a second storage unit (1460) to record said encrypted content data and reception log information indicating a processing status in a distribution process of said reproduction information, and  
 a reception control unit (1420) controlling data transfer with an external source,

5

wherein said reception control unit controls a redistribution process based on said reception log information when said communication path is cut during said distribution process.

10

13. The data distribution system according to claim 12, wherein said data storage unit is a memory card detachable from said terminal.

15

14. The data distribution system according to claim 13, wherein said content data supply apparatus further comprises

20

means (132) for verifying authenticity of said memory card by authentication data transmitted from said memory card, prior to distribution of said reproduction information, and  
 means (315) for generating distribution identify information to identify a distribution process every time a distribution process of said reproduction information is carried out,  
 said memory card further comprising a third storage unit (1460) storing said authentication data,  
 said reception log information including communication identify information to identify said transmission and said second symmetric key, generated at said transmission source at every communication of said reproduction information transmitted from said transmission source when authenticity of said memory card is verified in said authentication process.

25

30

35

40

15. The data distribution system according to claim 12, wherein said reception log information is erased from said second storage unit every time said reproduction information is stored in said first storage unit.

45

16. The data distribution system according to claim 12, wherein said reception log information includes a reception status flag rendered on every time distribution of said reproduction information is requested to said transmission source, and rendered off every time said reproduction information is stored in said first storage unit.

50

55

17. The data distribution system according to claim 12, wherein said reception log information includes at least said communication identify information and

said second symmetric key.

FIG.1

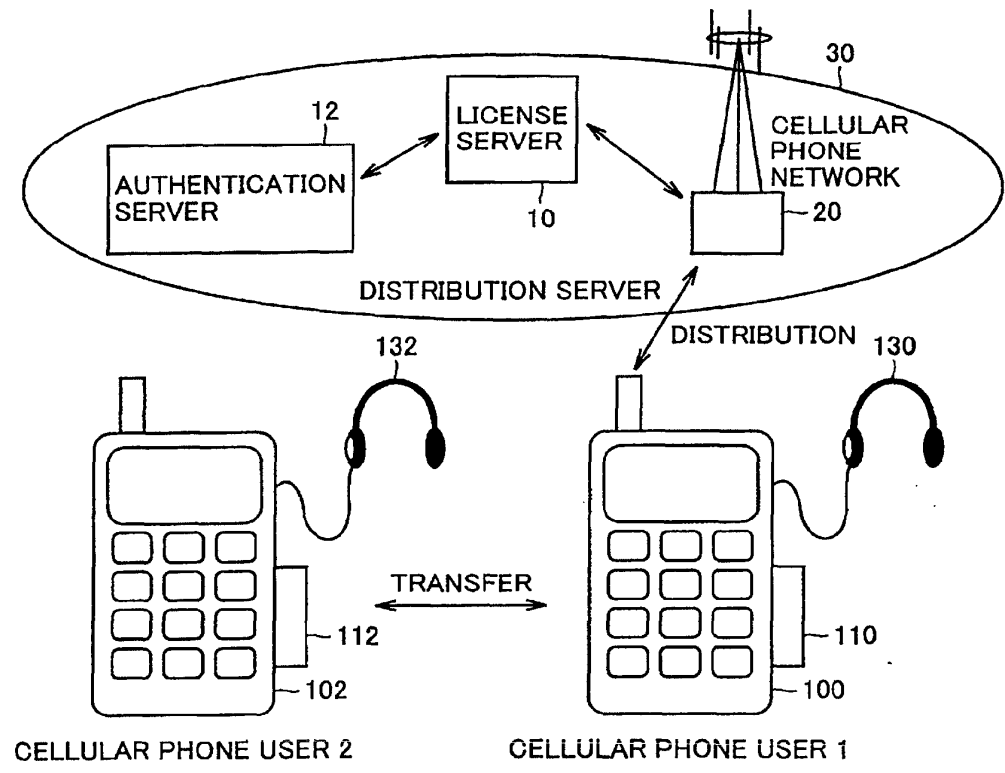




FIG.2

LABEL	FUNCTION-FEATURE	STORAGE-GENERATION SITE
Data	CONTENT DATA, DISTRIBUTED IN THE FORM OF [Data]K <sub>c</sub> AS ENCRYPTED CONTENT DATA SUBJECTED TO ENCRYPTION THAT CAN BE DECRYPTED WITH K <sub>c</sub>	DISTRIBUTION SERVER
Data-inf	ADDITIONAL INFORMATION, INFORMATION IN PLAINTEXT RELATED TO COPYRIGHT OF CONTENT DATA OR SERVER ACCESS	DISTRIBUTION SERVER
K <sub>c</sub>	CONTENT DECRYPTION KEY	DISTRIBUTION SERVER
K <sub>p</sub> (n)/K <sub>mc</sub> (n)	DECRYPTION KEY DEPENDING ON CONTENT REPRODUCTION/MEDIA CLASS (TYPE OR THE LIKE)	CELLULAR PHONE MEMORY CARD
K <sub>Pp</sub> (n)/K <sub>Pmo</sub> (n)	ASYMMETRIC ENCRYPTION KEY THAT CAN BE DECRYPTED USING K <sub>p</sub> /K <sub>mc</sub> , HAVING AUTHENTICATION FUNCTION, RECORDED IN THE FORM OF [K <sub>Pp</sub> ]K <sub>Pma</sub> /[K <sub>Pp</sub> ]K <sub>Pma</sub> AT THE TIME OF SHIPMENT	CELLULAR PHONE MEMORY CARD
K <sub>com</sub>	SECRET COMMON KEY COMMON TO REPRODUCTION CIRCUIT, USED FOR DECRYPTION OF ENCRYPTED K <sub>c</sub> AC2 (ASYMMETRIC, DISTRIBUTION SERVER K <sub>Pcom</sub> /REPRODUCTION CIRCUIT K <sub>com</sub> ALSO AVAILABLE)	DISTRIBUTION SERVER CELLULAR PHONE
K <sub>Pma</sub>	AUTHENTICATION KEY COMMON TO SYSTEM (PUBLIC)	DISTRIBUTION SERVER
AC	PURCHASE CONDITION OF LICENSE FROM USER SIDE (FUNCTION RESTRICTION, NUMBER OF LICENSES, ETC)	CELLULAR PHONE
AC1	INFORMATION AS TO RESTRICTION ON MEMORY ACCESS	DISTRIBUTION SERVER
AC2	CONTROL INFORMATION OF REPRODUCTION CIRCUIT	DISTRIBUTION SERVER
K <sub>m</sub> (i)	DECRYPTION KEY UNIQUE TO EACH MEMORY CARD (i IS IDENTIFIER TO IDENTIFY CARD)	MEMORY CARD
K <sub>Pm</sub> (i)	ASYMMETRIC PUBLIC ENCRYPTION KEY THAT CAN BE DECRYPTED WITH K <sub>m</sub> (i)	MEMORY CARD
K <sub>s1</sub>	SYMMETRIC KEY UNIQUE TO SESSION, GENERATED FOR EACH DISTRIBUTION SESSION	DISTRIBUTION SERVER
K <sub>s2</sub>	SYMMETRIC KEY UNIQUE TO SESSION, GENERATED FOR EACH DISTRIBUTION/TRANSFER (RECEPTION) SESSION	MEMORY CARD
K <sub>s3</sub>	SYMMETRIC KEY UNIQUE TO SESSION, GENERATED FOR EACH REPRODUCTION SESSION	MEMORY CARD
K <sub>s4</sub>	SYMMETRIC KEY UNIQUE TO SESSION, GENERATED FOR EACH REPRODUCTION SESSION	CELLULAR PHONE
CONTENT ID	CODE TO IDENTIFY CONTENT DATA Data	DISTRIBUTION SERVER
TRANSACTION ID	CODE TO IDENTIFY LICENSE ISSUE, (MAY INCLUDE CONTENT ID FOR IDENTIFICATION)	DISTRIBUTION SERVER

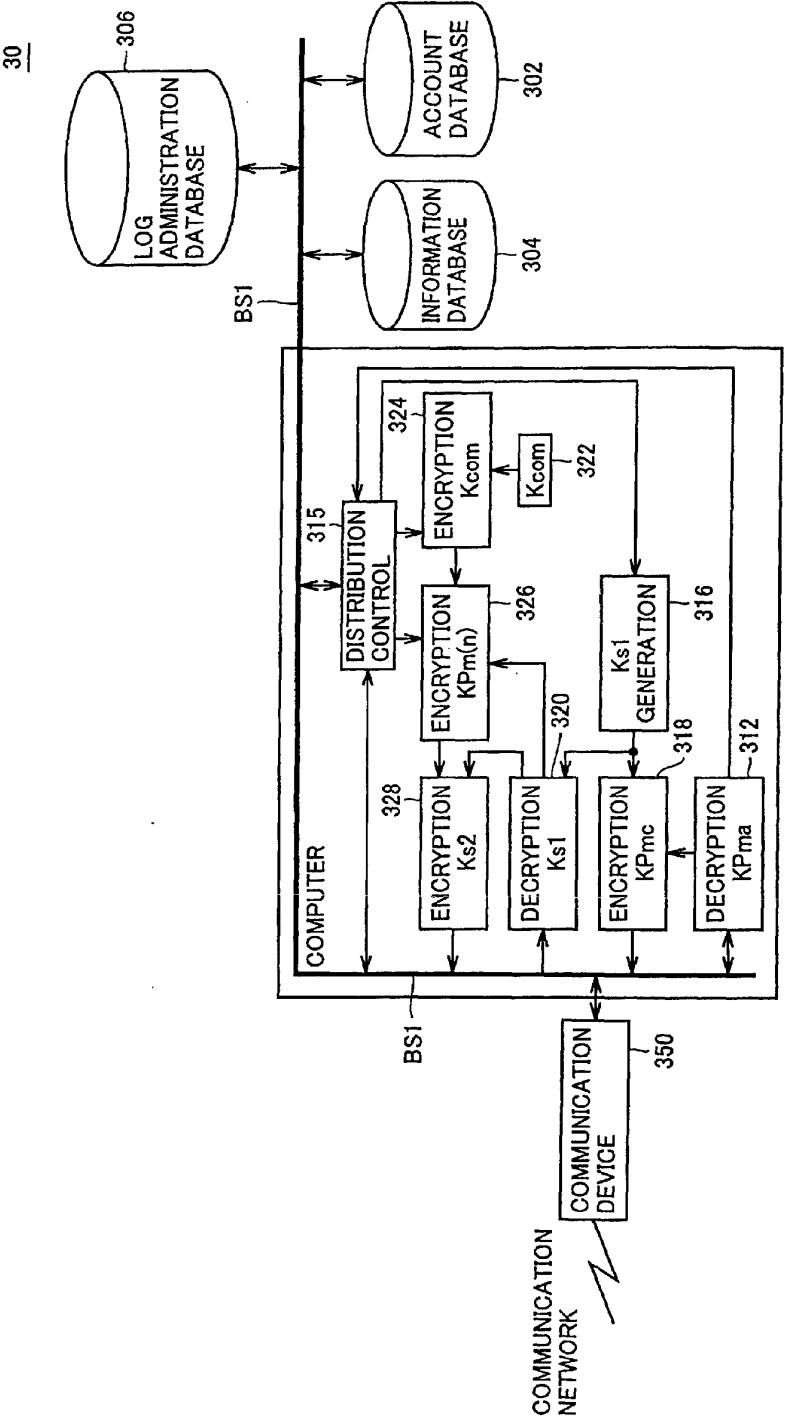
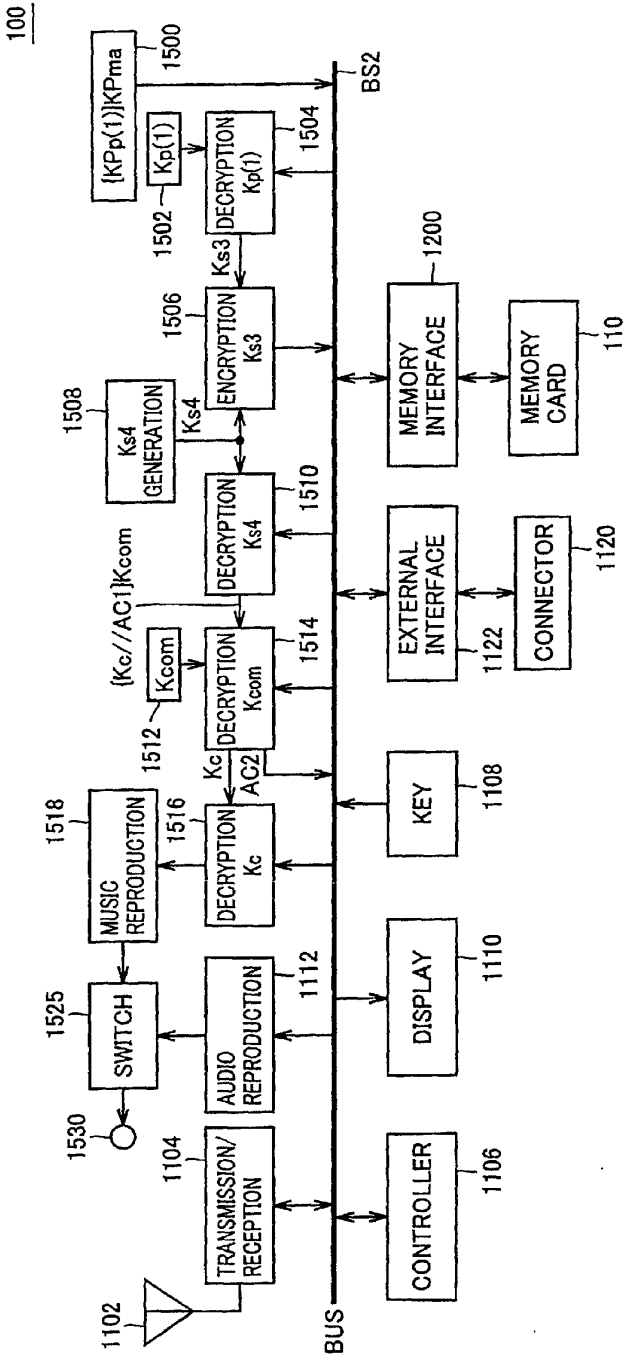


FIG.3

FIG.4



**FIG. 5**

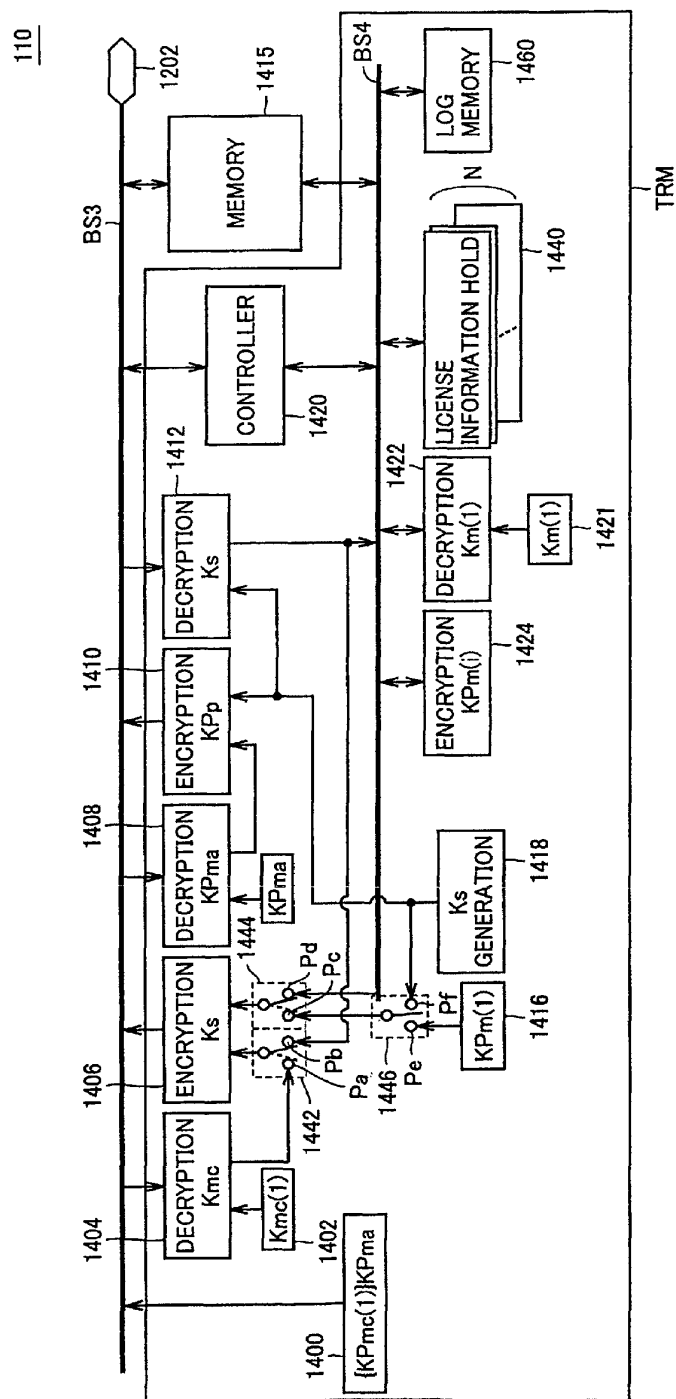
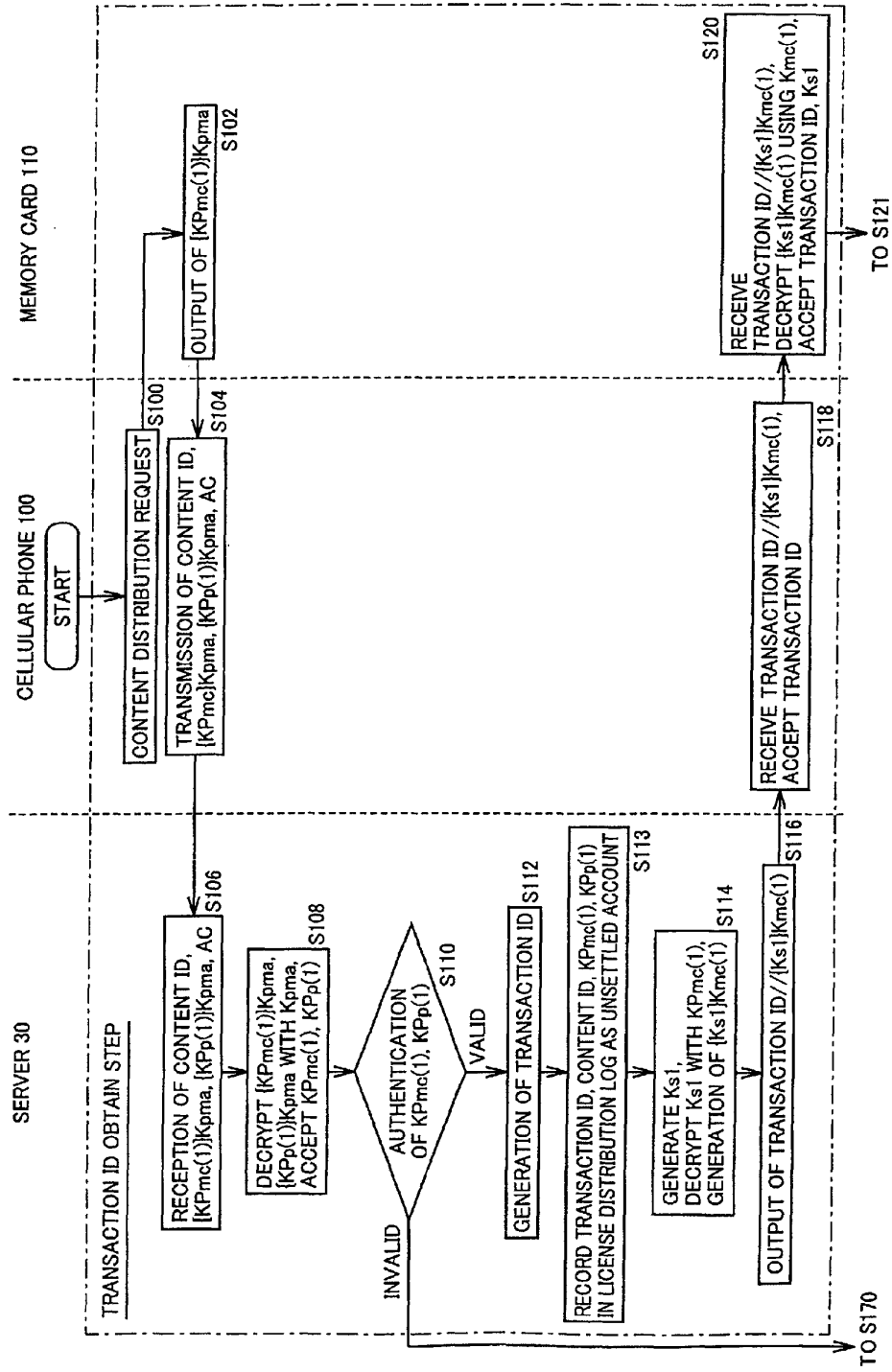


FIG. 6



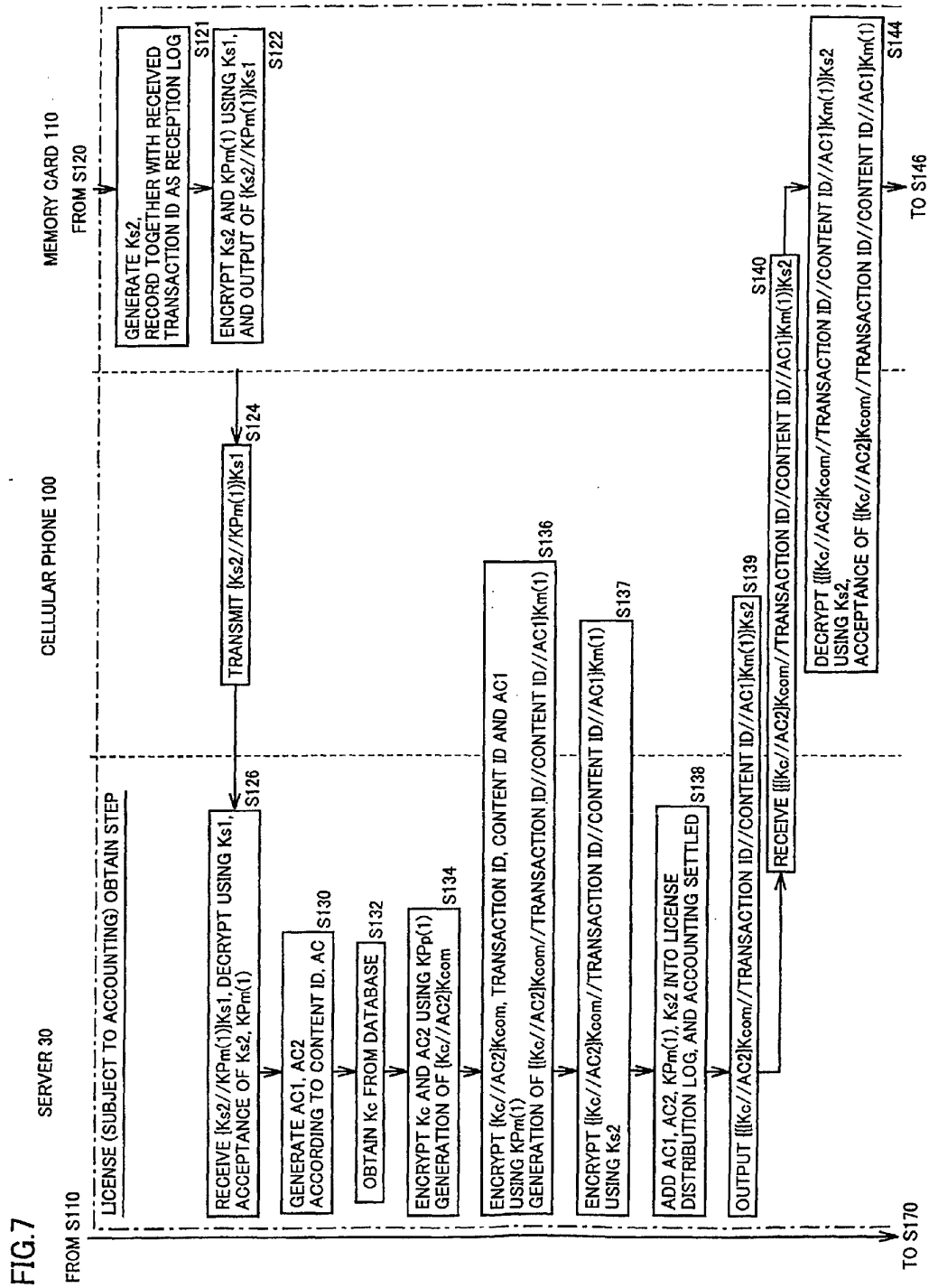


FIG. 8

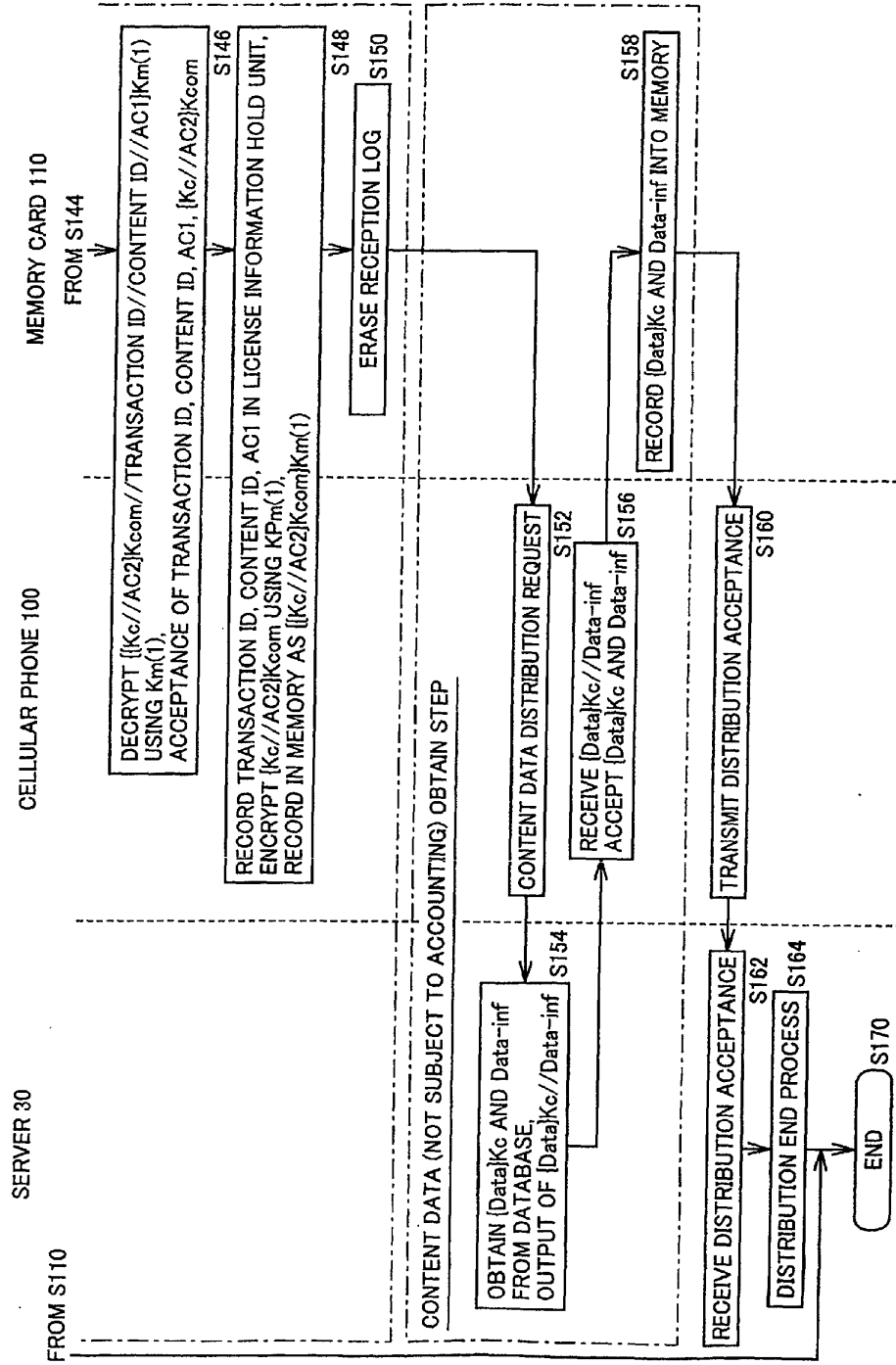
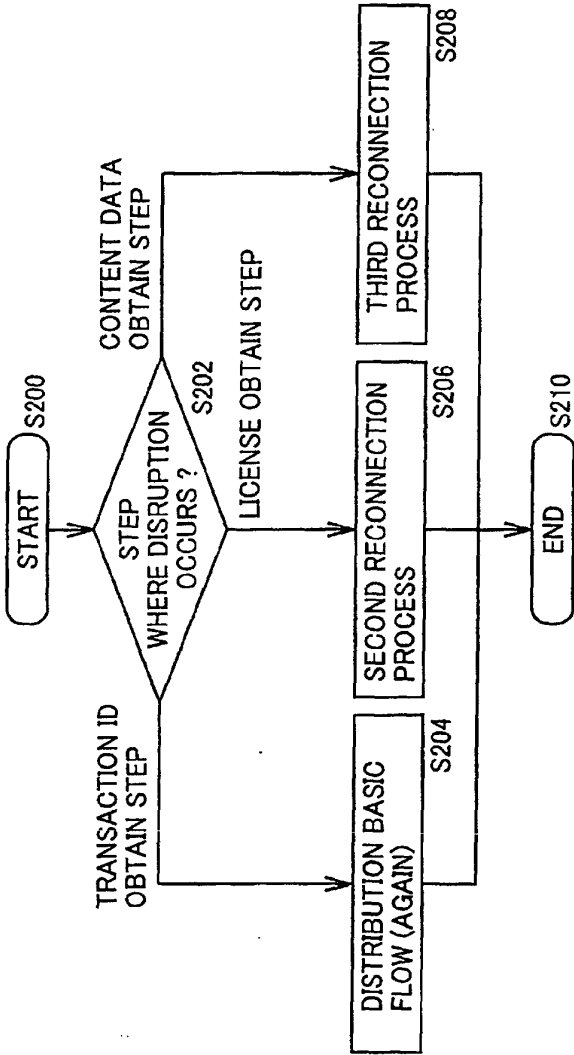


FIG.9





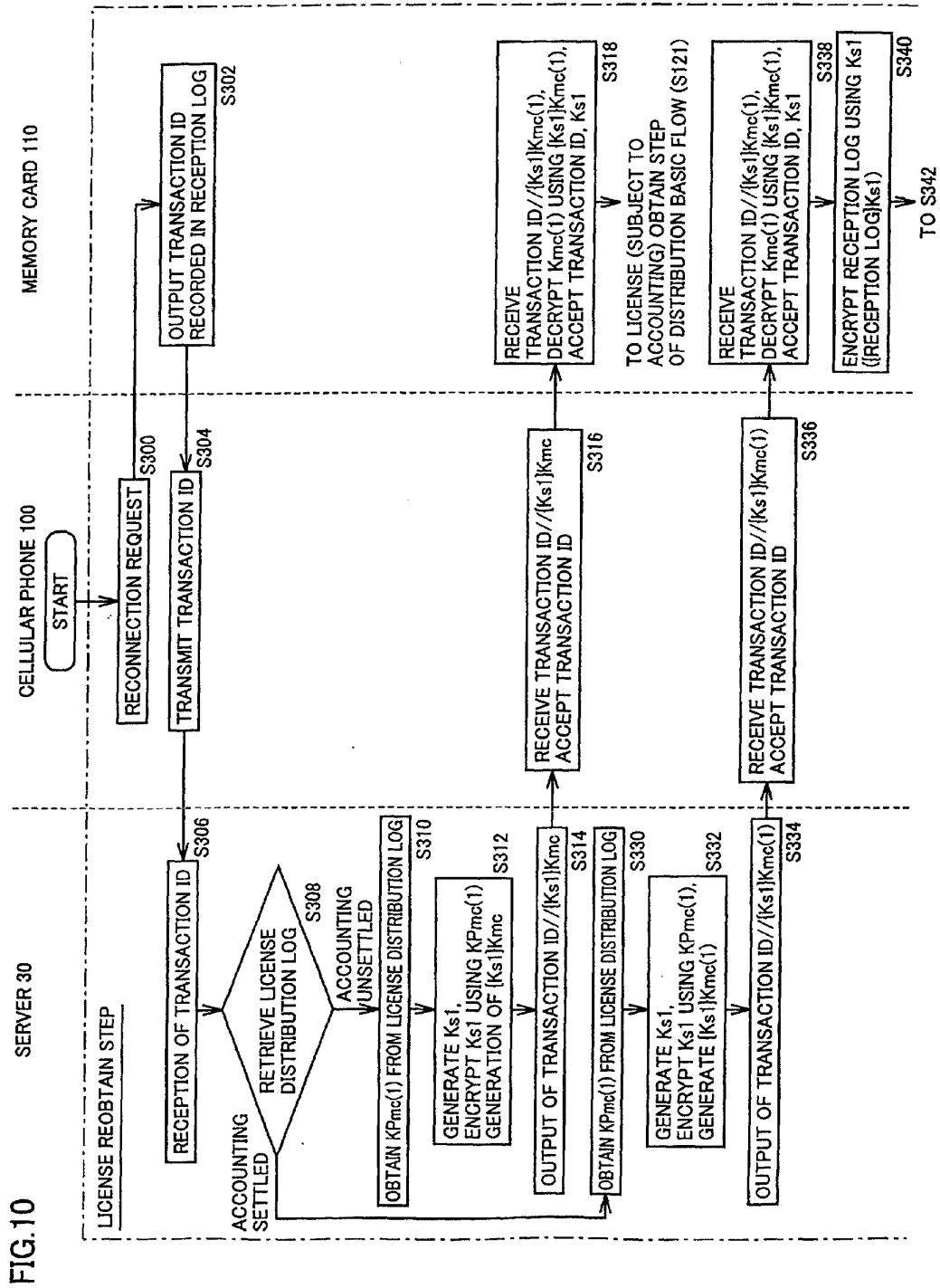
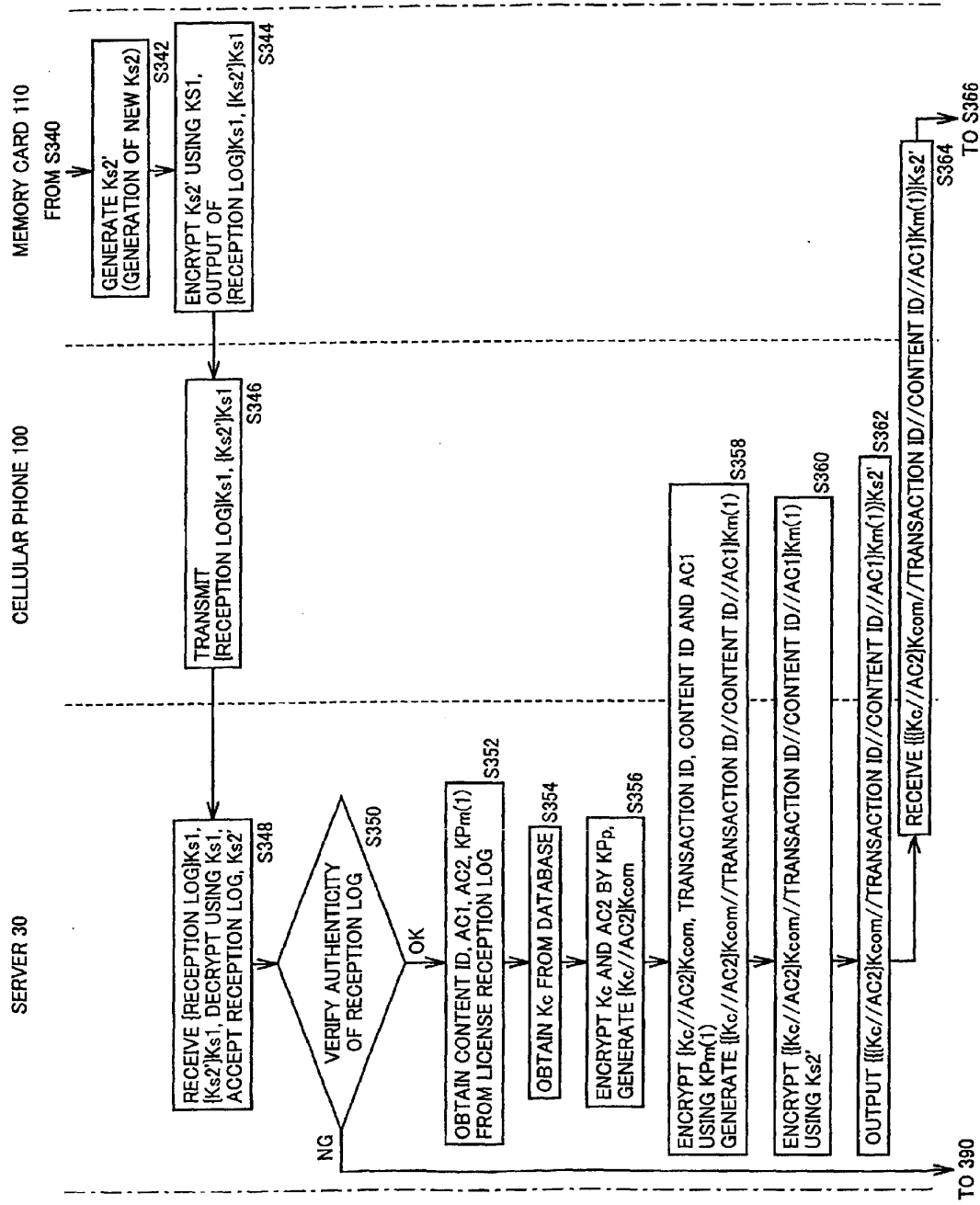


FIG. 11



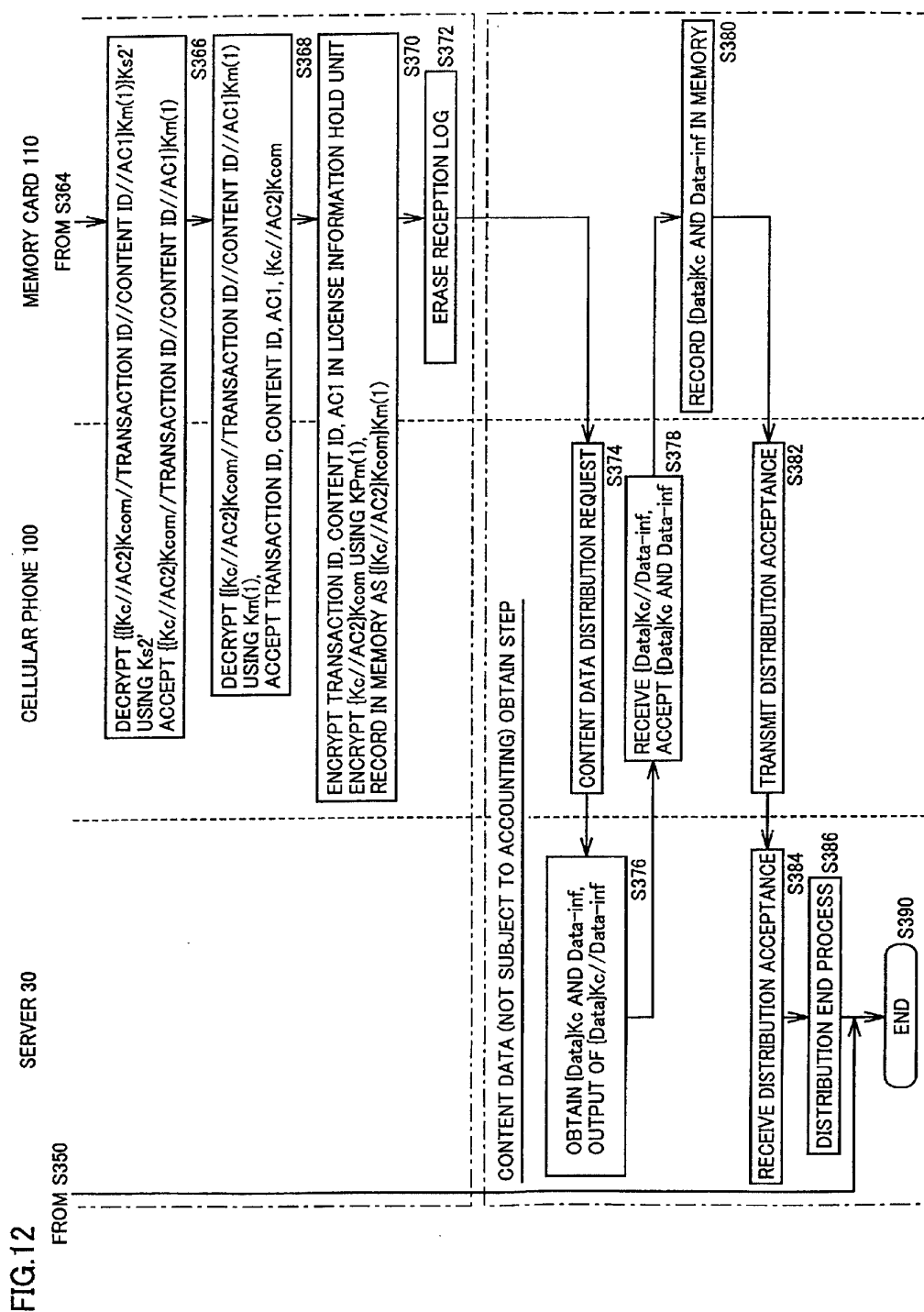


FIG.13

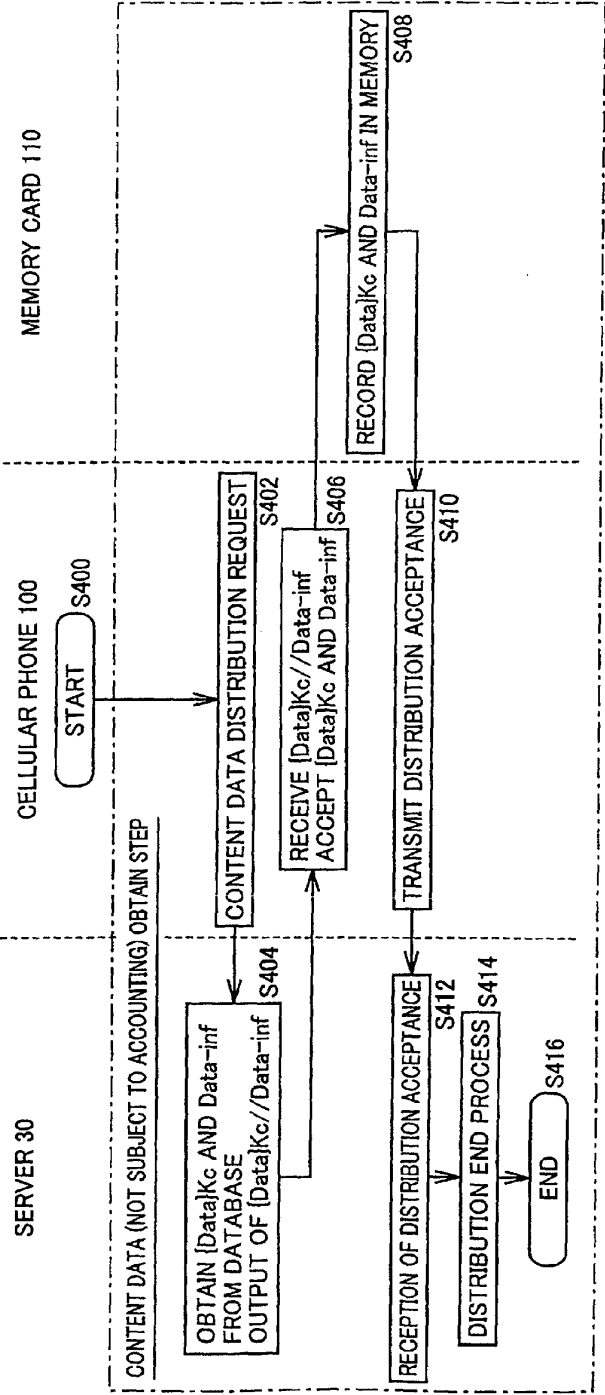


FIG.14

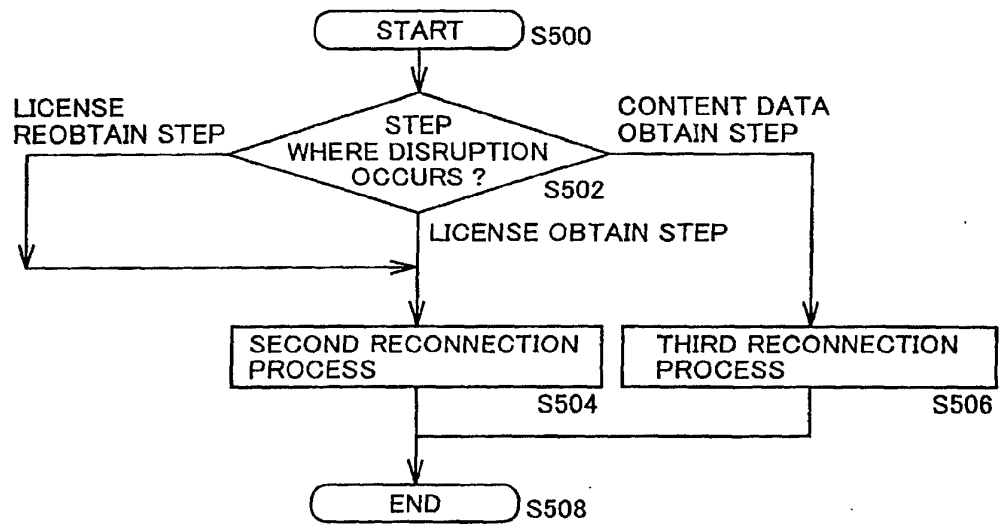


FIG.15

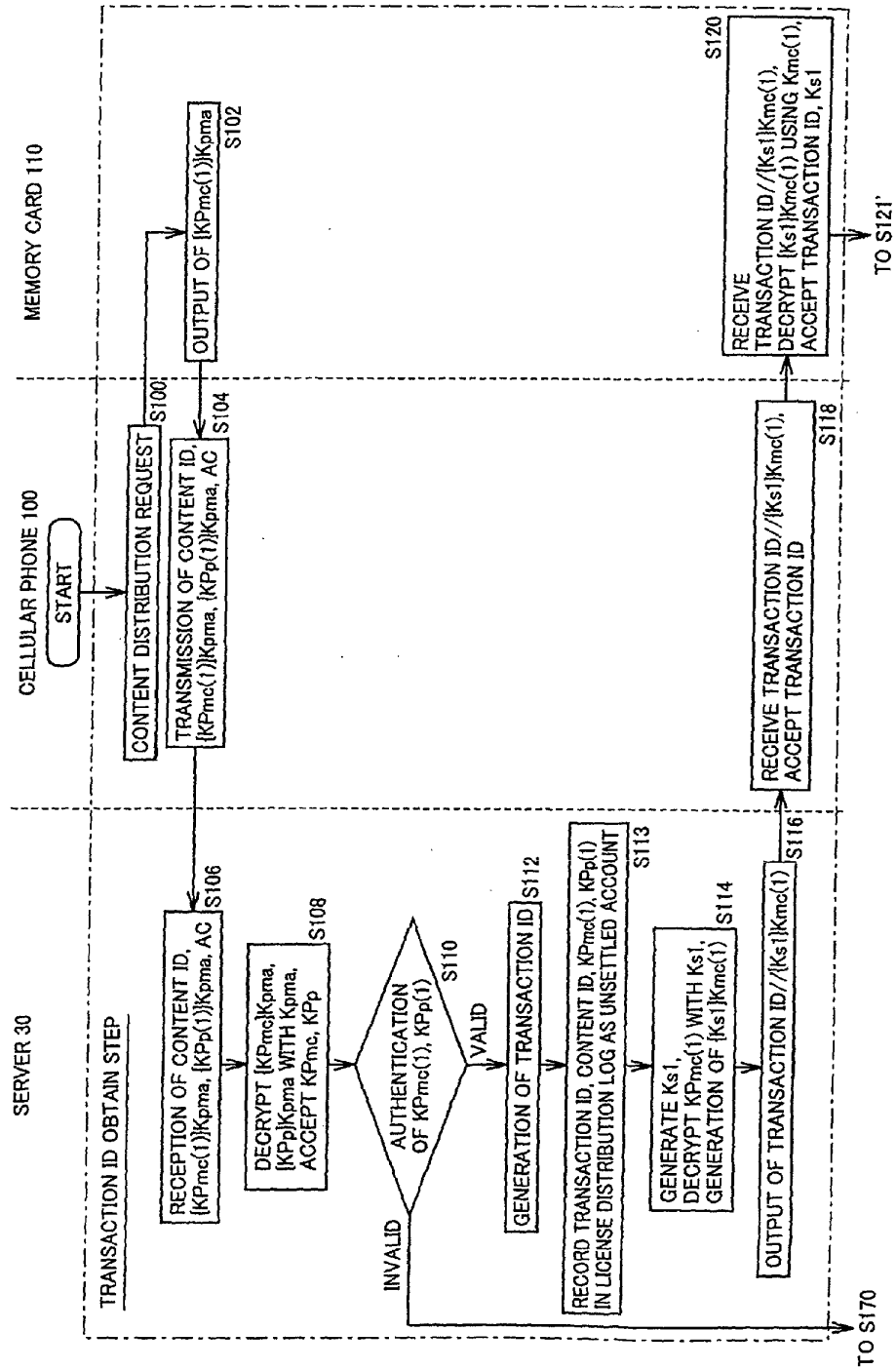
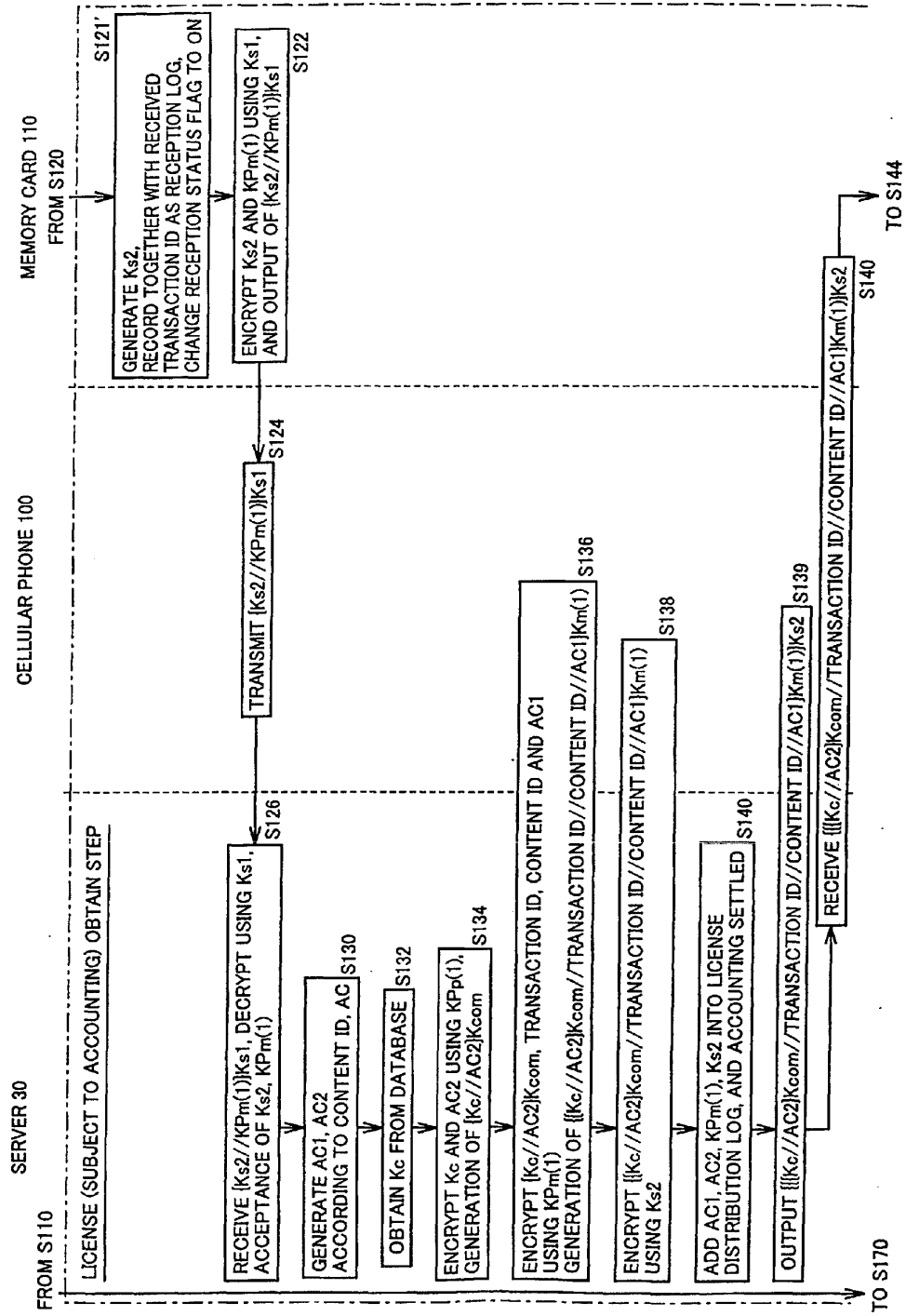
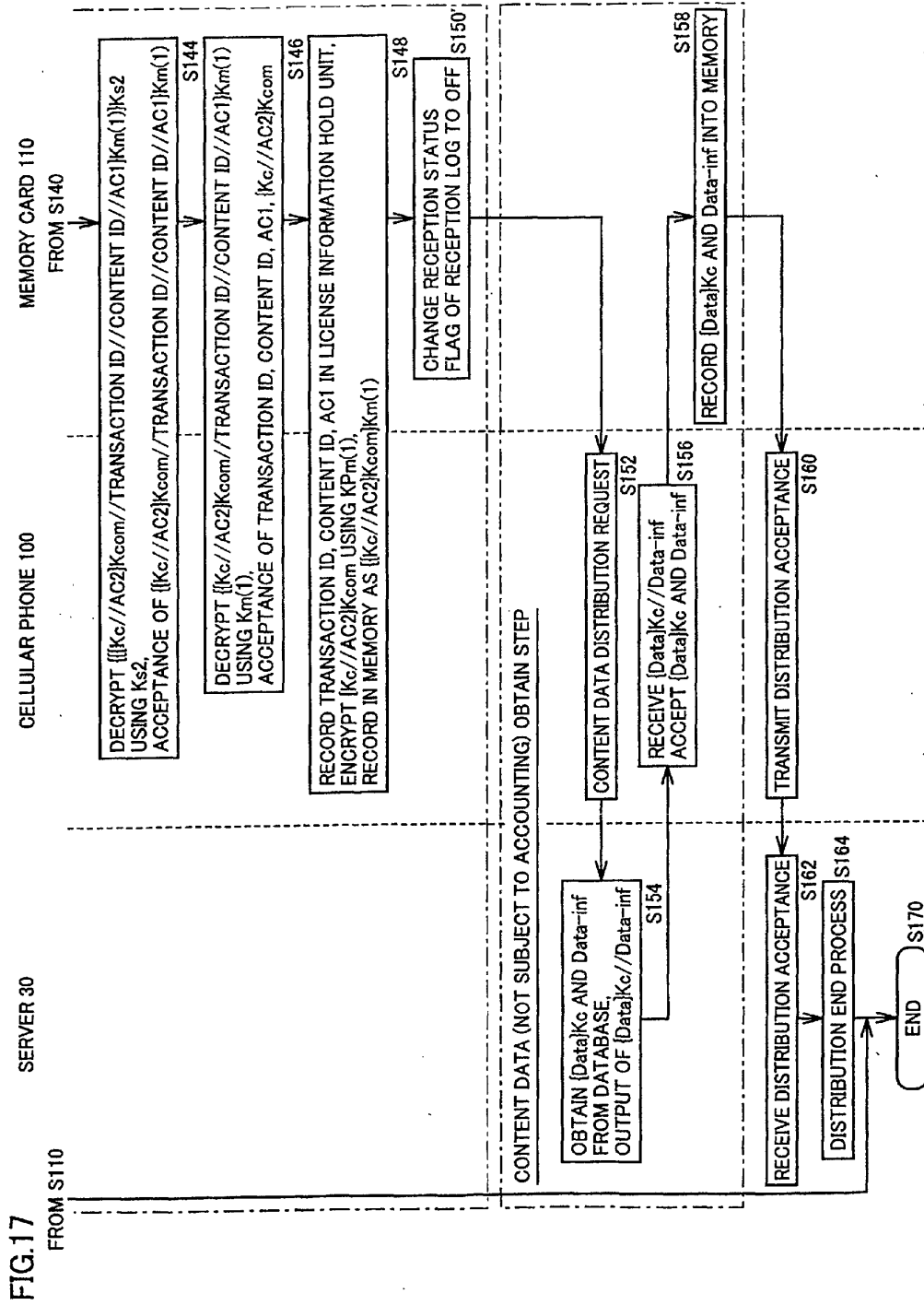


FIG.16







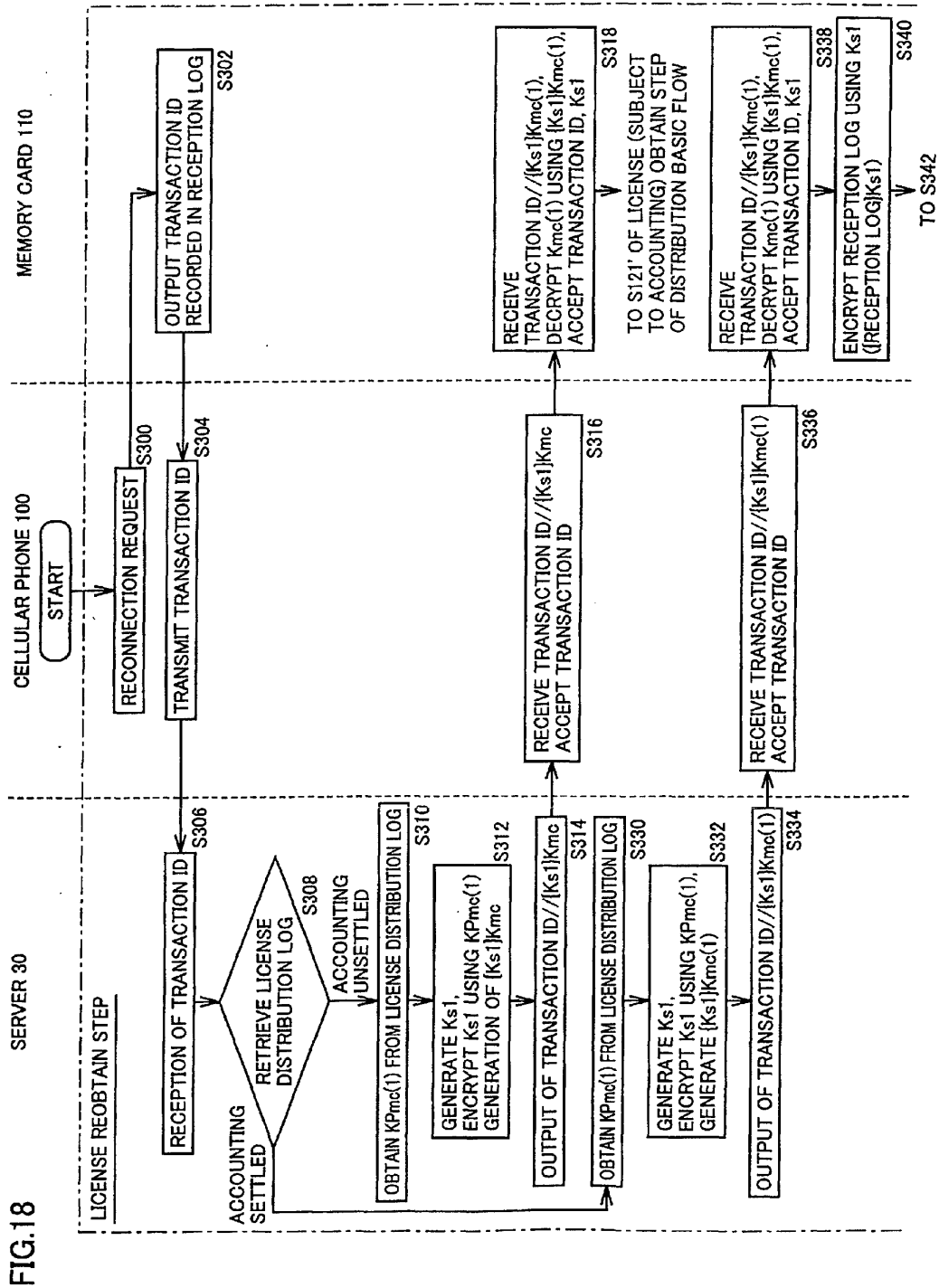
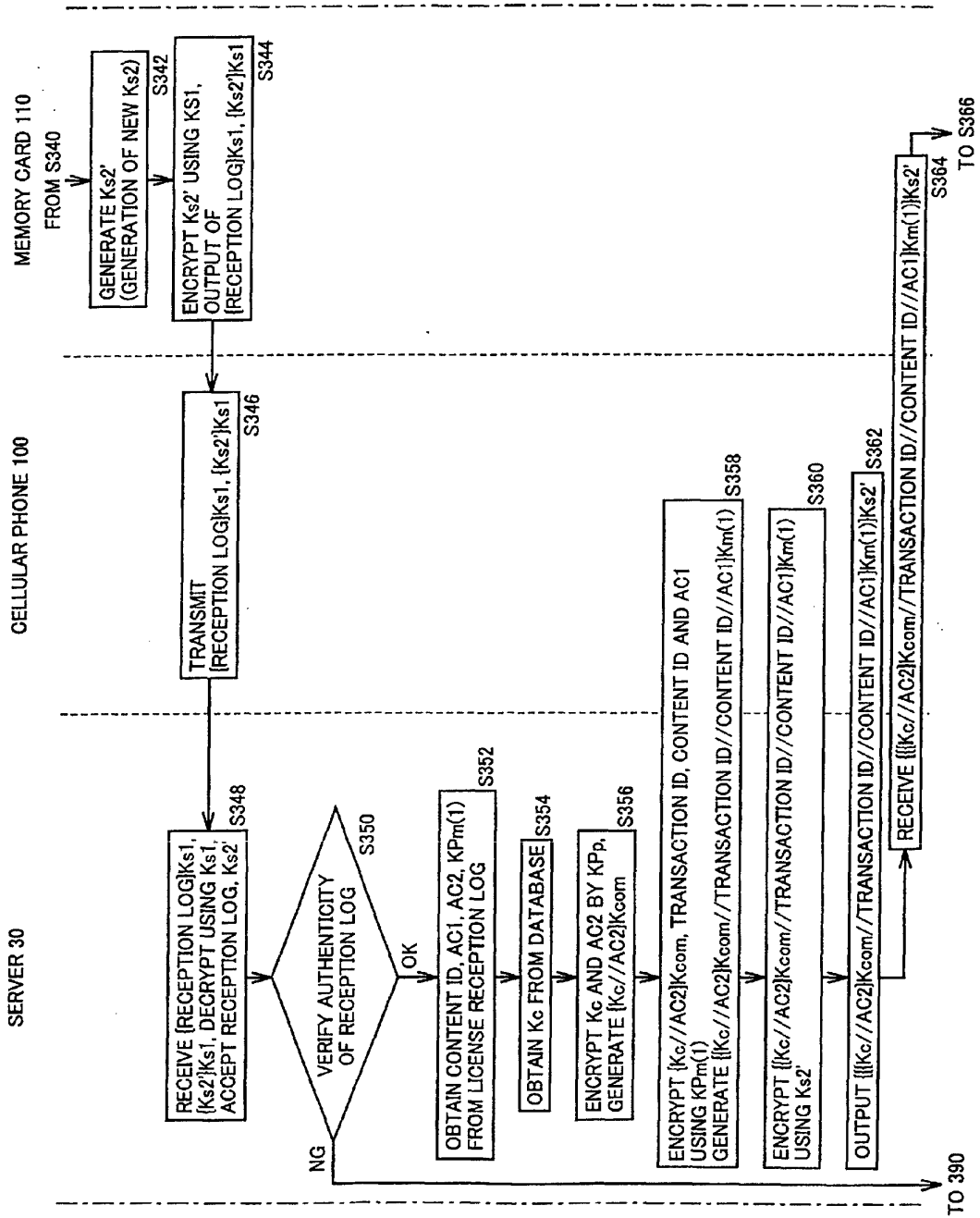


FIG.19



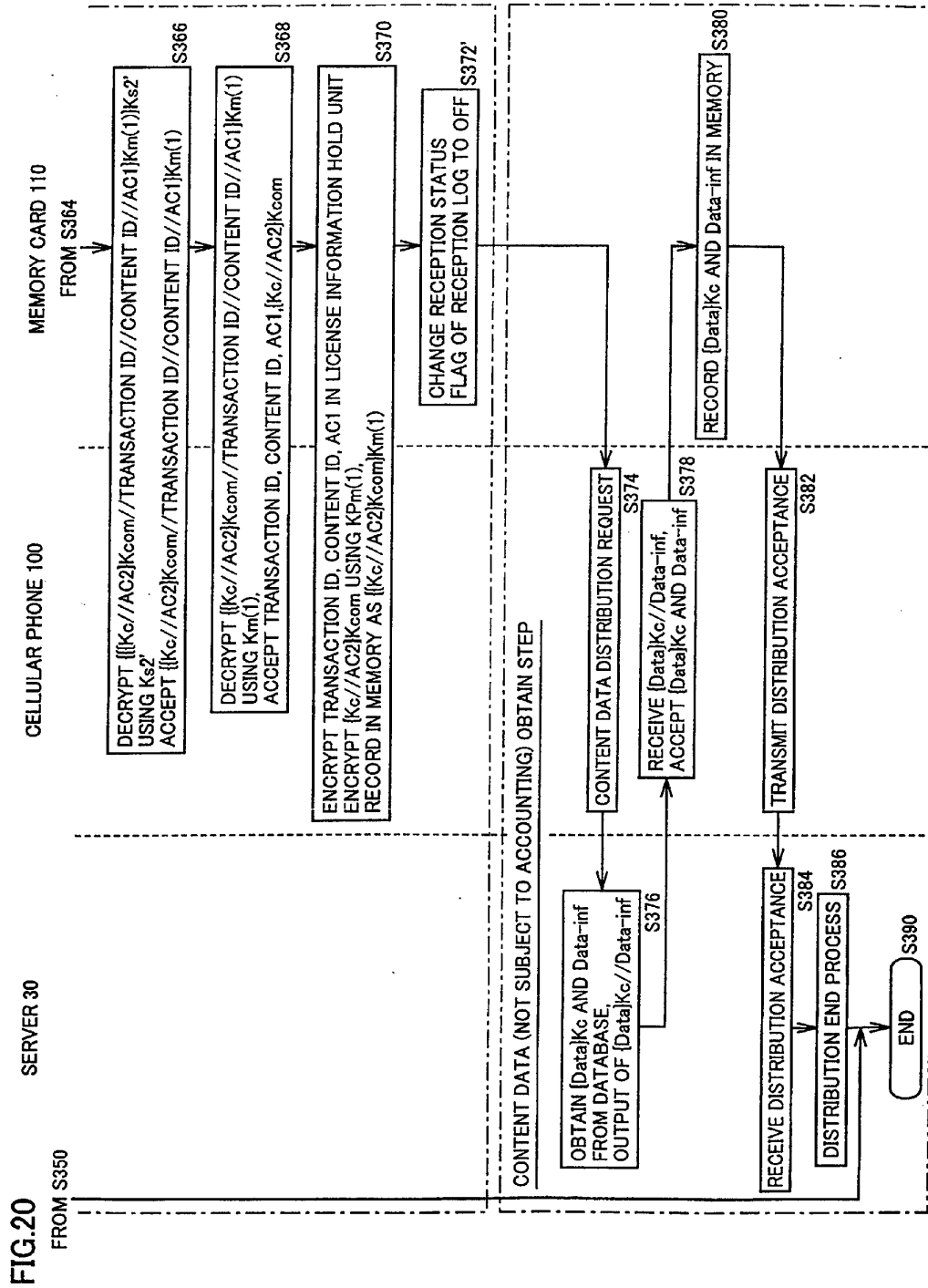


FIG.21

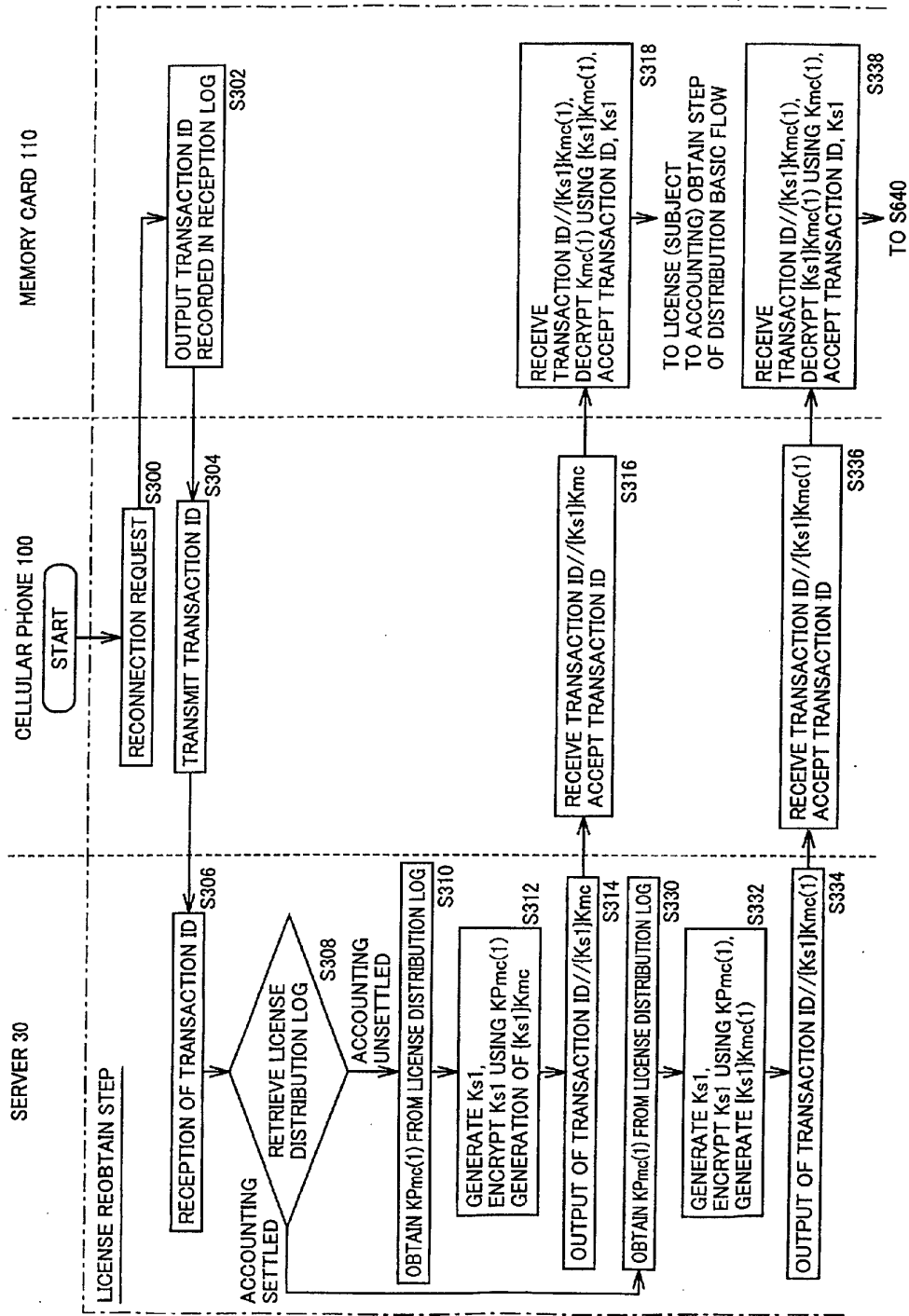


FIG.22

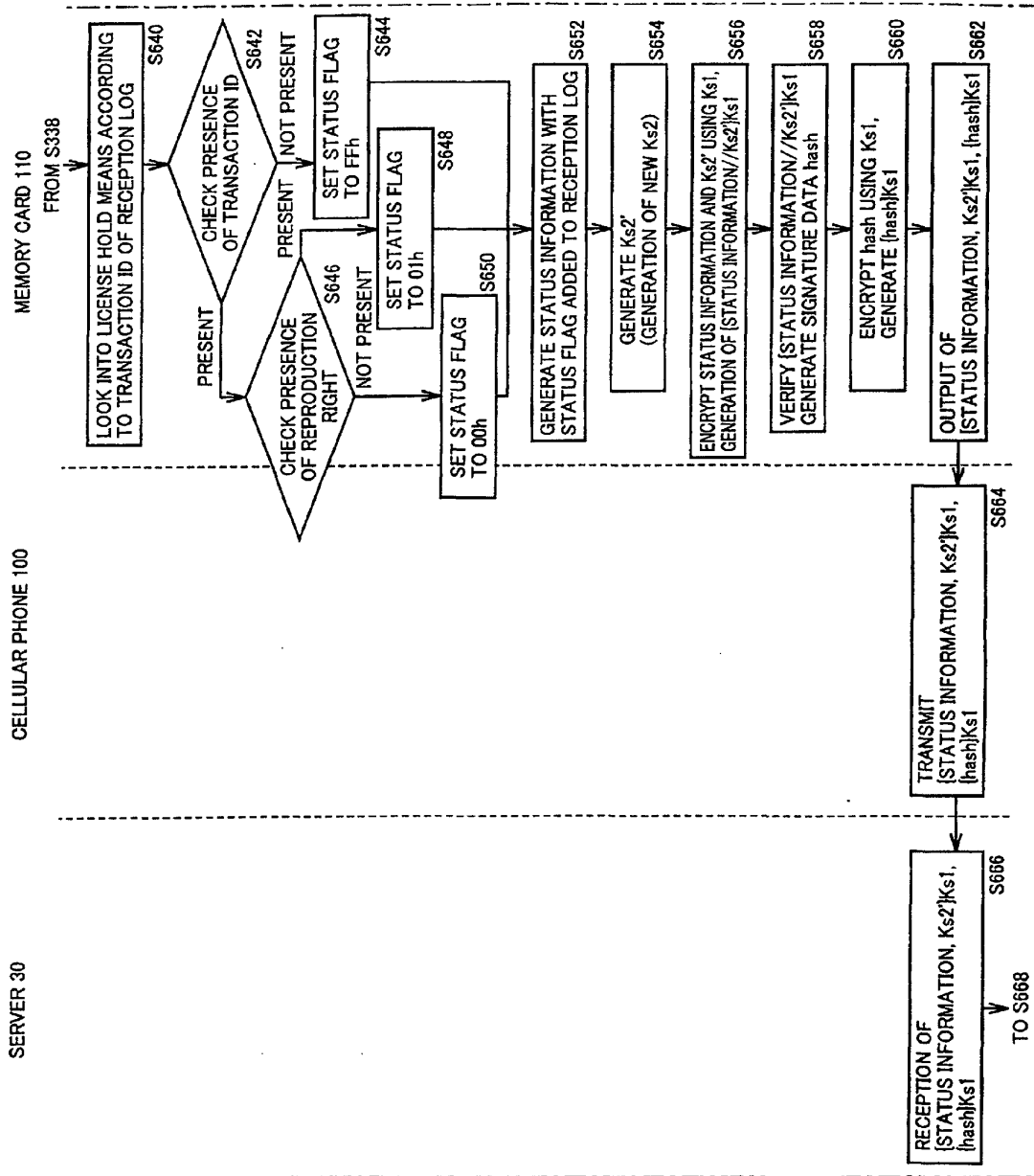
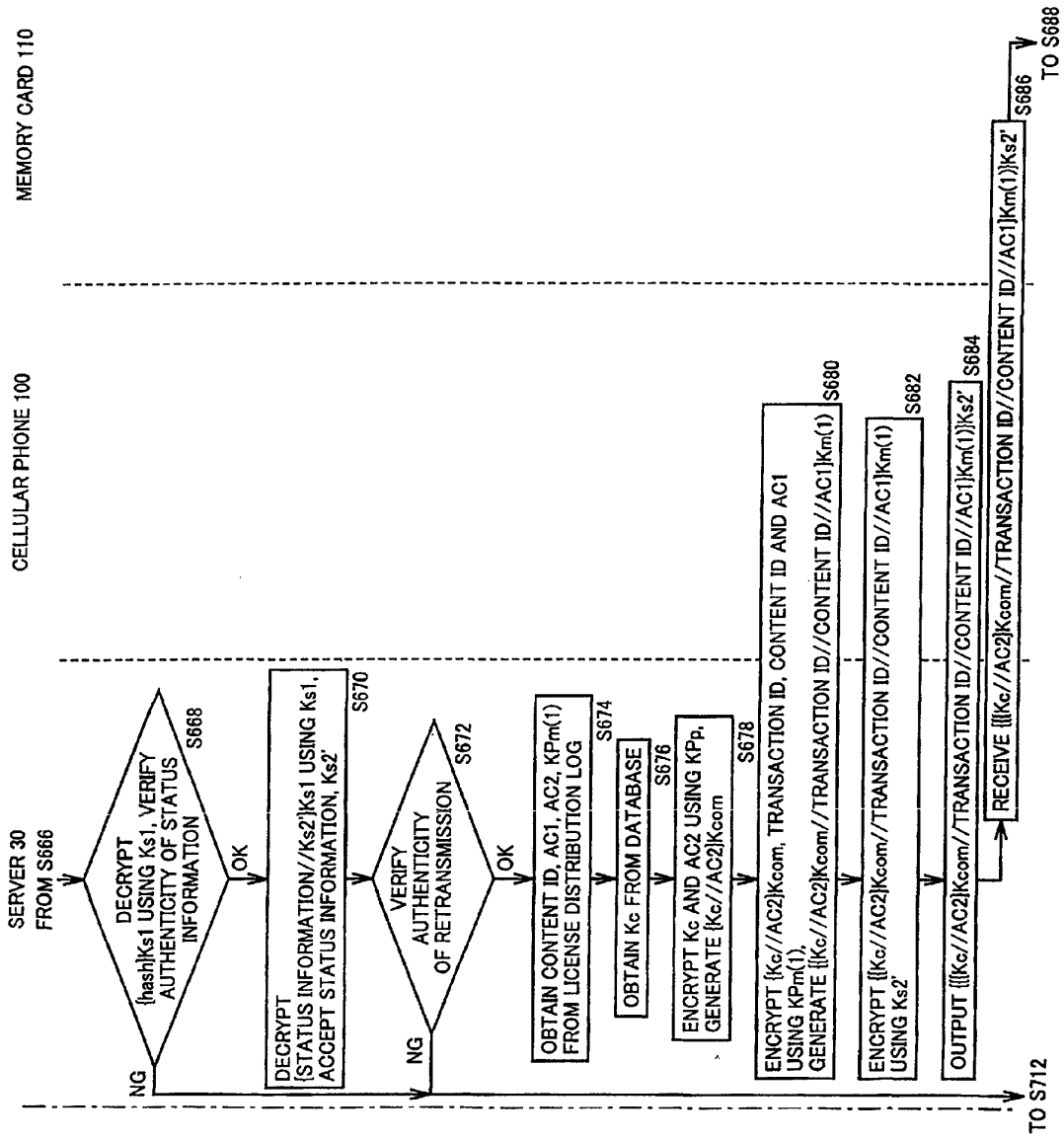
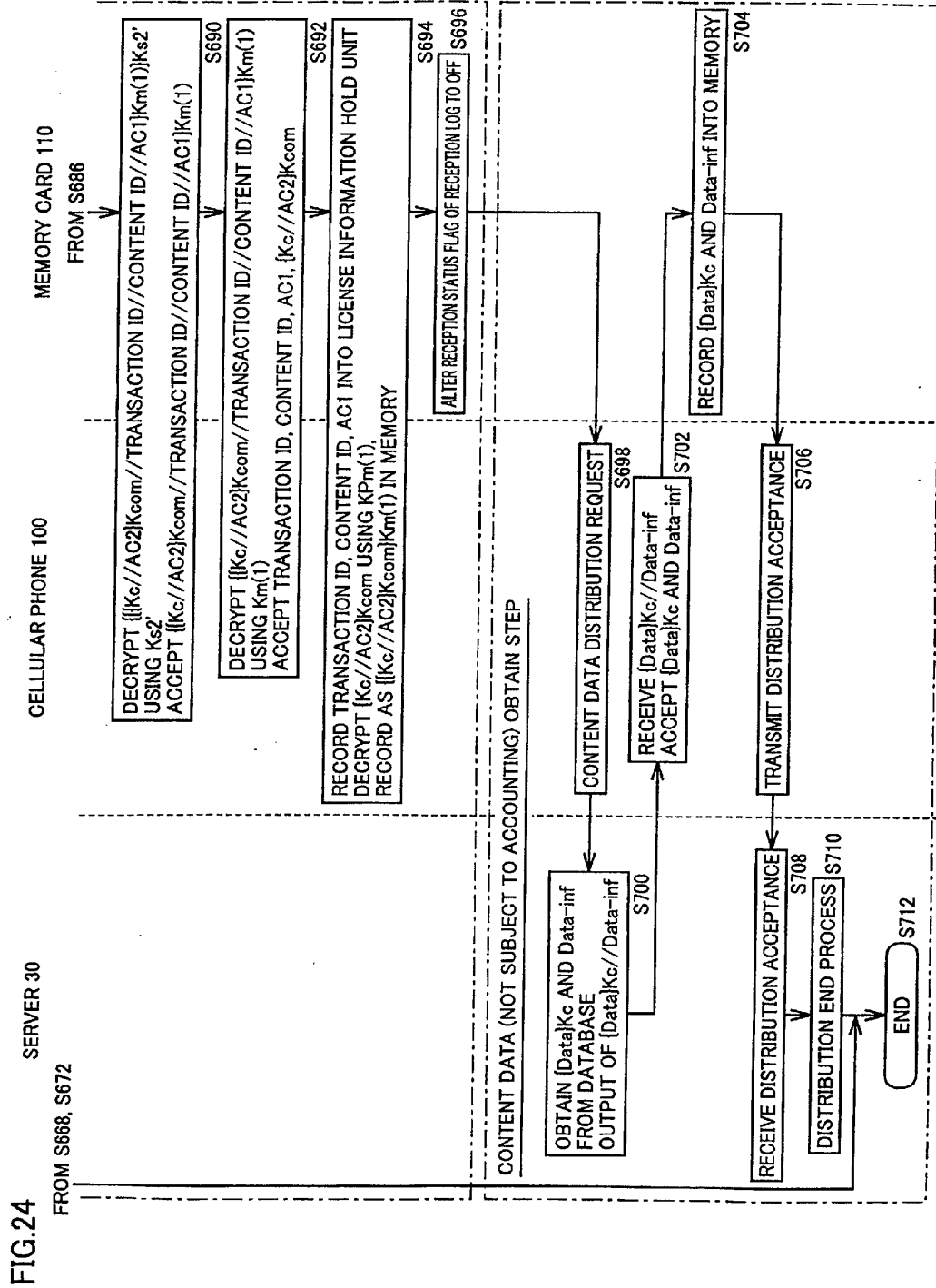


FIG.23





## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP00/08544

A. CLASSIFICATION OF SUBJECT MATTER Int.Cl. <sup>7</sup> H04L 9/10 G06F 12/14, G10K 15/02 G06F 13/00		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) Int.Cl. <sup>7</sup> H04L 9/00 H04K 1/00-3/00 G09C 1/00-5/00 G06F 12/00-13/00 G10K 15/00		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001 Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) JICST FILE (JOIS) WPI (DIALOG) INSPEC (DIALOG)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP, 11-224288, A (Hitachi, Ltd.), 17 August, 1999 (17.08.99) & EP, 935209, A2 & SG, 75914, A1 & CA, 2260536, A1	1-17
EA	JP, 2000-253453, A (Sony Corporation), 14 September, 2000 (14.09.00) & EP, 1033894, A2 & CN, 1266326, A	1-17
EA	JP, 2000-268096, A (Dainippon Printing Co., Ltd.), 29 September, 2000 (29.09.00) (Family: none)	1-17
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"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</p> <p>"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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p>		
Date of the actual completion of the international search 08 March, 2001 (08.03.01)		Date of mailing of the international search report 21 March, 2001 (21.03.01)
Name and mailing address of the ISA/ Japanese Patent Office		Authorized officer
Facsimile No.		Telephone No.

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



(19) World Intellectual Property Organization  
International Bureau



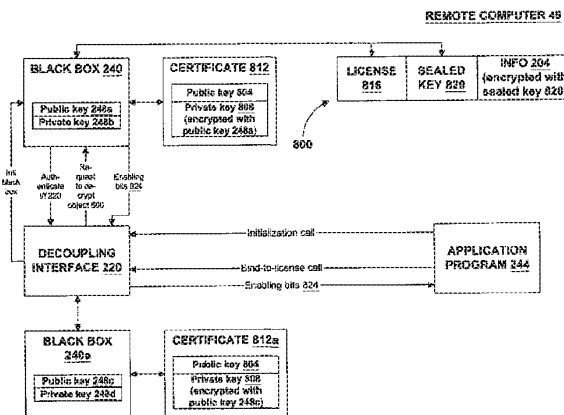
(43) International Publication Date  
3 January 2002 (03.01.2002)

PCT

(10) International Publication Number  
WO 02/01334 A2

- (51) International Patent Classification<sup>7</sup>: G06F 1/00
- (21) International Application Number: PCT/US01/40899
- (22) International Filing Date: 8 June 2001 (08.06.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/604,518 27 June 2000 (27.06.2000) US
- (71) Applicant: MICROSOFT CORPORATION [US/US];  
One Microsoft Way, Redmond, WA 98052 (US).
- (72) Inventors: MANFERDELLI, John, L.; 7921 245th Way NE, Redmond, WA 98053 (US). MARR, Michael, David; 21008 NE 36th Street, Sammamish, WA 98074 (US). KRISHNASWAMY, Vinay; 23319 N.E. 142nd Pl., Woodinville, WA 98072 (US). JAKUBOWSKI, Mariusz, H.; 1840 154th Avenue NE#C-222, Bellevue, WA 98007 (US).
- (74) Agents: ROCCHI, Steven, J. et al.; Woodcock Washburn Kurtz Mackiewicz & Norris LLP, 46th floor, One Liberty Place, Philadelphia, PA 19103 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:  
— without international search report and to be republished upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR INTERFACING A SOFTWARE PROCESS TO SECURE REPOSITORIES



(57) Abstract: A secure repository individualized for a hardware environment and a method and system for providing the same. The secure repository includes a hidden cryptographic key and code that applies the key without requiring access to a copy of the key. The code that implements the secure repository is generated in a manner that is at least partly based on a hardware ID associated with the hardware environment in which the secure repository is to be installed, and may also be based on a random number. Cryptographic functions implemented by the secure repository include decryption of encrypted information and validation of cryptographically signed information. The secure repository may be coupled to an application program, which uses cryptographic services provided by the secure repository, by way of a decoupling interface that provides a common communication and authentication interface for diverse types of secure repositories. The decoupling interface may take the form of a single application programmer interface (API) usable with multiple dynamically linkable libraries.

- 1 -

SYSTEM AND METHOD FOR INTERFACING  
A SOFTWARE PROCESS TO SECURE REPOSITORIES

FIELD OF THE INVENTION

5           The present invention relates generally to the field of computer security and, more particularly, to the interfacing of a software process to secure repositories.

BACKGROUND OF THE INVENTION

10           In the field of computer security, one enduring problem is to create a system that allows an owner of information to electronically distribute the information throughout the world while regulating use of that information on remote hardware over which the information owner has no control. For example, information may be delivered to an end user in encrypted form with the ultimate goal that it be viewed (but not copied or otherwise misused) by the end user. The information requires a key in  
15           order to be decrypted, but it may not be desirable to give the end user unfettered access to the key because the user could then copy the decrypted information and disseminate it at will.

          One solution is not to provide the key directly, but to provide the key to the end user in the form of software that applies the key (or that performs some other  
20           sensitive function to be hidden from the user). Such software may contain various types of protection designed to complicate or resist attempts to analyze or misuse the software or the secrets that the software is designed to protect. However, the drawback to this solution is that attempts to create "secure" software incorporating such resistance have so far proven ineffective, as such software has invariably been misused,  
25           ported to unauthorized installations, or broken in one way or another. A further drawback is that if technology advances to permit greater protection to be built into the software, it is not always possible to "renew" the security technology by replacing an old unit of "secure" software with a new one.

          In view of the foregoing, there is a need for a system that overcomes the  
30           limitations and drawbacks of the prior art.

- 2 -

SUMMARY OF THE INVENTION

The present invention provides advantageous systems and methods for creating and using a software-based secure repository. A black box provided herein is an exemplary secure repository that uses cryptographic techniques, preferably  
5 public/private key techniques, to perform decryption and authentication services in a secure manner that resists discovery of secret keys used by the cryptographic techniques.

A secure repository according to the invention, such as the exemplary "black box" provided herein, functions as the trusted custodian of one or more  
10 cryptographic keys. It performs cryptographic functions such as using a cryptographic key to decrypt information. A user who wishes to use encrypted information is provided with a black box that incorporates the key needed to decrypt the information. The black box contains code that applies the key without actually representing the key in memory, thus shielding the key from discovery by the user. Preferably, the  
15 information is encrypted with a public key pair that is unique (or substantially unique) to the user, and each user obtains a unique black box that applies that particular user's private key. The black box may provide the decrypted information to other software modules which the black box trusts not to misuse the information or to divulge it in an unauthorized way. The black box may use cryptographic authentication techniques to  
20 establish trust with these other software modules.

In order to obtain the black box, the user's computer contacts a black box server, preferably via a network, and uploads a hardware identifier associated with the computer. The black box server creates an "individualized" black box which contains code to apply a particular cryptographic key, where the code (as well as other  
25 code contained in the black box) is based on, and preferably bound to, the hardware identifier. The black box server may introduce various types of protections into the code, such as diversionary code, integrity checks, inline encryption, obfuscated execution, code reorganization, and self-healing code. The black box server then downloads an executable file or executable library containing the black box for  
30 installation on the user's computer.

- 3 -

In a preferred embodiment, the black box interfaces with an application program for which it provides secure functions by way of a decoupling interface that makes the details of the black box transparent to the developer of the application program. The decoupling interface may, for example, be an application programmer interface (API) usable with multiple dynamic-link libraries (DLLs), where a different  
5 DLL is linked to the application program at runtime depending on which black box is being used. A new DLL may be created for a new black box that did not exist at the time the application program was created. The use of a decoupling interface in this manner supports a "renewable" model of security - i.e., the black box can be replaced  
10 if it has been found to be defective or unsecure, or if later technological developments permit the creation of an even more secure black box. The DLL that implements the decoupling interface is an example of a software module that may be authenticated by the black box.

Other features of the invention are described below.

15

#### BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the  
20 drawings exemplary constructions of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

FIG. 1 is a block diagram showing an exemplary computing environment in which aspects of the invention may be implemented;

FIG. 2 is a block diagram showing an exemplary use of a preferred type  
25 of secure repository;

FIG. 3 is a block diagram showing a relationship between a computer that requests a secure repository and a computer that generates a secure repository;

FIG. 4 is a block diagram of an exemplary secure repository generator according to aspects of the invention;

30 FIG. 5 is a block diagram of a cryptographic code generator according to aspects of the invention;

- 4 -

FIG. 6 is a flow diagram showing an exemplary process for creating a machine-individualized secure repository;

FIG. 7 is a flow diagram showing an exemplary process for performing the code creation step of FIG. 6;

5           FIG. 8 is a block diagram showing an exemplary architecture that includes a decoupling interface for use with a secure repository and an application program;

FIG. 9 is a flow diagram showing an exemplary process of using a secure repository.

10

### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

#### Overview

Modern computing and network technology has permitted widespread and low-cost electronic distribution of information. One feature of electronic information is that it is easily copyable and, once it has been delivered to a computer, the operator of that computer may use and disseminate the information in ways that are neither contemplated by, nor consistent with the commercial interests of, the owner of the information. A secure repository may be used to control the use of information on hardware over which the information's owner otherwise has no control.

#### 20   Computing Environment

As shown in FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer or network server 20 or the like, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory 22 to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read-only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer or network server 20 may

- 5 -

further include a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD-ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules and other data for the personal computer or network server 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read-only memories (ROMs) and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35 (e.g., WINDOWS® 2000, WINDOWS NT®, or WINDOWS® 95/98), one or more application programs 36, other program modules 37 and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus 23, but may be connected by other interfaces, such as a parallel port, game port, universal serial bus (USB), or a 1394 high-speed serial port. A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor 47, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

- 6 -

The personal computer or network server 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer 49 may be another personal computer, another network server, a router, a network PC, a peer device or other  
5 common network node, and typically includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in FIG. 1. The logical connections depicted in Fig. 2 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks,  
10 Intranets and the Internet.

When used in a LAN networking environment, the personal computer or network server 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment, the personal computer or network server 20 typically includes a modem 54 or other means for establishing  
15 communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer or network server 20, or portions thereof, may be stored in the remote memory storage device 50. It will be appreciated that the network connections  
20 shown are exemplary and other means of establishing a communications link between the computers may be used.

#### Exemplary Use of a Secure Repository

Referring now to FIG. 2, there is shown an exemplary use of a secure repository in connection with remote computer 49. Remote computer 49 is a computer  
25 that may be able to receive information from other computers, for example by means of wide-area network 52 (which may be or comprise the network known as the Internet), local-area network 51, or physical delivery on media such as removable magnetic disk 29 or optical disk 31. Remote computer 49 typically is not in the control or dominion of the owner/provider of such information, and thus a secure repository,  
30 such as black box 240, may be used to control the use of information on a computer 49 over which the owner/provider of the information otherwise has no control.

- 7 -

Black box 240 is an exemplary embodiment of a secure repository. Black box 240 comprises code which executes on the general processing means provided by remote computer 49 (which may be analogous to the processing means of computer 20 depicted in FIG. 1, including, but not limited to, processing unit 21).

5 Black box 240 is preferably equipped with one or more cryptographic keys 248, and contains code to use the cryptographic keys to perform on or more cryptographic functions, such as decryption of encrypted information 204, and/or authentication of cryptographically signed data 212. In an exemplary embodiment, cryptographic keys 248 include asymmetric key pairs for use with public/private-key cryptographic

10 methods. Preferably, black box 240 also contains code that is designed to ensure that black box 240 performs its cryptographic functions only for trusted software, and in an environment that resists divulgence of sensitive results or attacks on black box 240 itself. It will be appreciated by those of skill in the art that decryption and authentication are merely exemplary, and not limiting, of the type of secure or

15 sensitive functions that could be performed by a secure repository, such as black box 240. Secure repositories that perform other functions may be installed or employed on remote computer 49, without departing from the spirit and scope of the invention.

Application program 244 is a software program that also executes on remote computer 49. FIG. 2 depicts an example where black box 240 performs

20 decryption and/or authentication services for application program 244, which is a program that in some way processes or uses information. In this example, application program 244 provides encrypted information 204 to black box 240. Black box 240, in turn, decrypts encrypted information 204 (e.g., by using one or more cryptographic key(s) 248) and returns decrypted information 208 to application program 244.

25 Similarly, application program 244 may call upon black box 240 to authenticate an item of data 212, and black box 240 may determine the authenticity of data 212 (e.g., by using one or more cryptographic key(s) 248) and may optionally provide to application program 244 an indication 216 of whether data 212 is authentic. Encrypted information 204 may, for example, be textual information (e.g., a novel), digital audio

30 (e.g. music), digital video (e.g., a movie), financial data, software, or any other type of information (confidential or otherwise). Data 212 to be authenticated may comprise



- 8 -

a signed certificate or other signed information. In a typical use, data 212 includes a certificate attesting to black box 240 that application program 244 is a sufficiently trustworthy program (i.e., that application program 244 can be trusted to handle decrypted information 208 without violating rules applying to the usage of decrypted information 208). Data 212 could also comprise a signed "license" to use decrypted information 208, and black box 240 (or trusted decoupling interface 220 (see below)) may authenticate the signed license to verify that the license is not actually a forgery that has been proffered by a user to gain unauthorized access to decrypted information 208. Depending on the nature of encrypted information 204, application program 244 may be a text-rendering program (i.e., a program that enables the reading of electronically-distributed encrypted text by displaying the text as print characters on a monitor), a video-rendering program, an audio-rendering program, or any other type of program that is capable of processing decrypted information 208 in some manner (or that, for any reason, needs to authenticate cryptographically signed information). It will be appreciated that the arrangement shown in FIG. 2 is particularly advantageous, because it allows application program 244 to make use of cryptographic services (i.e., decryption and authentication), without having to know the details of how those cryptographic services are performed, or the particular key(s) 248 that are used to perform them.

Associated with remote computer 49 are one or more hardware IDs 224. A hardware ID 224 comprises data that identifies, or otherwise relates to, particular hardware associated with remote computer 49. Preferably, hardware ID 224 comprises one or more of the following component IDs: CPUID 228, processor serial number 232, and hard disk ID 236. CPUID 228 may be a unique (or substantially unique) pseudo-random number assigned to a central processing unit (CPU) (not shown) of remote computer 49 by the manufacture of such CPU. CPUID 228 may be retrievable by executing an instruction on such CPU that provides the CPUID to a running program. Processor serial number 232 may be a sequentially-assigned number associated with a CPU (or other processor) of remote computer 49, which may also be retrievable by executing an instruction on such CPU. Some processors may have both a CPUID 228 and a serial number 232; other processors may have only one but not the

- 9 -

other. Hard disk ID 236 may be a number assigned (e.g., pseudo-randomly or sequentially) to a hard disk (not shown) associated with remote computer 49. Hard disk ID 236 may be assigned by the manufacturer or distributor of such hard disk, and may be stored in an accessible location on such hard disk. A function of hardware ID 224 is to provide a number that is uniquely (or substantially uniquely) associated with remote computer 49. Preferably, hardware ID 224 is based on all of the following component IDs: CPUID 228, serial number 232, and hard disk ID 236. However, hardware ID 224 may be based on a subset of those component IDs, or on entirely different component IDs (which may related to different hardware of remote computer 49, or, perhaps, to the serial number of an operating system (not shown) installed on remote computer 49). Moreover, hardware ID 224 may comprise the entirety of the component IDs on which it is based (e.g., a concatenation of the component IDs), a subset of those component IDs (e.g., the rightmost sixteen bits from each component ID), or may be derived by a function that generates hardware ID 224 based on the component IDs (e.g., the multiplicative product of a plurality of component IDs modulo  $2^{32}$ ). It is also possible to use one of the component IDs itself as a hardware ID 224; for example, hardware ID 224 could simply be equal to CPUID 228. Preferably, hardware ID 224 cannot easily be retrieved or derived in an environment other than remote computer 49, such that a program can be constructed that relies on retrieving hardware ID 224 from the execution environment during execution, thus binding the program to remote computer 49. Hardware ID(s) 224 may be created/determined in any manner that in some way relates to the hardware and/or environment of remote computer 49, without departing from the spirit and scope of the invention. Additionally, remote computer 49 may have a plurality of hardware IDs 224, with each hardware ID 224 being based on different component IDs, or on a different function of the same component IDs.

Black box 240 may access hardware ID(s) 224, and may use hardware ID(s) 224 in order to perform computations relating to its functions. For example, in the course of performing a function such as decryption, black box 240 may need to use a number  $n$ . Instead of storing  $n$  directly, black box 240 may store  $n$  minus the value of a hardware ID 224. In this case, in order to use the number  $n$ , black box 240 would

- 10 -

contain code that that retrieves or computes hardware ID 224 and adds it to the stored value to create  $n$ . This has the effect of "binding" black box 240 to a particular remote computer 49, since it needs a value that can only be accessed on remote computer 49. As a simple example, if hardware ID 224 is equal to CPUID 228, black box 240 could  
5 execute an instruction to retrieve CPUID 228 and add it to the stored value. In this case, the instruction will not produce the correct value  $n$  unless black box 240 is executing on the remote computer 49 that has the correct CPUID 228.

It should be noted that black box 240 is designed not merely to perform sensitive functions in a hidden and protected manner, but also to ensure that the  
10 sensitive results that it produces cannot escape to an untrusted software module – which means in practice that black box 240 performs the function of authenticating other software modules that are part of the secure environment in which black box 240 operates. Preferably, black box 240 authenticates every such software module. In general, each such software module maintains, in a known place, a signed hash of its  
15 code, where the signature is generated with a private key. Cryptographic keys 248 may include the corresponding public key which may be used by black box 240 to verify such signatures and establish trust with those software modules.

Black box 240 may communicate directly with application program 244, or, alternatively, it may communicate through decoupling interface 220. Decoupling  
20 interface 220 is essentially an intermediary by way of which black box 240 and application program 244 may communicate using a common language or protocol. The use of decoupling interface 220 may improve the versatility of a distribution system that uses black box 240, since it allows application program 244 to communicate with different types of black boxes 240, and allows black box 240 to communicate with  
25 different types of application programs 244. Additionally, decoupling interface 220 may authenticate application program 244 to black box 240. As discussed in further detail below, one issue that arises in the use of a black box 240 that is separate from the application program 244 for which it provides services is that black box 240 should not perform sensitive functions for application program 244 unless it is certain that  
30 application program 244 can be trusted to use the sensitive functions only in approved ways, which means in practice that application program 244 must satisfy the protocol

- 11 -

and other requirements necessary to proffer a trust certificate to black box 240. When decoupling interface 220 is used, application program 244 need only be able to communicate with decoupling interface 220, where various embodiments of decoupling interface can be provided each of which can authenticate application program 244 (and  
5 decoupling interface 220 itself) to a particular black box 240. Decoupling interface 220 may, for example, be provided by the supplier of black box 240 to facilitate communication between black box 240 and application program 244, as well as to facilitate the “renewability” of black box 240 (as more particularly discussed below). It will be appreciated by those skill in the art, however, that while decoupling interface  
10 220 is a convenient and advantageous structure, communications between black box 240 and application program 244 may take place either directly or through decoupling interface 220 without departing from aspects of the invention.

#### Provision/Acquisition of Black Box 240

Remote computer 49 acquires black box 240 from a black box server  
15 304. Black box server 304 may be implemented on a typical computing device, such as computer 20 (shown in FIG. 1). (Hereinafter, black box server 304 and the computer 20 on which it is implemented shall be referred to interchangeably, unless context indicates otherwise.) Referring now to FIG. 3, black box server 304 is preferably connected to remote computer 49 via wide-area network 52. Preferably, wide-area  
20 network 52 is or comprises the network known as the Internet. Black box server 304 includes a black box generator 37a. Black box generator 37a is preferably a computer program running on computer 20 (such as one of the “other programs” 37 depicted in FIG. 1). Black box generator 37a receives a hardware ID 224 of a remote computer 49, and generates a black box 240 that is “individualized” for remote computer 49.  
25 “Individualized” in this context means that the contents of a given black box 240 is at least partly based on the hardware ID associated with the remote computer 49 for which the black box is created, and that a first black box 240 created for a first remote computer 49 is different from (or, at least, is very likely to be different from) a second black box 240 for a second remote computer 49. The various black boxes 240 are  
30 “different” in the sense that they contain different cryptographic keys 248, different code to apply the keys 248, and different obfuscating code. The particular ways in

- 12 -

which the black boxes 240 can be made different are discussed more particularly below in connection with FIG. 4.

In a typical use of black box server 304, remote computer 49 contacts black box server 304 and transmits to black box server 304 a request for a black box 240 via wide-area network 52. The request for a black box may be generated as part of a registration or "activation" process relating to a particular type of software. For example, application program 244 may be delivered without a black box 240 that is needed for its use or some aspect thereof. For example, application program 244 may be able to work with unencrypted information (or even "keyless sealed" information) in the absence of black box 240, but may need black box 240 in order to be "activated" for use with encrypted information 204. In this case, application program 244 may include activation software 308 which is invoked upon the first use of application program 244, where activation software 308 contacts black box server 304 in order to obtain a black box 240 which will enable use of application program 244. As another example, software 308 may be general-purpose web-browsing software by means of which a user of remote computer 49 may issue a request to black box server 304 by navigating to a site operated by black box server 304.

The request for a black box 240 that comes from remote computer 49 preferably includes the hardware ID(s) 224 of remote computer 49. Upon receiving the request, black box server 304 invokes black box generator 37a and proceeds to create a black box 240 that is at least partly based on hardware ID(s) 224, and is thus "individualized" for remote computer 49. As discussed above, the black box 240 that is created will have one or more cryptographic keys 248 associated therewith and hidden therein. Once black box 240 has been created, black box server 304 transmits black box 240 via wide-area network 52 to remote computer 49 for installation thereon.

It is particular advantageous to transmit hardware ID 224 and black box 240 via a network such as wide-area network 52 or (if applicable) local area network 51, because this allows a black box 240 to be obtained in "real-time" (e.g., the entire transaction of requesting, creating, and receiving a black box may take only seconds). However, it will be appreciated that the use of a network is merely exemplary and not limited of the means by which information can be exchanged between remote computer

- 13 -

49 and black box server 304. For example, if hardware ID 224 is deemed to be sensitive information such that there is no network 51 or 52 over which it can be transmitted with sufficient privacy, then remote computer 49 may store its hardware ID(s) 224 on magnetic disk 29 or optical disk 31 for physical delivery to black box server 304, or hardware ID 224 may be printed on paper and then physically delivered to the operator of black box server 304 for entry using keyboard 40. Moreover, the black box 240 created by black box server 304 may be delivered on magnetic disk 29 or optical disk 31 for installation on remote computer 49; or it may be delivered via wide-area network 51 even if hardware ID 224 was not delivered to black box server 304 in that way.

#### Black Box Generator 37a

FIG. 4 shows the detail of black box generator 37a. Black box generator 37a comprises a random number generator 404, a code database 408, a key generator / key database 448, a code generator 440, a compiler 436, and a postprocessor 452. Random number generator 404 generates random (or pseudo-random) numbers 432, which are used in the black box generation process as described below. Systems and methods for generating pseudo-random numbers are known in the art and are therefore not described herein. Code database 408 contains "diversionary code" for use by diversionary code generator 416. Code database 408 may also contain templates for use by cryptographic code generator 412. The nature and use of the code contained in code database 408 is further described below. Key generator / key database 448 is an object that either generates cryptographic keys 248 to be installed in black box 240, or that stores previously-generated cryptographic keys 248 for installation in black box 240. Methods of generating cryptographic keys are known in the art, and thus are not provided herein. Moreover, methods of implementing a database such as code database 408 or key database 448 are also known in the art and thus are not provided herein. Code generator, as more particularly discussed below, generates the code that implements the individualized black box 240. Code generator 440 may produce code in a source language (such as C or C++), in which case compiler 436 is used to compile the code. If code generator 440 generates executable code directly, then compiler 436 may be omitted. Postprocessor 452, as more particularly discussed below, perfects

- 14 -

certain code-obfuscation techniques that are specified at the source level by code generator 440 but cannot actually be performed and/or implemented until executable code has been produced.

Code generator 440 generates the code for black box 240. Recalling that  
5 black box 240 is "individualized" for remote computer 49, code generator 440 accepts as input the hardware ID 224, which is received by black box server 304 at the time that remote computer 49 requests an individualized black box 240. Code generator 440 generates the code for the black box based on hardware ID 224 – that is, the code that code generator 440 includes in black box 240 is at least partly determined by hardware  
10 ID 224, such that, all other things being equal, different hardware IDs 224 cause code generator 440 to produce different code. Code generator 440 may also accept as input one or more random numbers 432, and the code produced by code generator 440 may be based on random number(s) 432 in the sense that different random number(s) cause code generator 440 to produce different code. Preferably, code generator 440 accepts  
15 as input both hardware ID(s) 224 and random number(s) 432, and produces code based both on such hardware ID(s) 224 and such random number(s) 432. However, it will be appreciated that code generator 440 could use one but not the other, or could use some other value as input. It is particularly advantageous, however, for code generator 440 to use both hardware ID(s) 224 and random number(s) 432, because this has the effect  
20 of: (a) producing a black box 240 with code that is at least partly "random"; and (b) allowing code to be included in black box 240 that binds black box 240 to hardware ID(s) 224. The "randomness" aspect of the code in black box 240 is advantageous because it helps to ensure that if a first black box 240 is successfully "hacked," the techniques used to "hack" the first black box 240 are not easily reused on a second  
25 black box 240 that has been generated with a different random number 432. The aspect of the code for black box 240 being bound to hardware ID(s) 224 is advantageous because it tends to make black box 240 resistant to portability.

Code generator 440 contains various components that handle different aspects of the code generation process. The components preferably include: a  
30 cryptographic code generator 412, a diversionary code generator 416, a healing code generator 420, an obfuscating code generator 424, and a code reorganizer 428. In FIG.

- 15 -

4, these components are depicted as separate components of code generator 440. However, it will be appreciated by those skilled in the art that the implementation depicted is merely exemplary, as a single component could perform one or more of the various functions described. Moreover, code generator 440 could contain a subset of the components depicted, or additional components. Additionally, while separate code generation and/or code manipulation components are depicted, it should be appreciated that the code generated by these components need not be or remain separate; the generated code can be interleaved or combined in any manner. For example, cryptographic code generator 412 may create a first set of code, and diversionary code generator 416 may create a second set of code, where the first and second sets do not necessarily remain as separate contiguous blocks but may be woven together in any manner. These variations and other may be effected without departing from the spirit and scope of the invention.

The exemplary elements depicted in FIG. 4 are each discussed below.

15 Cryptographic Code Generator 412

Cryptographic code generator 412 generates the code for inclusion in black box 240 that applies cryptographic keys 248. It will be recalled that a function of the exemplary black box 240 generated by code generator 440 is to use cryptographic key(s) 248 to perform decryption and authentication services while hiding key(s) 248 from the operator of remote computer 49 on which black box 240 is to be installed. Code generator 440 obtains these cryptographic keys from key generator / key database 448 and creates code that hides them in black box. The keys 248 obtained from key generator / key database 448 may be any type or size of cryptographic keys for use with any type of cryptographic algorithm. Preferably, cryptographic keys 248 include asymmetric or "public/private" key pairs for use with public/private key encryption/decryption and authentication algorithms. One non-limiting and preferred example of a public/private key algorithm for encryption/decryption and/or authentication is the RSA algorithm described in U.S. Patent No. 4,405,829 (Rivest, et al.), although it will be appreciated by those skilled in the art that other public/private key algorithms could be used. Keys 248 are "hidden" in black box 240 in the sense that they are never actually represented in numerical form in black box 240. Instead,



- 16 -

black box 240 contains code that performs actions that are functionally equivalent to those that would be performed by the relevant cryptographic algorithm if keys 248 were available. Regarding key "hiding" techniques, see generally *A. Shamir & N. van Someren, "Playing Hide and Seek with Keys."*

5           An understanding of how cryptographic code generator 412 can generate code that applies keys 248 where the code does not have access to keys 248 begins with the mathematical principle that some computations involving a given number can be performed without directly using the number, but merely by using certain properties of the number. For example, one can determine whether a decimal integer is even or odd  
10 without knowing its entire value, but merely by knowing its least significant digit: a decimal integer is even if and only if its least significant digit is 0, 2, 4, 6, or 8 (or, in the case of a binary integer, if and only if its least significant bit is 0). One can determine whether a number is negative or non-negative without examining the entire number, but merely by examining the sign of the number. The number's least  
15 significant digit (or bit) and its sign are "properties" of a number, which, in the foregoing examples, may be all that one needs to know about the number in order to compute the information desired. Thus, a program might receive a secret number as input, where the program performs a first action if the number is negative and a second action if the number is non-negative. In this example, the program could be constructed  
20 to store only the sign bit of the number. Alternatively, instead of storing any information about the number, the program could read the sign bit and then represent it in memory by dynamically creating code that (non-obviously) produces either 0 or 1 depending on what the sign bit is, where any portion of the program that needs to know the sign bit simply executes the dynamically-created code instead of retrieving the sign  
25 bit from memory. This is a simple example of how a program can be constructed to use a number without storing the number in memory or otherwise exposing the number to discovery by a user.

Analogously, cryptographic code generator 412 of the present invention makes use of mathematical properties of a particular cryptographic key 248, and  
30 creates code that computes the decrypted message that results from applying key 248 to a given ciphertext message without actually using key 248 itself or otherwise requiring

- 17 -

access to key 248. Cryptographic code generator 412 similarly creates code that uses key 248 to validate cryptographic signatures – again, without requiring access to key 248. An explanation of the mathematics used in generating code to apply key 248 without accessing a copy of key 248 is provided in Appendix A below. In addition to  
5 the technique discussed in Appendix A, an additional key-hiding technique that may be used is to embed a random number in the code for black box 240, and to use that random number together with hardware ID 224 as a seed for a pseudo-random number generator; as the pseudo-random number generator produces numbers, bytes of a particular key may be plucked out of the number stream as needed by the  
10 cryptographic calculation.

FIG. 5 shows the detail of an exemplary cryptographic code generator 412. Cryptographic code generator includes a key analysis module 504 and a code producing module 508 that produces the actual code that applies key 248. The key 248 obtained from key generator / key database 448 is received by cryptographic code  
15 generator 412 for analysis by key analysis module 504. Key analysis module 504 performs an analysis of key 248 in light of both its numerical properties and the cryptographic algorithm that will be used to apply it to a ciphertext message (or, in the case of authentication, to apply it to a digital signature). Key analysis module may identifies one or more properties or “attributes” of key 248 and produces or identifies  
20 one or more actions and/or functions 512 that would be performed by a cryptographic algorithm in the course of applying a key 248 having the identified attributes. For example, suppose that key analysis module 504 determines that the forty-second bit in the binary representation of key 248 is a one (i.e., “on”), and key analysis module 504 is programmed with information that a particular action/function 512 is always  
25 performed by a particular cryptographic algorithm whenever that algorithm applies a key whose forty-second bit is a one. This action/function 512 can be identified by key analysis module 504 and provided as input to code producing module 508. Key analysis module 508 may identify any number of attributes of key 248, and may provide all of the actions/functions 512 corresponding to these attributes to code producing module  
30 508.

- 18 -

Code producing module 508 receives the actions/functions 512 that it will need to perform in order to apply key 248. It will be appreciated by those skilled in the art that it is possible to write a large – possibly infinite – variety of code that performs a given action or function 512. It will moreover be appreciated by those skilled in the art that, given a particular action or function 512, it is possible to generate different code to perform that function where the particular code generated is based on factors other than the action or function itself. In the example shown in FIG. 5, these “other factors” include hardware ID 224 and/or random number 432. Specifically, for each action or function 512, code producing module 508 produces code that performs such action or function 512, where the particular code produced is based on hardware ID 224 and/or random number 432.

Code producing module 508 can produce code based on hardware ID 224 and/or random number 432 in two ways. First, code database 408 may contain a plurality of alternative programs that perform the same action or function 512. For example, code database 408 may contain five different programs that perform a certain aspect of exponentiation, with each of the programs having been written by a human being and stored in code database 408. In this case, code producing module 508 may select a particular one of these five different programs based on the value of hardware ID 224 and/or random number 432. Second, code producing module 508 may generate code to perform an action or function 512 without the need to resort to a selection of human written code, but rather may write the code itself on an as-needed basis. Techniques for automated generation of code by machine are known in the art and therefore are not provided herein. Code producing module 508 may use automatic code generation techniques to generate code, where the particular code produced is based on hardware ID 224 and/or random number 432.

In addition to using hardware ID 224 as a parameter that merely determines the particular code used to perform certain actions or functions 512, code producing module may also produce code that specifically binds the produced code to a remote computer 49 having hardware ID 224. For example, certain critical values stored in the code could be increased by the value of hardware ID 224, and code could be created to retrieve hardware ID 224 directly from the hardware of remote computer

- 19 -

49 and subtract the retrieved hardware ID 224 from those stored values, where the critical value itself is the result of the subtraction. In this event, the code produced will not work (or will behave unexpectedly) unless the machine on which it is running provides access to the correct hardware ID 224. There are a multitude of possibilities  
5 as to how code may be built to rely on hardware ID 224 or otherwise to bind the code to hardware ID 224, and any of those possibilities may be used without departing from the spirit and scope of the invention.

#### Diversiónary Code Generator 416

Returning now to FIG. 4, code generator 440 includes a diversionary  
10 code generator 416. Diversiónary code generator 416 adds to black box 240 "diversionary code." The code produced by diversionary code generator 416 is "diversionary" in the sense that it performs computations - possibly an enormous quantity of computations - which preferably achieve no result of any importance to black box 240's function(s) of applying cryptographic keys 248 to encrypted  
15 information 204 or authenticatable data 212. While execution of the code produced by diversionary code generator 416 may produce no meaningful result, the code itself serves a purpose: it serves to confound attempts by hackers to analyze the code of black box 240. If one (e.g., a "hacker") begins with no knowledge of how black box 240 will perform its function, it is difficult or impossible to look at a particular piece of  
20 the code of black box 240 and determine whether it is the part of the code that performs sensitive functions, or merely the red herring produced by diversionary code generator 412. The code produced by diversionary code generator 412 may appear to be producing intermediate results used in the course of performing sensitive functions, where those results may never actually be used in the cryptographic process. Thus, one  
25 who is attempting to analyze the code of black box 240 will have to wade through (possibly) enormous amounts of code before finding the crucial part of the code that performs sensitive functions.

The code added to black box 240 by diversionary code generator 416 may be stored in code database 408. For example, code database may contain a  
30 gigabyte of computationally-intensive code, and diversionary code generator 416 may retrieve, for example, ten megabytes of this code for inclusion in black box 240. The

- 20 -

particular ten megabytes of code retrieved may be based, for example, on hardware ID 224 and/or random number 432. The code stored in code database 408 for use by diversionary code generator 416 may, for example, be obtained from computationally-intensive programs that were written for purposes otherwise unrelated to black box 240. Although any code may be used, the code is preferably computationally-intensive, and preferably chosen to appear to a hacker as if it is doing something of importance to the cryptographic functions that black box 240 implements. For example, inasmuch as public/private key cryptographic algorithms rely heavily on exponentiation, code could be chosen from programs that perform mathematical computations involving large amounts of exponentiation. Alternatively, instead of using code stored in code database 408, diversionary code generator 416 could programmatically generate the diversionary code.

The code produced by diversionary code generator 412 may be included in black box 240 in a variety of ways. For example, the code that actually implements the cryptographic process could be scattered throughout the diversionary code. As another variation, instead of making the code that implements the cryptographic process completely non-dependent on the diversionary code, diversionary code could comprise computationally intensive code that always (but non-obviously) produces the number one and stores it in a register (although the code may appear to one unfamiliar with it as if it might be capable of producing some other value). The register could then be multiplied by a value used during the cryptographic process. This may have the beneficial effects of (a) making it appear to a hacker as if the diversionary code is actually performing some computation of use in the cryptographic process, thereby making it difficult to distinguish the diversionary code from the "real" code, and (b) causing the black box 240 to produce unexpected results if the diversionary code is modified (e.g. if it is modified in such a way that it no longer produces the number one).

#### Healing Code Generator 420

Code generator 440 includes a healing code generator 420. Healing code generator 420 creates code that uses "error-correction" principles to detect "patches" (i.e., additions or modifications to the code of black box 240 that were not part of that

- 21 -

code as originally constituted). Healing code generator 420 may also create code that can be used to "repair" those patches by dynamically "re-modifying" the code of black box 240 to restore it to its original configuration (or to some intermediate state that exists (or should exist) during the execution of the black box 240 program).

5           In a conventional context, error-correction principles are employed to correct errors brought about by noise during data transmission. Application of these error correction principles to executable code is based on the notion that executable code, like any other data, is merely a sequence of bits. Therefore, minor modifications to the code (or any other data) can be detected and corrected -- whether the  
10       modifications are brought about innocently by random noise, or nefariously by deliberate attacks on the code (or data). In order to create healing code, healing code generator 420 may incorporate one or more error correcting codes into the code for black box 240 as part of the individualization process. These error correcting codes (ECCs) may include linear codes (such as Hamming codes), cyclic codes (such as BCH  
15       codes) and Reed Muller codes. The code generated by healing code generator 420 can detect tampering based on parity checks and on the error syndrome that is calculated based on code and data in black box 240. Based on the calculated error syndrome, the code generated by healing code generator 420 may either correct the modification to the code, or effectively destroy the executing code by effecting other changes to the  
20       code that will cause incorrect operation of the black box 240 program. The code generated by healing code generator 420 may be designed to detect/heal tampering both as to the initial state of the black box 240 program (i.e., the state of the black box 240 program immediately after loading it into memory) and/or the running state of the program (i.e., the state of the execution space of the program at some intermediate  
25       point during its execution). Additionally, as part of the individualization process, healing code generator 420 may create error correcting code based on a random number and/or hardware ID 224.

#### Obfuscating Code Generator 424

Code generator 440 includes an obfuscating code generator 424.  
30       Obfuscating code generator 424 creates code that complicates examination and modification of the functions performed by black box 240. It will be appreciated that

- 22 -

code produced by other components of code generator 440 is already considerably resistant to analysis. As noted above, the code that applies key 248 resists discovery of the key due to the fact that the key is never stored. Additionally, the "diversionary" code makes it difficult to discover which code is performing the sensitive cryptographic functions. Obfuscating code generator 424 adds an additional layer of protection to black box 240.

Obfuscating code generator 424 complicates analysis and/or modification of black box 240 by such techniques as: encrypting portions of the code of black box 240; introducing "integrity checks" into black box 240; and/or obfuscating execution of code segments (e.g., by loading the code into scattered, randomly-located portions of memory for execution). Obfuscating code generator 424 selects portions of the code of black box 240 to be protected by encryption, integrity checks, and obfuscated execution. Preferably, the encrypted portions include the portions of the code that perform the most security-sensitive cryptographic functions performed by black box 240, but also include other portions in order to complicate the identification of the truly crucial portions of the code. (I.e., if only the most sensitive portions were encrypted, then it would be easy for a hacker to locate the most sensitive portions simply by looking for encrypted code; therefore, these portions can be hidden by encrypting unimportant portions of the code including, but not limited to, the diversionary code produced by diversionary code generator 416.) Additionally, obfuscating code generator 424 selects segments of the black box code to be preserved by integrity checks and obfuscated execution.

As noted above, the code generated by code generator 440 is preferably in a source language, such as C or C++. One method of introducing the above-mentioned security protection into code that is created in a source language is to use a software development tool such as a "secure coprocessor simulation" ("SCP") toolkit. The following is a brief description of an SCP system, which is an exemplary tool used by obfuscating code generator 424 to introduce protections into the code of black box 240.

SCP: Overview

The SCP system works at the source-code level by creating a number of classes called *SCPs*, each of which supports an API for security operations such as encryption, decryption, integrity verification, and obfuscated execution. Operations  
5 such as encryption and checksum computation handle relocatable words, so that the SCP system may work whether or not code is relocated (e.g., as with DLLs).

To protect an application with the SCP system, one must have access to the application source code. Although the entity with access to the source code may be a human program, this need not be the case. Specifically, an automatic code generator,  
10 such as the code generator 440 used to create black box 240, may also use the SCP system to insert protections into code. The entity (i.e., human programmer or code generator) performs the following steps to use the SCP system to protect code:

1. SCP macros and labels are inserted into the source file(s) to be protected. These macros and labels indicate code sections to be protected by such  
15 measures as encryption, integrity verification, and obfuscated execution.
2. Places in the source code are selected where functions such as obfuscated execution, integrity verification, decryption, and other SCP functions are to be performed. SCP macros and calls are inserted into the source code at these selected places to perform these actions.
- 20 3. SCP files are added to the application and/or compiled with the application source code to yield an application executable (e.g., a .EXE or .DLL file).
4. A post-processing tool (e.g., postprocessor 452) is run on the application executable file. The post-processing tool may locate sections of the executable file to encrypt, checksum, or otherwise process in the  
25 executable. The post-processing tool may also randomly (or pseudo-randomly) generate cryptographic keys, encrypt the code, scatter some keys throughout the executable file's code (i.e., ".text") section, and perform other setup actions.

The follow sections describe exemplary features of the SCP system.



- 24 -

SCP: Code-integrity verification

A programmer (or code generator 440) may checksum any number of (possibly overlapping) application segments at any time during execution. SCP provides two verification methods, which are accessed via a macro-based API that generates appropriate functions and data in the protected application's code (.text) section. The SCP system may, for example, assume that only one code section exists and that its name is ".text," as is typically the case with standard WIN32 executables. Verification checks can be inlined (as described below) to avoid exposing boolean return values and single points of attack.

Method 1: This method works with overlapping segments. The programmer or code generator marks a region to be verified by inserting the macros:

BEGIN\_VERIFIED\_SEGMENT(ID1, ID2)

and

END\_VERIFIED\_SEGMENT(ID1, ID2)

inside functions, and adding the macro

VERIFIED\_SEGMENT\_REF(ID1, ID2)

outside of functions. The former two macros mark regions, and the latter macro creates a "segment-reference function" that returns the region's addresses and checksum. The variable (ID1, ID2) is a unique two-byte segment identifier, and checksums are pre-computed in order of the ID1 values of regions; this is to allow cross-verification of code regions by ensuring a definite order of checksum computation and storage in the code section. To verify a region, the programmer or code generator inserts a function such as

SCP.VerifySeg(VerifiedSegmentRefID1\_ID2),

which returns a boolean value and makes available both the valid checksum and the computed checksum (which must match if the code segment is intact). Any SCP object can verify any segment.

Method 2: The second method works only with non-overlapping segments. The programmer or code generator specifies a section to be verified using the macros

BEGIN\_VERIFIED\_SECTION(Section, ID1, ID2)

- 25 -

and

END\_VERIFIED\_SECTION(Section, ID1, ID2)

which must be placed outside of functions. The variables (Section, ID1, ID2) specify a unique section name and a pair of identifier bytes. The programmer or generator can

5 verify a section by inserting into the source code a function call such as

SCP.VerifyAppSection(BeginSectionID1\_ID2, EndSectionID1\_ID2)

both to obtain a boolean verification value and two checksums, as above.

It should be noted that integrity checks can be performed in a way that is dependent on hardware ID 224, such that black box 240 will not operate properly if it  
10 is running on the wrong machine.

SCP: secret scattering

Method 2 above uses cryptographic keys scattered using a subset-sum technique. Each key corresponds to a short string used to compute indices into a pseudo-random array of bytes in the code section, retrieve the bytes specified by the  
15 indices, and combine these bytes into the actual key.

SCP: Obfuscated function execution

Using labels and macros, the programmer or code generator may subdivide a function into any number of blocks that the SCP system encrypts. When the function runs, SCP preferably uses multiple threads to decrypt each block into a  
20 random memory area while executing another block concurrently. Code run in this manner is preferably self-relocatable. For example, one way to implement self-relocatable code on the Intel x86 platform is to make all function calls via function pointers. Alternatively, code could be relocatable by means of dynamically adjusting call offsets during execution.

25 SCP: Code encryption and decryption

The programmer or code generator can specify any number of code regions to be encrypted by enclosing them within the macros

BEGIN\_ENCRYPTED\_SEGMENT(ID1, ID2)

and

30 END\_ENCRYPTED\_SEGMENT(ID1, ID2)

and adding the macro

- 26 -

### ENCRYPTED\_SEGMENT\_REF(ID1, ID2)

outside encrypted regions. These macros and the previously described macros for verified segments serve similar purposes. If a verified region with the identifier (ID1, ID2) exists, its checksum is used as the encryption key for the corresponding encrypted segment (i.e., the one with identifier (ID1, ID2)). The programmer or code generator calls for a segment to be decrypted prior to its execution (by inserting appropriate SCP call(s)), and can then re-encrypt the segment (also using appropriate SCP call(s)). Encrypted segments may overlap, and may preferably be encrypted based on their order of appearance in the source code.

#### 10 SCP: Probabilistic checking

Each SCP class has its own pseudo-random-number generator that can be used to perform security actions such as integrity verification only with certain probabilities. Additionally, SCP macros may be available that produce pseudo-random generators inline for this function, as well as for any other function that requires pseudo-random numbers.

#### 15 SCP: Boolean-check obfuscation

Several SCP macros provide means of hiding boolean verification of checksums. In particular, SCP macros or objects may use checksums to mangle stack and data, and compute jump addresses and indices into simple jump tables, so that boolean checks become implicit and non-obvious.

#### 20 SCP: Inlining

To avoid single points of attack, SCP provides macros for inline integrity checks and pseudo-random generators. These macros may essentially duplicate the code from the SCP segment-verification functions.

#### 25 Code Reorganizer 428

Code generator 440 may include a code reorganizer 428. Code reorganizer 428 reorders or resequences the code that has been produced by the other components of code generator 440. For example, code reorganizer 428 may move instructions in the code into a different sequential order, and then add jumps (such as "conditional" jumps that test for a condition that is always, but non-obviously, true.

- 27 -

Other Features of Code Generator 440

Code generator 440 has some features that cut across the exemplary components depicted in FIG. 4. For example, code generator 440 may create different code to perform the same function within black box 240. This technique has the effect of obfuscating the function being performed, since one must perform an analysis of the different code segments in order to determine whether they are functionally equivalent. Additionally, the differing code segments may be "inlined," so as to guard against single point attacks. Other techniques that may be used by code generator 440 including introducing timing loops (to detect "hijacking" of calls by an outside program), or "interleaving" data, state and code by hashing and jumping on the result (which may have the effect of making local modification more difficult). Code generator 440 may create code that implements techniques aimed at detecting observation of black box 240 by a kernel-level debugger; such techniques may include changing debug registers, checking to see if debug registers were changed, trapping randomly and examining stack addresses. Moreover, code generator 440 may create non-sensitive code for black box 240, such as routine code that performs the startup and shutdown of black box 240, or that performs routine "housekeeping" tasks (although this "routine" code may still be protected by techniques such as those introduced by obfuscating code generator 424).

Code generator 440 may also produce "diversionary data" for inclusion in black box 240. Like the "diversionary code" discussed above, diversionary data is not of direct relevance to the functions that black box 240 performs, but serves to complicate analysis of black box 240. Diversionary data may include data that is never used, data that is used exclusively by diversionary code, or as input to functions which are designed to nullify the effect of the diversionary data. An additional feature of the diversionary data is that, since its quantity can be varied, it has the effect of shifting the address of black box 240's useful code between different instances of black box 240, which makes it less likely that a successful attack on one black box will aid an attack on another black box.

- 28 -

Compiler 436

Compiler 436 is used when code generator 440 creates code in a source language instead of machine-executable code. For example, in an exemplary embodiment code generator 440 creates code in the C++ programming language, and  
5 compiler 436 is or comprises a C++ compiler. Compiler 436 receives the source code produced by code generator 440 and converts it into executable code. Compilers are well-known in the art and commercially available, and therefore are not discussed in detail herein. Compiler 436 may be a specially-configured compiler included in black box generator 37a, or it may be a general purpose compiler that is separate from black  
10 box generator 37a.

Postprocessor 452

The SCP tool discussed above provides a convenient means to specify certain code security techniques, such as inline code encryption, integrity verification (e.g., checksums), and obfuscated execution of code. However, certain aspects of these  
15 techniques cannot be performed directly on source code. For example, it is not possible to encrypt source code prior to compilation. Instead, portions of the executable code to be encrypted can be delimited in the source file, but the actual encryption must await creation of the executable code. Similarly, segments to be protected by integrity checks can be delimited in the source, but the actual creation of the integrity value must await  
20 creation of the executable, since it is not possible to obtain a hash of the executable code (by means of which integrity will be verified) until the source code has been compiled. Postprocessor 452 performs these functions.

Postprocessor 452 performs, for example, the functions of: encrypting the code specified for encryption, computing integrity value, generating cryptographic  
25 keys for the encryption of code, and storing integrity values and code decryption keys in the executable file.

Process of Creating an Individualized Black Box

Referring now to FIG. 6, an exemplary process is shown by way of which black box server 304 creates and provides a secure repository that is  
30 individualized for remote computer 49, such as exemplary black box 240. First, at step 601, black box server 304 receives the hardware ID 224 associated with remote

- 29 -

computer 49. As previously noted, hardware ID 224 may be received via a network, such as wide-area network 52, in the form of a request for a black box 240, where the request comes from remote computer 49. Alternatively, hardware ID 224 may be received by other means, such as physical delivery on a digital medium (e.g., magnetic disk 29, optical disk 31, etc.), or on paper for entry by keyboard. Black box server 304 may use the process depicted in FIG. 6 to create an individualized black box regardless of how it receives hardware ID 224. Moreover, it will be appreciated by those skilled in the art that hardware ID 224 is merely exemplary of the type of information that supports the creation of a black box 240 individualized for remote computer 49. For example, instead of hardware ID 224, black box server 304 may receive the serial number of an operating system running on remote computer 49. Any type of information will suffice, provided that it identifies remote computer 49, or is in some way related to remote computer 49 or to the environment present on remote computer 49 (e.g., the serial number of the operating system installed on remote computer 49).

At step 602, black box server creates the executable code for an individualized black box 240, where the code created is at least partly based on hardware ID 224. As discussed above, the process of creating this executable code may, for example, be performed by black box generator 37a. An exemplary process by which step 602 performed is shown in FIG. 7 and discussed in detail below.

At step 603, black box server 304 provides the individualized black box 240 to remote computer 49. Black box server 304 may provide black box 240 via, for example, wide-area network 52, which may be the same network over which black box server 304 received hardware ID 224. Alternatively, black box server 304 may provide black box 240 to remote computer 49 by physical delivery of magnetic disk 29 or optical disk 31, or by any other means by which the executable file(s) in which black box 240 is contained may be communicated from one computer to another.

FIG. 7 shows an exemplary process for performing the code creation step 602 depicted in FIG. 6. The process shown in FIG. 7 may, for example, be carried out by black box generator 37a (shown in FIG. 4). At step 701, black box generator 37a obtains a cryptographic key 248 (or possibly several cryptographic keys)

- 30 -

to be hidden in black box 240. Cryptographic key 248 may, for example, be obtained from key generator / key database 448.

At step 702, black box generator 37a analyzes the newly obtained cryptographic key 248 in order to identify one or more actions that would be performed in the course of using a given cryptographic algorithm to apply cryptographic key 248 to data. The cryptographic algorithm may either be a decryption algorithm (that uses cryptographic key 248 to convert ciphertext into cleartext), or an authentication algorithm (that uses cryptographic key 248 to verify that a particular data was created by the holder of a particular secret). The particular cryptographic algorithm by which cryptographic key 248 will be applied is taken into account when identifying actions will be performed in the course of applying cryptographic key 248. As one example, step 702 may include the act of using key analysis module 504 of cryptographic code generator 412 to identify actions/functions 512, as depicted in FIG. 5.

At step 703, black box generator 37a generates code to perform the actions identified at step 702. The code generated at step 703 is capable of applying cryptographic key 248 to data, preferably without requiring access to cryptographic key 248 itself. The process of generating the code may, for example, be carried out by code producing module 508 of cryptographic code generator 412, depicted in FIG. 5. Code may be generated programmatically, or it may be retrieved from code database 408. The particular code that is generated or retrieved may be at least partly based on hardware ID 224 and/or random number 432. Additionally, the code produced may be designed to rely in some way upon correct dynamic retrieval of hardware ID 224 from the environment in which black box 240 is intended to run.

At step 704, black box generator 37a generates "diversionary" code for inclusion in black box 240. Diversionary code may, for example, be stored in code database 408. A portion of the code stored in code database 408 may then be retrieved by diversionary code generator 416. The particular code retrieved for inclusion in black box 240 may be based, for example, on random number 432 and/or hardware ID 224.

- 31 -

At step 705, black box generator 37a generates "healing" code, which is designed to replace "patches" (i.e., unauthorized modifications to the code of black box 240) with the originally intended code. The healing code may be generated, for example, by healing code generator 420. The particular code generated at step 705 is generally based on other parts of the code to be included in black box 240, since the code generated at step 705 is specifically designed to replace portions of black box 240 if they become modified.

At step 706, black box generator 37a introduces obfuscation and integrity measures into the code of black box 240, such as encrypted code, integrity checks, and obfuscated execution of code. Black box generator 37a may, for example, perform this function by using obfuscating code generator 424 to introduce the macros and function calls of the SCP system described above.

At step 707, black box generator 37a reorganizes the code for black box 240, for example by reordering or resequencing segments of the code and introducing jumps to cause the code to be executed in the correct sequence. The reorganization may, for example, be performed by code reorganizer 428.

At step 708, the code generated by black box generator 37a at steps 703 through 707 is optionally compiled. Preferably, the code generated at steps 703 through 707 would be generated in a source language such as C or C++, in which case it needs to be compiled. The compilation may, for example, be performed by compiler 436, which is a compiler appropriate for the language in which the code was generated, such as a C compiler or a C++ compiler. In an alternative embodiment in which black box generator 37a generates code directly in a machine executable format (e.g., a .EXE or .DLL file), then it is unnecessary to perform step 708.

At step 709, black box generator 37a performs postprocessing of the executable code for black box 240. Step 709 may, for example, be performed by postprocessor 452. As previously discussed in connection with FIG. 4, there are some aspects of the code obfuscation and integrity measures introduced at step 706 that cannot be perfected at the source level, but must wait until the executable code has been produced. For example, one obfuscation measure is to encrypt executable code inline, and decrypt it prior to its use. It is not possible to encrypt the executable code



- 32 -

until it has been produced (and it is not possible to encrypt the source code because it will not compile). Moreover, it is not possible to create the integrity values (e.g., hashes, checksums, etc.) that are used to implement integrity checks because these measures, too, require encrypted code. Thus, at step 709 postprocessing of the executable code produced at step 708 is performed. Specifically, sections of code delimited for encryption are encrypted, and sections of code marked for integrity checks are hashed. The keys used for encrypted code may be selected at step 709. The keys may be based in part on hardware ID 224 and/or random number 432. The particular method of performing an integrity check may be based in part on hardware ID 432.

Once postprocessing is complete, black box generator 37a proceeds to step 603 shown in FIG. 6. It will be appreciated that, while the steps of FIG. 7 are depicted as taking place in a certain order, certain of the steps shown may take place in a different order. For example, the code generated at step 703 through 705 could be generated in sequences other than that depicted in FIG. 7. The reorganization of code at step 707 could take place either before or after the compilation performed at step 708. Other modification may be made to the order of steps without departing from the spirit and scope of the invention.

#### Example Architecture Incorporating Black Box 240 and Decoupling Interface 220

As noted above in connection with FIG. 2, black box 240 and application program 244 may communicate either directly or through a decoupling interface 220. FIG. 8 shows an exemplary architecture of a system in which a decoupling interface is employed for use with black box 240 and application program 244.

Decoupling interface 220 addresses the issue of how black box 240 can be replaced with another black box 240a, while still permitting communication and authentication to take place between application program 244 on the one hand, and black box 240 or 240a on the other hand. Replacement of black box 240 may become necessary if black box 240 has become damaged (e.g., if a hacker has made a deliberate and irreparable attempt to modify black box 240), if black box 240 has become obsolete due to the development of new secure repository technology (which

- 33 -

may, for example, include either new software techniques or a hardware-based repository), or if application program 244 may be usable on different types of platforms having different types of secure repositories. (E.g., an open platform such as a PC running one of the MICROSOFT WINDOWS 95, 98, NT, or 2000 operating systems may require a different type/level of security from a closed platform such as a dedicated text viewing device.) However, it will be noted that application 244 preferably should be able to interface with any black box – either original or replacement – in at least two ways. First, application program 244 needs to authenticate itself to the black box, since the black box should not perform sensitive function for an entity who has not established trustworthiness. Second, application program 244 needs to be able to communicate with black box 244 in order to request performance of the sensitive functions that black box 244 is designed to perform. Decoupling interface 220 provides a single authentication and communication protocol that application program 244 can use regardless of the black box being used, thus making the particular secure repository transparent to the developer of application program 244.

Application program 244 and black box 240 in the exemplary architecture shown communicate through decoupling interface 220. Decoupling interface 220 may, for example, be or comprise an application programmer interface (API) having an “initialization” call and a “bind to license” call. (Licenses are explained in further detail below; briefly, a license permits the use of content protected by black box 240.) The API may be provided in the form of a dynamic-link library (.DLL) file that is loaded with application program 244 and executes in the process (i.e., in the address space) of application program 244. Both of the calls provided by the API may be exposed to application program 244 and implemented by decoupling interface 220 (i.e., the calls have addresses located in the address space of application program 244). Additionally, these two calls may preferably provide all of the interaction that is necessary in order for application program 244 to make use of black box 240. For example, the “initialization” call may perform the functions of (a) authenticating application program 244 to black box 240, and (b) causing decoupling interface to execute instructions that initialize black box 240 for use and allow black

- 34 -

box 240 to authenticate decoupling interface 220. A purpose of black box 240's authenticating decoupling interface 220 is to establish a chain of trust. Since decoupling interface 220 presents application program 244's proffer of trustworthiness to black box 240, black box 240 must trust decoupling interface 220 to perform this authentication properly (e.g., black box 240 may need to ensure that decoupling interface 220 is not a rogue DLL presenting a stolen certificate in an unauthorized context). The bind-to-license call may request that black box 240 perform its sensitive function(s) for application program 244. In the example depicted in FIG. 8, the sensitive function that black box 240 performs is to enable the use of encrypted licensed data object 800 when permitted by license 816 (as more particularly described below), and the bind-to-license call may provide "enabling bits" 824 that allow application program 244 to use licensed data object 800.

For example, application program 244 may issue an initialization call (e.g., *DecouplingIF::Init()*) which is implemented by decoupling interface 220. This call may cause decoupling interface 220 to execute code that starts black box 240, gives it an opportunity to authenticate decoupling interface 220, and prepares a secure environment that discourages modification or observation of black box 240 while black box 240 is executing. It should be noted that the act of authenticating decoupling interface 220 is optional, as there may be environments in which the authenticity of decoupling interface 220 can be presumed from circumstance (such as purpose-built devices for which all software is in the form of "firmware" installed by the manufacturer). Black box 240 may take steps to prepare a secure environment, such as attaching to application program 244 using the *DebugActiveProcess* system call of the MICROSOFT WINDOWS 95/98/NT/2000 operating systems, in order to prevent other debuggers from attaching. Preferably, decoupling interface 220 does not process the bind-to-license call until decoupling interface 220 has completed the initialization process with black box 240. After the initialization process is complete, application program 244 may issue a bind-to-license call (e.g., *DecouplingIF::BindToLicense()*), which causes decoupling interface 220 to execute code that requests black box 240 to perform its sensitive functions for application program 244. In the example of FIG. 8, the bind-to-license call requests that black box provide enabling bits 824. Black box

- 35 -

240 then provides enabling bits 824 to decoupling interface 220 (assuming that circumstance permit), and decoupling interface 220 provides enabling bits 824 to application program 244. It will be appreciated that this arrangement allows application program 244 to authenticate itself and request services from black box 240 without  
5 knowing the details, mechanics, or communications protocols needed to interface directly with black box 240. In effect, decoupling interface provides a common language through which application program 244 and black box 240 may communicate.

Still referring to FIG. 8, in the exemplary architecture shown, data object 800 is a file that includes encrypted information 204, a sealed key 820, and a  
10 license 816. Sealed key 820 may be the key that decrypts encrypted information 204. In a preferred embodiment, sealed key 820 is a symmetric key that was used to encrypt encrypted information 204. License 816 governs the use of encrypted information 204. For example, license 816 may specify the rights to decrypt and/or render encrypted information 204. Encrypted information 204 may, for example, be the text of a book.  
15 In other examples, encrypted information 204 may be audio or video content, such as a digital audio file or a digital video file. Application program 244 is a software application appropriate for the nature of the information contained in data object 800. For example, if encrypted information 204 is the text of a book, then application program 244 may be a text-rendering application or "reader" program. If encrypted  
20 information 204 is a digital audio or digital video, then application program 244 may be an audio-rendering or video-rendering application.

The exemplary architecture of FIG. 8 includes black box 240. Black box 240 includes a public key 248a, and a corresponding private key 248b. As discussed above, private key 248b is preferably "hidden" in black box 240, such that no copy of  
25 private key 248b actually appears in black box 240, although black box 240 contains the code necessary to apply private key 248b.

In the example of FIG. 8, there is shown a certificate 812, which has associated therewith an asymmetric key pair including a public key 804 and a private key 808. Private key 808 is not represented directly in certificate 812, but rather is  
30 encrypted with the public key of black box 240. Sealed key 820 is "key"-sealed in data object 800 with public key 804 of certificate 812 such that it can only be "unsealed"

- 36 -

with private key 808. It will be appreciated that the series of encrypted and sealed keys establishes a chain of control from black box 240, to certificate 812, to sealed key 820, to encrypted information 204, such that it is not possible to decrypt information 204 without all of these objects. Specifically, information 204 can only be accessed with  
5 sealed key 820. Sealed key 820, in turn, can only be accessed with private key 808. Private key 808, in turn, can only be accessed with private key 248a of black box 240. This is a particularly advantageous arrangement, since it allows access to data object 800 to be tied to the private key 808 of certificate 812, rather than to a particular black box 240. For example, if black box 240 were replaced with black box 240a (which has  
10 public/private keys 248c and 248d, which are different from public/private keys 248a and 248b of black box 240), data object 800 could be used by black box 240a without any modification to data object 800, merely by issuing a new certificate 812a, which has the same public/private keys 804 and 808, but with private key 808 being encrypted with public key 248c of new black box 240a, instead of being encrypted with  
15 public key 248a of old black box 240.

Exemplary black box 240 depicted in FIG. 8 performs the function of enabling application program 244 to render encrypted content 204 by providing "enabling bits" 824 to application program 244. Enabling bits 824 may comprise either decrypted content 208 (in which case black box 240 performs the function of applying  
20 key 820 to convert encrypted content 204 into decrypted content 208), or key 820 itself (in which case application program 244 uses key 820 to perform the actual decryption of data object 800). Exemplary black box 240 also perform the function of validating license 816 to determine whether license 816 permits use of encrypted information 240. If license 816 does not permit the use of encrypted information 804 (e.g., if license 816  
25 is expired, or if data object 800 is not present in an authorized environment), then black box 240 does not provide enabling bits 824 to application program 244.

Black box 240 preferably provides enabling bits to application program 244 by way of decoupling interface 220. However, application program 244 and black box 240 may communicate directly in accordance with aspects of the invention.  
30 Decoupling interface is particularly useful in the example of FIG. 8, because it allows for simple replacement of black box 240 with black box 240a without requiring any

- 37 -

modification to rendering application 244. Thus, in accordance with aspects of the invention, black box 240 could be replaced with black box 240a (e.g., if black box 240 has become corrupted or obsolete) without any change to either rendering application 244 or data object 800 – merely by providing black box 240a with certificate 812a and, if necessary, replacing the code of decoupling interface 220 to allow it to communicate/authenticate with black box 240a. (E.g., If decoupling interface 220 is a dynamically linkable library of executable code (i.e., a DLL) which links to application program 244 at runtime, the DLL can be replaced with a new DLL. This replacement may transparent to application program 244, since the calls made by application program 244 (i.e., *Init()* and *BindToLicense()*) would simply reference the new code in the new DLL.) Thus, the use of decoupling interface 220 is particularly advantageous because it supports the replaceability of black box 240, and thus supports a “renewable” model of security. For example, if a hardware-based repository should become available, decoupling interface 220 may provide communication between the not-yet-created hardware based repository and application program 244 in a manner that is transparent to the developer of application program 244. As another example, technology for the creation of software-based repositories could progress, thereby allowing a first software-based repository to be replaced with a second software-based repository having features that were not present when the first repository was created. As a further exemplary feature, application program 244 may specify a “type” of repository that it is able and/or willing to work with, where new repositories of that “type” may not be in existence at the time that application program 244 is created, but which may be developed later, thereby further supporting the “renewable” model of security.

It is possible for black box 240 and application program 244 to communicate without decoupling interface 220. For example, application program 244 could authenticate itself directly to black box 240 and could receive directly from black box 240 the enabling bits 824 needed to use object 800. However, the use of decoupling interface 220 is particularly advantageous because it supports the “renewability” of black boxes, as described above. More particularly, since application program 244 only needs to be able to communicate through decoupling interface 220, it

- 38 -

does not need to know any of the details about black box 240, such as how to authenticate itself to black box 240, or what communication protocol is used to communicate with black box 240. Thus, black box 240 could be replaced with a different black box 240a, with the change being transparent to the developer of application program 244. For example, black box 240a may be a different type of black box for use on a different type of remote computer 49 (such as a dedicated rendering device), or, as noted above, it may be a black box 240a that incorporates new security technology that had not been developed at the time that black box 240 was provided to remote computer 49, or it may be a hardware-based secure repository. It will be appreciated that the use of decoupling interface 220 enables the use of black box 240a with application program 244 and data object 800 even if black box 240a has not been developed or is otherwise not yet in existence at the time that application program 244 and data object 800 are installed on remote computer 49.

Referring now to FIG. 9, there is shown an exemplary process by which application program 244 uses black box 240 through decoupling interface 220. At step 901, application program 244 starts execution. Application program 244 may be a program executing on an open platform or general purpose computer, such as remote computer 49. Alternatively, application program 244 may be a program for a closed platform such as purpose-built hardware (not shown). For example, application program 244 may be a program that displays electronically-distributed text to a (human) user. One version of application program 244 may be designed to run on general-purpose open-platform remote computer 49. Another version of application program 244 may be designed to run on a dedicated hand-held reading device that runs no software other than that necessary to display electronic text. Preferably, application program 244 loads decoupling interface 220 (e.g., by linking with a DLL that contains instructions which implement decoupling interface 220), in order to facilitate communication with black box 240.

At step 902, application program issues an instruction to initialize black box 240. Preferably, the initialization instruction is implemented by decoupling interface 220 and takes the form of a method that decoupling interface 220 exposes to application program 244. For example, the code of application program 244 may

- 39 -

include an instruction of the form (*DecouplingIF::Init()*), which executes code to perform the initialization function (where the code is in the decoupling interface DLL and linked to application program 244 at runtime). The call to *Init()* may, optionally, be parameterized by data representing a type of black box 240 (e.g., if the developer and/or provider of application program 244 has deemed that application program 244 should work only with a particular type of black box, decoupling interface 220 may provided support for such a condition if it is specified as a parameter). Steps 903 and 904, described below, may be performed as part of, or in response to, the initialization instruction issued at step 902.

At step 903, decoupling interface 220 proffers to black box 240 proof of the authenticity and/or trustworthiness of application program 244. The process of authenticating application program 244 generally includes obtaining a certificate from application program 244 (where the certificate is signed with the private key of a trusted certifying authority) and then validating the signature. The certificate may be located in the code of application program 244 in a place known to the DLL that implements decoupling interface 220. Black box 240 preferably incorporates the public key of the certifying authority. The process of authenticating application 244 may be particularly simple when the platform on which application 244 is running is a purpose built device or features hardware-enforced isolation, since it may then be presumed that only trusted applications 244 would be installed on such a device.

At step 904, black box 240 is started and, optionally, is given the opportunity to prepare a secure environment, including authenticating decoupling interface 220 and application program 244. The details of this process depend on the particular environment in which black box 240 and application program 244 are running. For example, in the case of a closed, purpose built device, the only actions that may need to take place are retrieving a private key from a ROM and validating its certificate.

In the case of "authenticated boot" environments, the process may include using an operating system call to check a software image (e.g., of the decoupling interface 220 DLL), checking the certificate of application program 244, and isolating the code and data space of application program 244.



- 40 -

In the case of an open platform, such as where application program 244 runs on a typical personal computer using one of the MICROSOFT WINDOWS 95/98/NT/2000 operating systems, the process of authenticating decoupling interface 220 and preparing a secure environment may include the following actions. The  
5 initialization call of decoupling interface 220 (i.e., *DecouplingIF::Init()*) may cause the code in decoupling interface 220's DLL to issue an initialization call exposed by black box 240 (e.g., *BlackBox::Init*). Preferably, one of the arguments to *BlackBox::Init* may be a binary certificate that contains a cryptographic hash of the code of decoupling interface 220 signed by the private key of a certifying authority, where the  
10 corresponding public key is available to (e.g., inside) black box 240. Black Box 240 reads the code of decoupling interface 220, for example by using the system call *ReadProcessMemory* (or a private in-process protocol that accesses the address space of decoupling interface 220), and computes its hash and compares it to the one signed by the certifying authority. *BlackBox::Init* may also check that the return address from  
15 its *Init* call is in the execution space of decoupling interface 220. Decoupling interface 220 may also receive two additional certificates from black box 240: a certificate from a certifying authority (which may or may not be the same certifying authority that signs the certificate of decoupling interface 220) binding the name and public key of the developer of application program 244, and also a binary certificate from the developer  
20 of application program 244 naming the hash, security level and expiration of its code. Decoupling interface 220, which is in the same address space as application program 244, computes the hash of the application code, verifies the certificate and compares the computed hash to the signed hash. These action may be performed before decoupling interface 220 returns from the initialization call issued by application  
25 program 244.

Additionally, during the call to *BlackBox::Init*, black box 240 preferably attaches to application program 244 using the *DebugActiveProcess* system call (or similar call, when an operating system other than one of the MICROSOFT WINDOWS operating systems is being used), which has the advantageous effect of preventing user  
30 mode debuggers from attaching to application program 244. Preferably, no sensitive code within black box 240, and no key within the black box 240 (e.g., key 248) are

- 41 -

“uncovered” until this attachment is done, internal black box integrity checks are performed, and the code of application program 244 and/or decoupling interface 220 have been authenticated. Thereafter, all sensitive data is “poked” into the application process (using WriteProcessMemory, or a private in-process interface that accesses the address space of the application process) and does not travel via COM, RPC or system buffers.

Preferably, in order to establish a secure environment, system components used by black box 240, decoupling interface 220, and application program 244 are also authenticated. Components to be authenticated may include user libraries, kernel components, or any other software that has the potential to affect the security of the environment in which protected actions are to be performed.

At the conclusion of step 904, establishment of a chain of trust between black box 240 and application program 244 is complete. Black box 240 may now proceed to perform sensitive functions, such as cryptographic services, for application program 244.

At step 905, application program 244 requests services from black box 240. For example, black box 905 may provide or identify data object 800 and request that that data object 800 be decrypted if permitted by license 816. In the case where decoupling interface 220 is present, application program 244 may make this request through decoupling interface 220, which then issues the necessary instructions to black box 240.

At step 906, black box 240 performs functions necessary to validate license 816 and provide application program 244 with access to data object 800. For example, in the exemplary architecture depicted in FIG. 8 where license 816, key 820, and content 204 are cryptographically “sealed” together, black box 240 may use its (preferably hidden) private key 248b to decrypt private key 808 of certificate 812, and then may use private key 808 to unseal data object 800 and read its license 816. Black box 240 may then proceed to evaluate license 816, checking whatever conditions are necessary to perform the evaluation. For example, license 816 may permit decryption of data object 800 up to a particular date, in which case the license evaluation process includes checking a (preferably trusted) calendar source. As another example, license

- 42 -

816 may permit data object 800 to be used only in a particular environment (e.g., the license may permit data object 800 to be used on an isolated hardware device but not on an open platform), in which case the evaluation process includes the act of identifying the environment.

5           If the license permits use of data object 800, black box 240 may obtain sealed key 820 and use it to provide "enabling bits" 824 to application program 244 at step 907. In one exemplary embodiment, enabling bits 824 may include sealed key 820, which application program 244 uses to object 800's encrypted information 204. In another exemplary embodiment, black box 240 uses sealed key 820 to convert  
10 encrypted information 204 into decrypted information 208 and then provides decrypted information 208 to application program 244. In the latter example, the decrypted information itself is the "enabling bits" 824. When decoupling interface 220 is present, black box 240 preferably provides enabling bits 824 to application program 244 through decoupling interface 220.

15           It is noted that the foregoing examples have been provided merely for the purpose of explanation and are in no way to be construed as limiting of the present invention. While the invention has been described with reference to various embodiments, it is understood that the words which have been used herein are words of description and illustration, rather than words of limitations. Further, although the  
20 invention has been described herein with reference to particular means, materials and embodiments, the invention is not intended to be limited to the particulars disclosed herein; rather, the invention extends to all functionally equivalent structures, methods and uses, such as are within the scope of the appended claims. Those skilled in the art, having the benefit of the teachings of this specification, may effect numerous  
25 modifications thereto and changes may be made without departing from the scope and spirit of the invention in its aspects.

## APPENDIX A

### Background

30           Let  $e$  be the exponent to be computed.

- 43 -

Let  $e_{acc}$  be the exponent of the "accumulation," the number that accumulates the message raised to  $e$ .

Let  $e_{err}$  be the current error exponent.

When using the binary method to compute an addition chain,  $e$  is scanned from left to right. For each scanned bit, the accumulation is squared, which doubles its exponent (the shift stage:  $e_{acc} = e_{acc} << 1$ ). If the scanned bit is 1, the accumulation is then multiplied by the message, which increases its exponent by 1 (the add stage:  $e_{acc} = e_{acc} + 1$ ). This process may be characterized as shifting the MSB off  $e$  and on to  $e_{acc}$ . The add stage effectively corrects the exponent error that accumulated when the exponent was left shifted.

The  $m$ -ary method generalizes this method by shifting  $m$  bits at a time ( $e_{acc} = e_{acc} << m$ ). At each step, the exponent error that is corrected in the add stage equals the  $m$ -bit number currently being scanned ( $e_{err} = e_m$ ). Usually the error is resolved with one multiply using a pre-computed table of messages raised to all possible  $m$ -bit exponents.

In greater generality, when the exponent error is "resolved" by multiplying from a pre-computed table, the exponent in the table is effectively subtracted from the current exponent error (the reduction stage:  $e_{err} = e_{err} - e_l$ ). In the binary and  $m$ -ary methods, the reduction exponent is always chosen to equal the current error, so that the reduction always yields an error of zero ( $e_l = e_m$ ). However, a random exponent could be chosen to reduce the error as long as it was less than or equal the current error. The excess error would be carried into the next shift stage ( $e_{err} = (e_{err} << m) + e_m$ ).

#### Random Addition Chains:

A library of functions (e.g., `Addchain.{h,cpp}`) can be created which describe a class that computes a random addition chain for a big number exponent. Computation of the addition chain assumes a storage mechanism, the "register store," that holds several big numbers (typically 15) and the original message. These registers

- 44 -

store the message raised to various 29-bit exponents as well as the number that accumulates the message raised to the total exponent (the accumulation).

The register store is primed with random exponents computed using random multiplications between already computed exponents, starting with the original message in one of the registers. While accumulating  $e$ , multiplications between random registers are performed and stored, so that “pre-computed” exponents are constantly changing. These random multiplications are tuned so that register exponents do not exceed 29 bits.

While accumulating, the exponent error is not allowed to grow more than twice as large as the largest exponent in the register store. If this limit is about to be exceeded, the error is reduced. These “required” reductions prefer larger exponents to keep the number of required multiplies small. “Optional” reductions randomly occur if there exists any register that is less than or equal the current error. The parameters for these required and optional reductions are tuned so that about 700 multiplies occur for a 512-bit exponent in addition to the 512 necessary squarings – about 3 reductions for every 2 shifts.

Since squarings are treated as multiplies, reductions are fairly unpredictable, and extraneous multiplies frequently occur, an opponent must track the contents of all the registers to determine the accumulating exponent.

#### 20 Register Store:

Two categories of techniques are used to make it difficult to track the contents of a virtual register: data obfuscation and engineering tricks. Data obfuscation involves storing registers redundantly, swapping data stored in two registers, and hiding data by XOR'ing it with another register. Engineering tricks help defeat common techniques used for backwards engineering code, such as fall through cases on switch statements or extraneous no-op calculations injected between real calculations.

#### Data Obfuscation:

The register store contains 66 “real” registers that are used to store data for the 16 “virtual” registers used by the addition chain calculation. A virtual register can keep track of up to 4 redundant copies of its data with 4 “slots” that reference real registers. When data needs to be stored, a real register is randomly selected from the

- 45 -

set of unallocated registers. Real registers are released to the unallocated store when a virtual register is overwritten. Dynamically allocating real registers to a virtual register prevents an adversary from statically examining data stored at particular memory offsets to determine the contents of the virtual register.

5           When data is needed, one of the slots is randomly selected. Duplicating data for redundancy is scheduled randomly, but the process is biased to prefer data that has been used recently (ex. as a source or result of a multiply). Slots are sometimes released randomly, even though the data in the real register is not changed and might appear to be in use. This way, a virtual register's data may actually appear in more  
10 than 4 real registers even though the system is only selecting data from one of the 4 slots it is tracking.

          In the process of copying data, a real register may also become XOR'd with another random real register, to mask its contents. The other register may or may not contain data that is being actively used. In fact, a copy operation may change the  
15 XOR state of up to four other registers, in addition to the real registers being copied to and from. These six registers can also swap their contents in various ways during a copy via a series of XOR's. Thus, a particular copy generally involves several XOR's that can change both the XOR state of several real registers and the real registers referenced by virtual register slots.

20           Dynamic register allocation, surreptitious swapping, and XOR'ing constitutes a kind of "shell game," where data is being shuffled around and masked in memory before and after virtually every multiplication. Data stored in one register may be retrieved from an entirely different register later on. The same register may never be used to retrieve the same piece of data twice in a row. The XOR'ing makes it more  
25 difficult to track the swapping in memory.

#### Engineering Tricks

          Static analysis of the register store's memory is deterred in three ways. Memory is allocated on the heap so techniques that examine offsets from the stack pointer do not work. Random blocks of unused space are also added between real  
30 registers so that obvious offsets cannot be used to examine a particular register.

- 46 -

Finally, the register store is initialized with random content so that unused or unprimed registers are not obvious by examination.

The bulk of the code in rsagen.cpp is devoted to generating the `_Copy` function, and tracking the state changes that it performs as side effects. The `_Copy` function performs the copy/swap/state-change operation for the data in the register store. A single 32-bit unsigned integer is passed to `_Copy` that encodes the type of copy that is to occur. Part of this UINT encodes the 6-8 real registers involved in the copy/swap/state-change, and another part performs the actual calculation. Both of these parts are implemented with large switch statements (150+ cases) that make extensive use of fall through cases. Fall through makes it more difficult to analyze the code generated for the switch based on jumps, since 1 to 4 cases may use the same break.

The major vulnerability of the system up to this point involves a dynamic attack, where the adversary breaks out of execution at each multiply and examines the data going in and out. If the adversary knows what exponent is represented by the source data, it can compute the exponent of the result. Since the whole system starts with the original message, the adversary could backwards engineer any exponent using this attack – most notably the secret exponent.

This is deterred in two ways. First, data for the sources and result of the multiply are directly read from and written to the register store, without making an intermediate copy to stack memory. This makes it more difficult for an automatic breaker to analyze the stack frame to infer exponents. Second, a percentage of the calculations are performed using inlined copy cases and multiplies. These inlined calculations mostly occur at the beginning of the computation, but other clusters are scattered throughout the calculation randomly. With copies and multiplies mixed together, it should be more difficult to determine where exactly a multiply is occurring and therefore where to examine data to backwards engineer exponents.

#### Vulnerability and Possible Solution:

Currently, the greatest vulnerability of the system is still the dynamic attack. If an adversary can infer where multiplies are occurring in the inlined code, it might still backwards engineer the exponents automatically. This will be difficult with

- 47 -

BBT code shuffling and bogus code injected into the real calculations – but it is still conceivable.

A number theory result could be used to make this more difficult. If during a copy, the code performed a surreptitious add much like the swaps and XOR's, the code could mask a source value of the multiply. After the multiply, a second number is added to the result, such that the total amount added is congruent to zero with respect to the modulus.

Given  $(A+B) \bmod N = 0$ , it is needed to hide the sources and result of the multiply:

$$\begin{aligned}
 10 \quad & (X*Y) \bmod N = \\
 & (X*Y + (A+B)*Y) \bmod N = \\
 & (X*Y + A*Y + B*Y) \bmod N = \\
 & ((X + A) * Y + B*Y) \bmod N
 \end{aligned}$$

So, add A to X before the multiply, and  $(B*Y) \bmod N$  after the multiply. It is possible to add  $(B*Y) \bmod N$  after the mod – as long as the 512-bit number does not overflow with the add, and as long as another mod is done before the end. The copy code is designed to support a surreptitious add, although this new kind of state is not implemented in the state tracking code.



- 48 -

CLAIMS

## WHAT IS CLAIMED IS:

1. A method of facilitating the use of a software process with one of a plurality of secure repositories, said method comprising the acts of:

5                   providing an interface, said interface being callable by said software process;

                  if said one of said plurality of secure repositories is said first of said plurality of secure repositories, providing a first set of computer-executable instructions which are invocable by said callable interface; and

10                   if said one of said plurality of secure repositories is said second of said plurality of secure repositories, providing a second set of computer-executable instructions which are invocable by said callable interface, said second set of computer-executable instructions being different from said first set of computer-executable instructions.

15

2. The method of claim 1, wherein said secure repository converts encrypted data to decrypted data using a cryptographic algorithm to apply a cryptographic key to said encrypted data, and wherein said software process performs an operation on said decrypted data.

20

3. The method of claim 2, wherein said operation comprises rendering said decrypted data.

4. The method of claim 1, wherein said first or said second sets of  
25 computer-executable instructions is provided in the form of an executable file dynamically linkable with said software process.

5. The method of claim 1, wherein said interface comprises a first function callable by said software process, said first function being parameterized by  
30 first data representative of a type of secure repository.

- 49 -

6. The method of claim 5, wherein said interface is callable by said software process without regard to whether said one of said plurality of secure repositories is said first of said plurality of secure repositories or said second of said plurality of secure repositories.

5

7. The method of claim 1, wherein said interface comprises a second function callable by said software process, said second function requesting that said secure repository perform at least one action.

10

8. The method of claim 1, wherein said first of said plurality of secure repositories executes on a closed-platform device, and wherein said second of said plurality of secure repositories executes on an open-platform device.

9. A method of communicating between a software process and a one of  
15 a plurality of secure repositories, said method comprising the acts of:

said software process issuing a first interface call which authenticates said software process to said one of said plurality of secure repositories; and

said software process issuing a second interface call which  
20 requests performance of an action by said secure repository for said software process; wherein said software process issues said first and second interface calls without regard to whether said one of said plurality of secure repositories is a first of said plurality of secure repositories or a second of said plurality of secure repositories.

25

10. The method of claim 9, wherein said secure repository converts encrypted data to decrypted data using a cryptographic algorithm to apply a cryptographic key to said encrypted data, and wherein said software process performs an operation on said decrypted data.

30

11. The method of claim 10, wherein said operation comprises rendering said decrypted data.

- 50 -

12. The method of claim 9, wherein said first secure repository comprises a software-based secure repository, and wherein said second secure repository comprises at least some isolated hardware.

5

13. The method of claim 9, wherein each of said first and second secure repositories are software-based repositories, said first secure repository having at least one feature not present in said second secure repository.

10

14. The method of claim 9, wherein said one of said plurality of secure repositories is said first of said plurality of secure repositories, and wherein said software process issues said first and second interface calls without regard to whether said second repository exists.

15

15. The method of claim 9, wherein said first interface call is parameterized by first data representing a first type of secure repository, and wherein said first and said second of said plurality of secure repositories are each of said first type.

20

16. The method of claim 15, wherein said software process performs a second action if said one of said plurality of repositories is either said first or said second of said plurality of secure repositories, and wherein said software process does not perform said second action if said one of said plurality of secure repositories is a third of said plurality of secure repositories, said third of said plurality of secure repositories being of a second type different from said first type.

25

17. The method of claim 9, further comprising the acts of:

dynamically linking to said software process a first set of computer-executable instructions, if said one of said plurality of repositories is said first of said plurality of secure repositories; and

30

- 51 -

dynamically linking to said software process a second set of computer-executable instructions different from said first set of computer-executable instructions, if said one of said plurality of secure repositories is said second of said plurality of secure repositories.

5

18. The method of claim 9, further comprising the act of said software process receiving second data in response to said second interface call, said second data being generated by said one of said plurality of secure repositories, wherein said second data does not expose to said software process whether said data was generated  
10 by said first secure repository or said second secure repository.

19. A computer-readable medium encoded with computer-executable instructions to perform the method of claim 9.

15

20. A secure repository comprising:

a first set of computer-executable instructions which converts encrypted data into decrypted data by applying a cryptographic key to said encrypted data; and

a second set of computer-executable instructions which provides  
20 said decrypted data to a software process if said secure repository trusts said software process;  
wherein said secure repository establishes trust of said software process at least in part by establishing trust with an intermediate object, said intermediate object comprising a third set of computer-executable instructions dynamically linked to said software  
25 process.

21. The secure repository of claim 20, wherein said software process renders said decrypted data.

30

22. The secure repository of claim 20, further comprising a fourth set of computer-executable instructions which establishes trust with said intermediate object,

- 52 -

said fourth set of computer-executable instructions including instructions to perform acts comprising:

receiving from said intermediate object first data comprising:

second data based at least in part on at least some code

5 contained in said intermediate object; and

a signature of said second data; and

validating said signature.

23. The secure repository of claim 22, wherein said second data  
10 comprises a hash of said at least some code.

24. The secure repository of claim 22, wherein said fourth set of computer-executable instructions further performs acts comprising:

receiving from said intermediate object second data based at least

15 in part on code contained in said software process.

25. A method of communicating with one of a plurality of secure repositories, said method comprising the acts of:

issuing a first interface call without regard to whether said one of

20 said plurality of secure repositories is a first of said plurality of secure repositories or a second of said plurality of secure repositories;

if said one of said plurality of secure repositories is said first of said plurality of secure repositories, dynamically linking with a first set of computer-executable instructions invocable by said first interface call; and

25 if said one of said plurality of secure repositories is said second of said plurality of secure repositories, dynamically linking with a second set of computer-executable instructions invocable by said first interface call, said second said of computer-executable instructions being different from said first set of computer-executable instructions.

30

- 53 -

26. The method of claim 25, wherein each of said plurality of secure repositories converts encrypted data to decrypted data using a cryptographic algorithm to apply a cryptographic key to said encrypted data.

5           27. The method of claim 25, wherein said first secure repository comprises a software-based secure repository, and wherein said second secure repository comprises at least some isolated hardware.

28. The method of claim 25, wherein each of said first and second  
10 secure repositories are software-based repositories, said first secure repository having at least one feature not present in said second secure repository.

29. The method of claim 25, wherein said act of performing said first action comprises executing a first set of computer-executable instructions, and wherein  
15 said first action comprises the act of providing to said first secure repository first data based at least in part on at least some of said first set of computer-executable instructions.

30. A computer-readable medium encoded with a second set of  
20 computer-executable instructions to perform the method of claim 25.

31. A method of authenticating a first software process to a second software process, said method comprising the acts of:

                  establishing to said second software process the authenticity of an  
25 intermediary object; and  
                  using said intermediary object to establish to said second software process the authenticity of said first software process.

32. The method of claim 31, wherein said second software process  
30 converts encrypted data to decrypted data by using a cryptographic algorithm to apply a

- 54 -

cryptographic key to said encrypted data, and wherein said first software process performs an operation on said decrypted data.

33. The method of claim 32, wherein said operation comprises rendering  
5 said decrypted data.

34. The method of claim 33, wherein said first software process is a text-rendering application, and wherein said decrypted data comprises text.

10 35. The method of claim 31, wherein said intermediary object comprises a set of computer-executable instructions having a first function callable from said first software process, and wherein the act of establishing to said second software process the authenticity of said intermediary object includes, or is actuated by, the act of said first software process calling said first function.

15 36. The method of claim 35, wherein said act of establishing to said second software process the authenticity of said intermediary object includes the act of providing said second software process with a certificate based at least in part on said set of computer-executable instructions.

20 37. The method of claim 36, wherein said certificate comprises a signed hash of at least some of said computer-executable instructions.

38. The method of claim 35, wherein said intermediary object is in the  
25 address space of said first software process, and wherein said first function is referenceable by an address within said address space.

39. The method of claim 35, wherein said set of computer-executable instructions is dynamically linkable with said first software process, and wherein said  
30 method further comprises the act of linking said set of computer-executable instructions with said first software process.

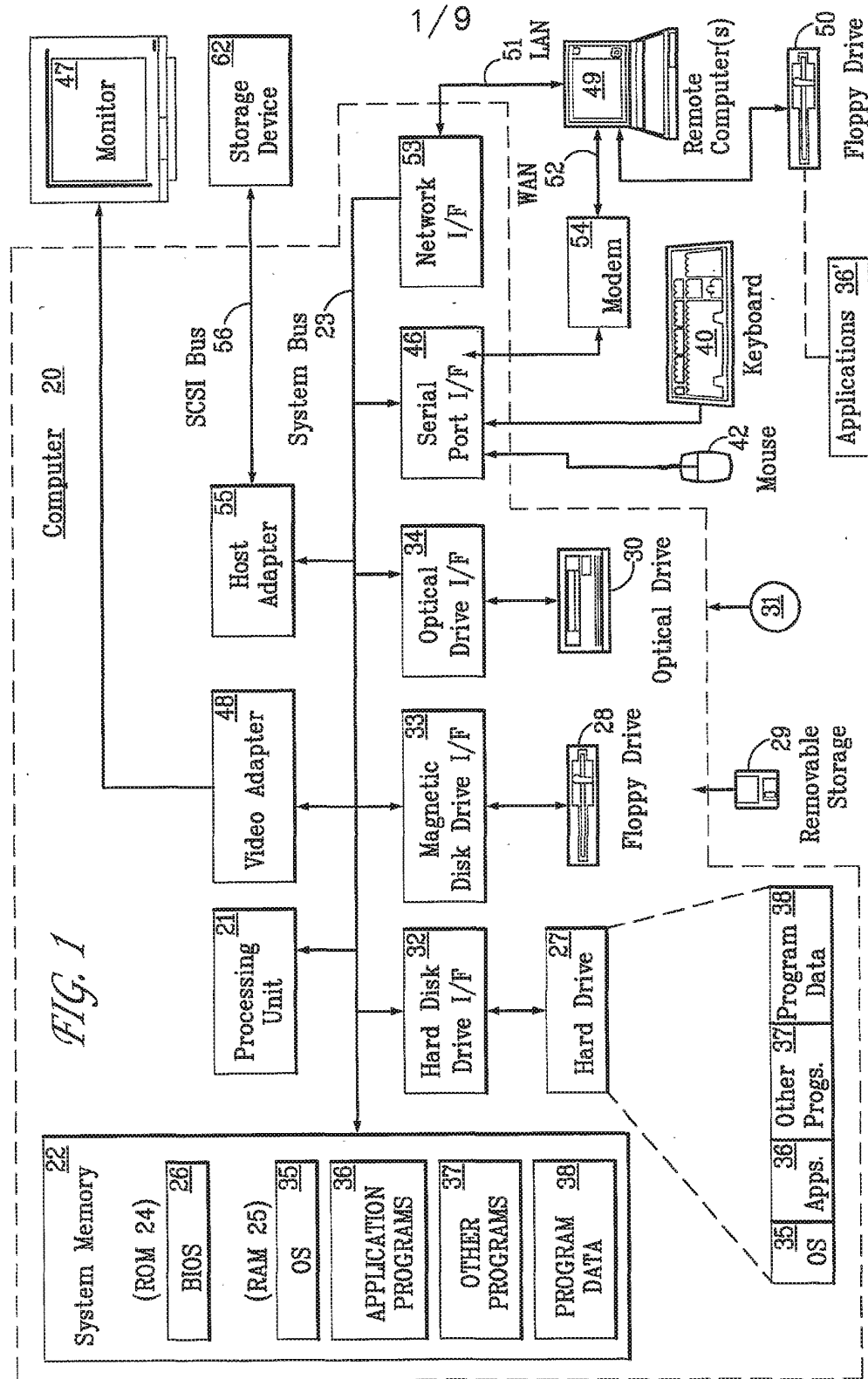
- 55 -

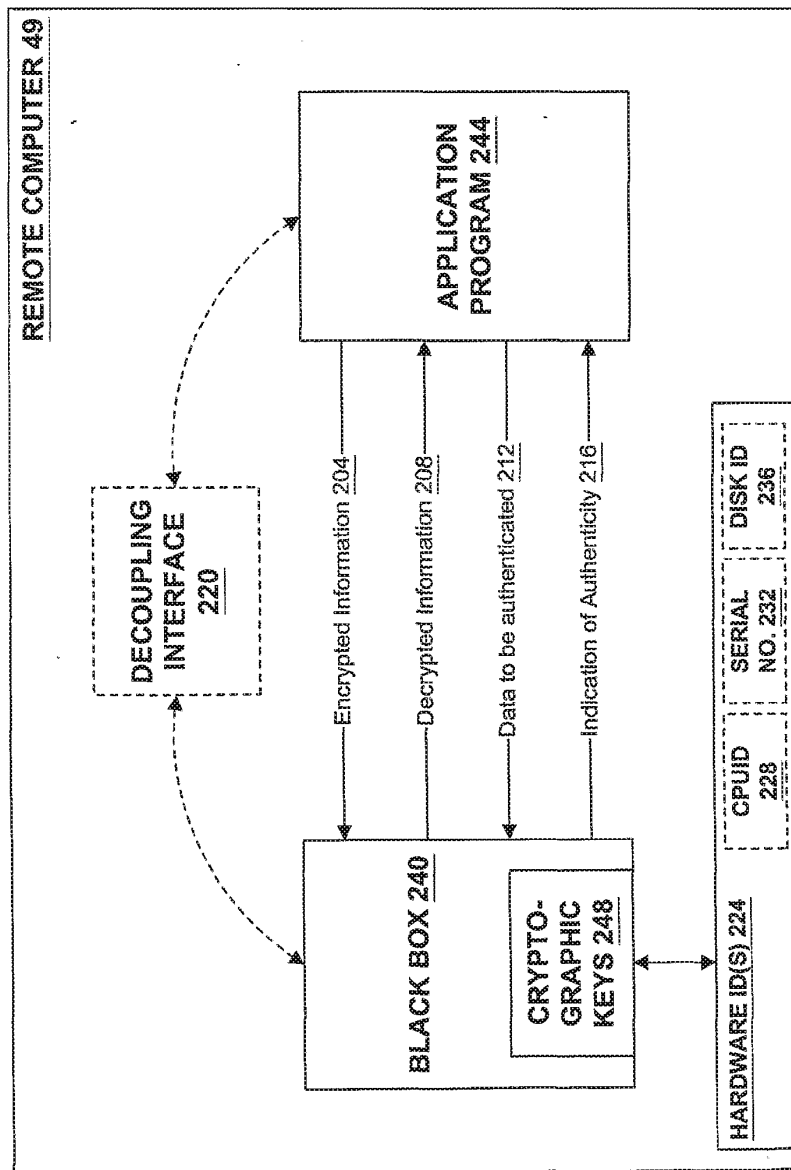
40. The method of claim 31, wherein said intermediary object comprises a set of computer-executable instructions having a first function callable from said first software process, and wherein said act of using said intermediary object to establish to  
5 said second software process the authenticity of said first software process includes, or is actuated by, the act of said first software process issuing a call to said first function.

41. A computer-readable medium encoded with a second set of computer-executable instructions to perform the method of claim 31.

10

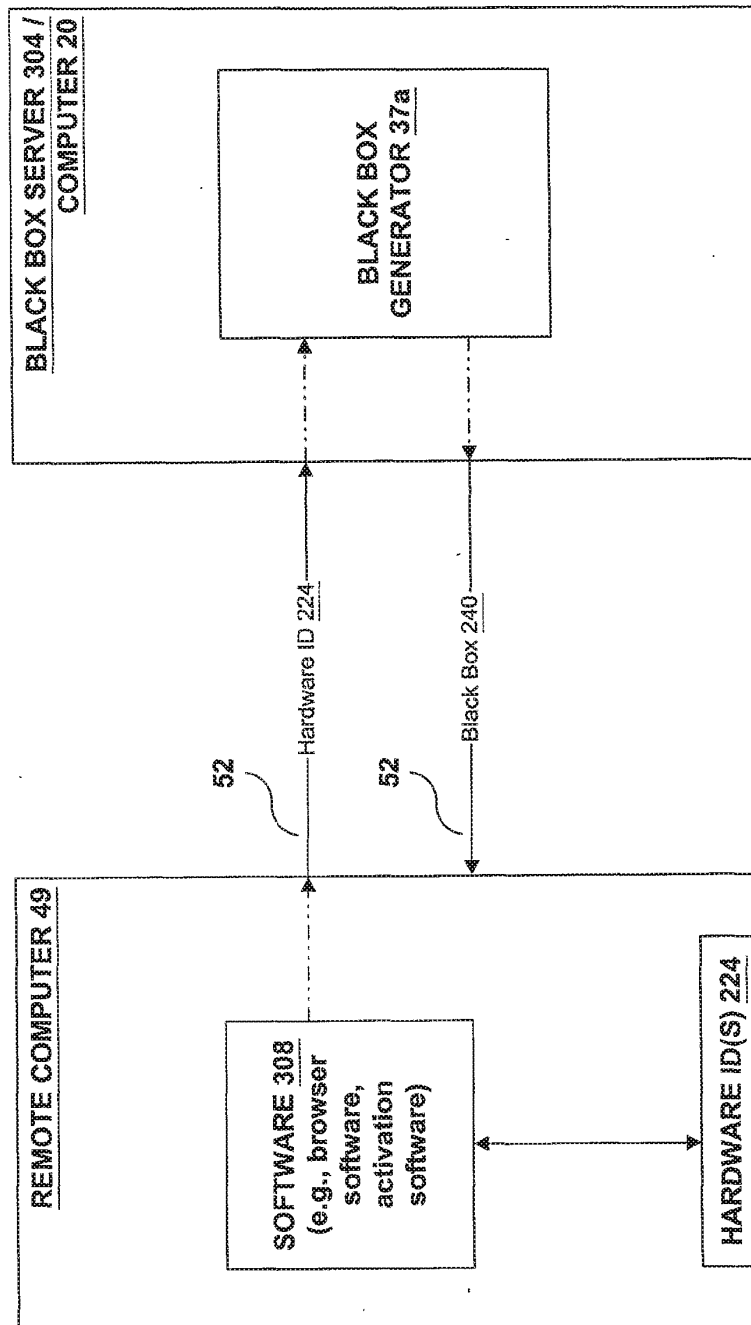






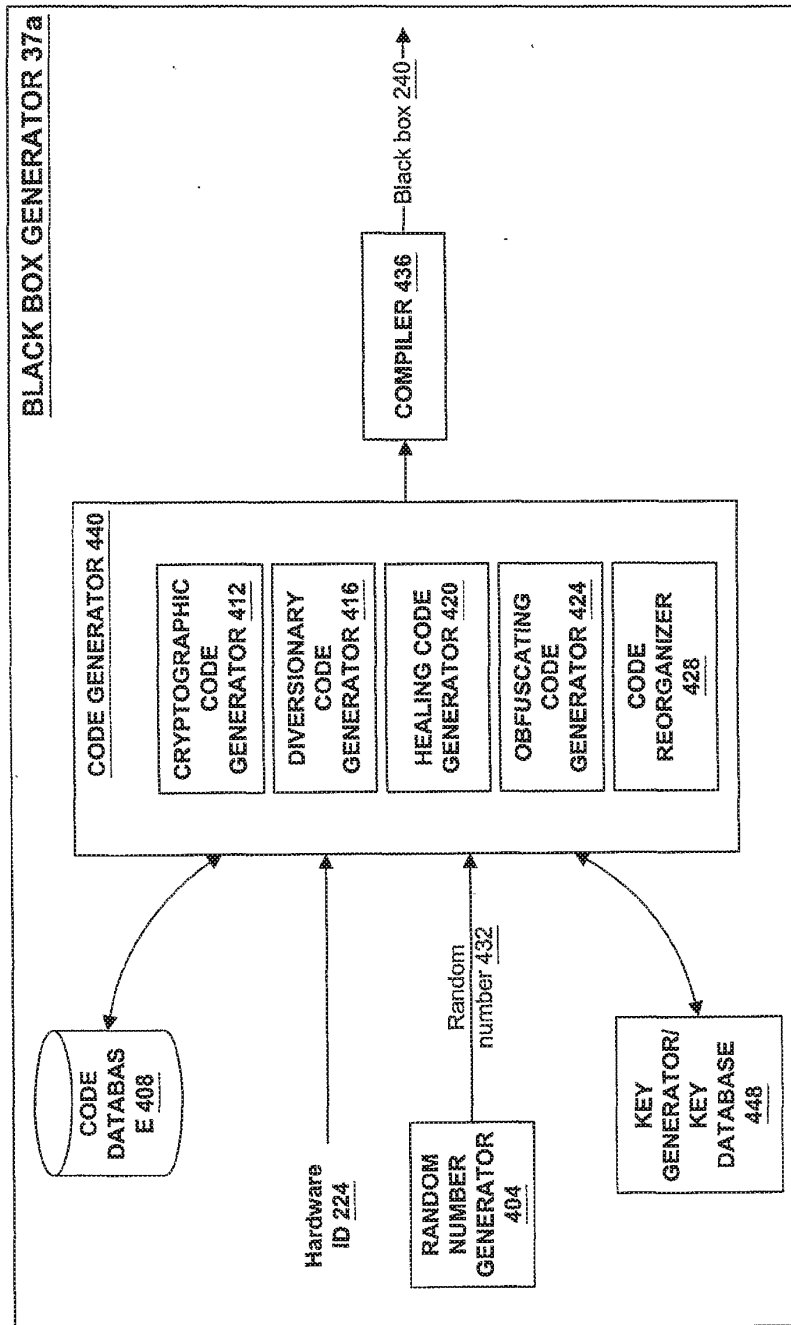
**FIG. 2**

3/9



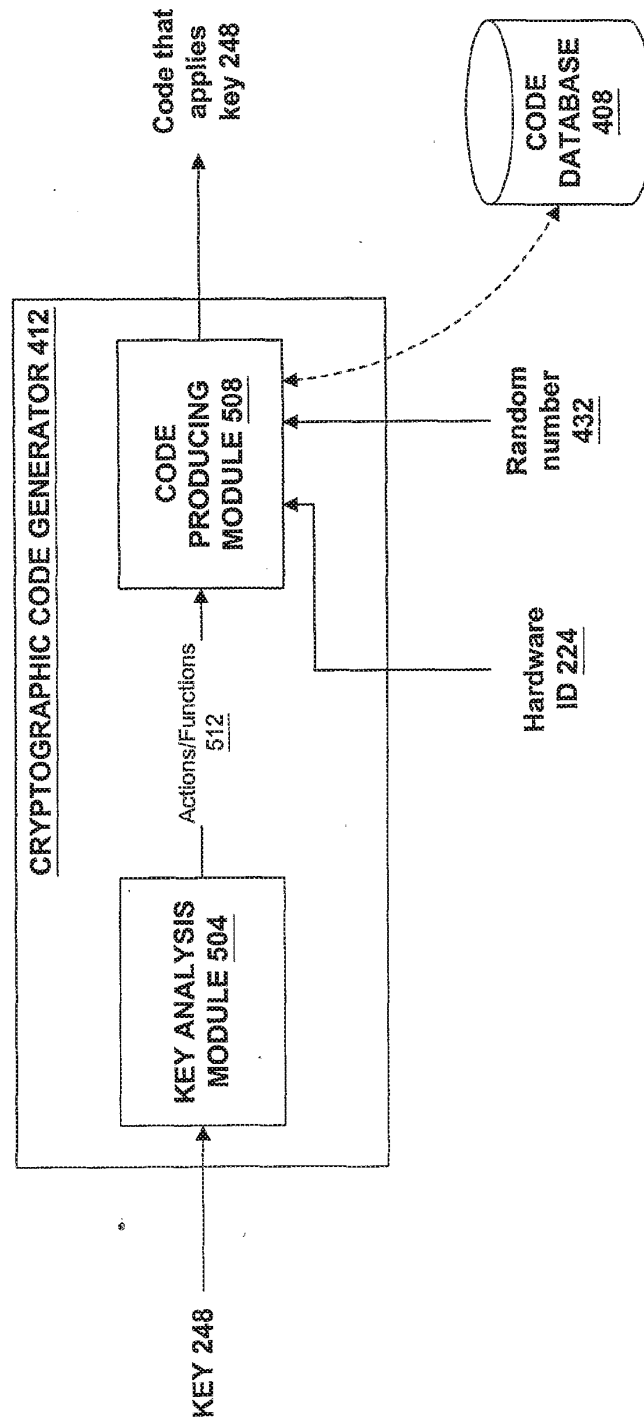
**FIG. 3**

4/9



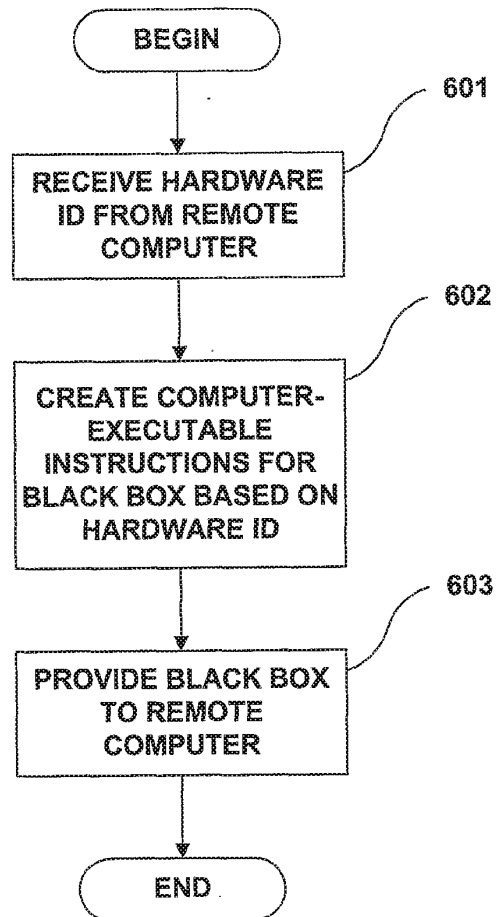
**FIG. 4**

5/9

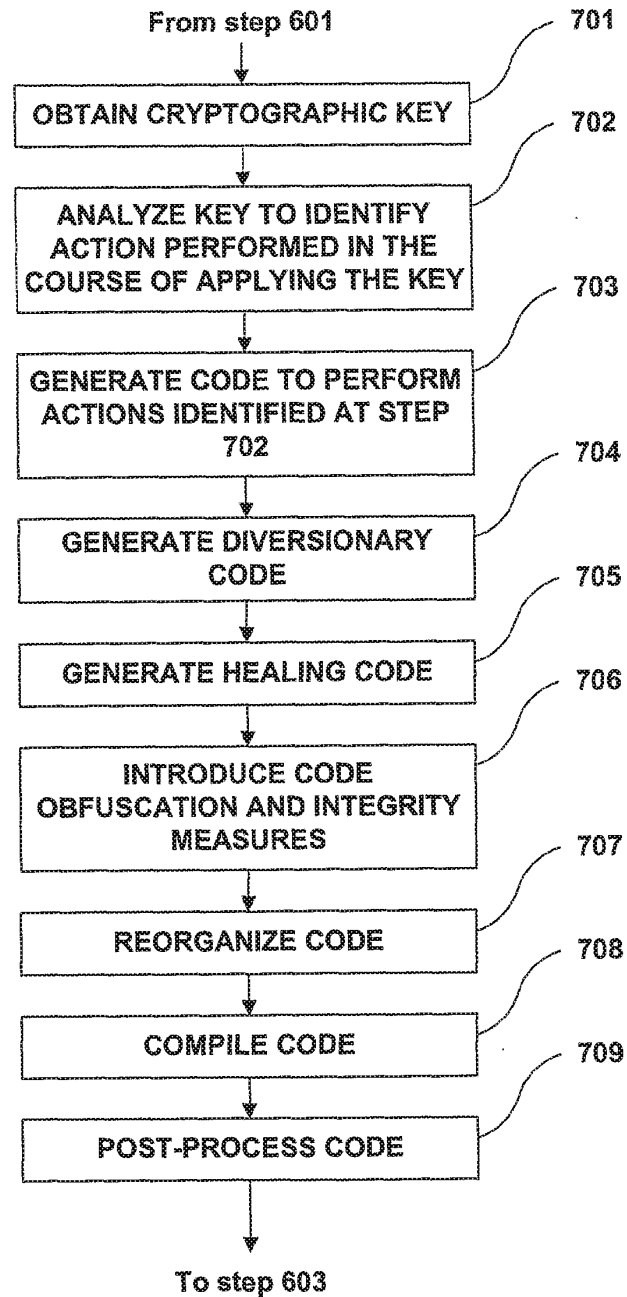


**FIG. 5**

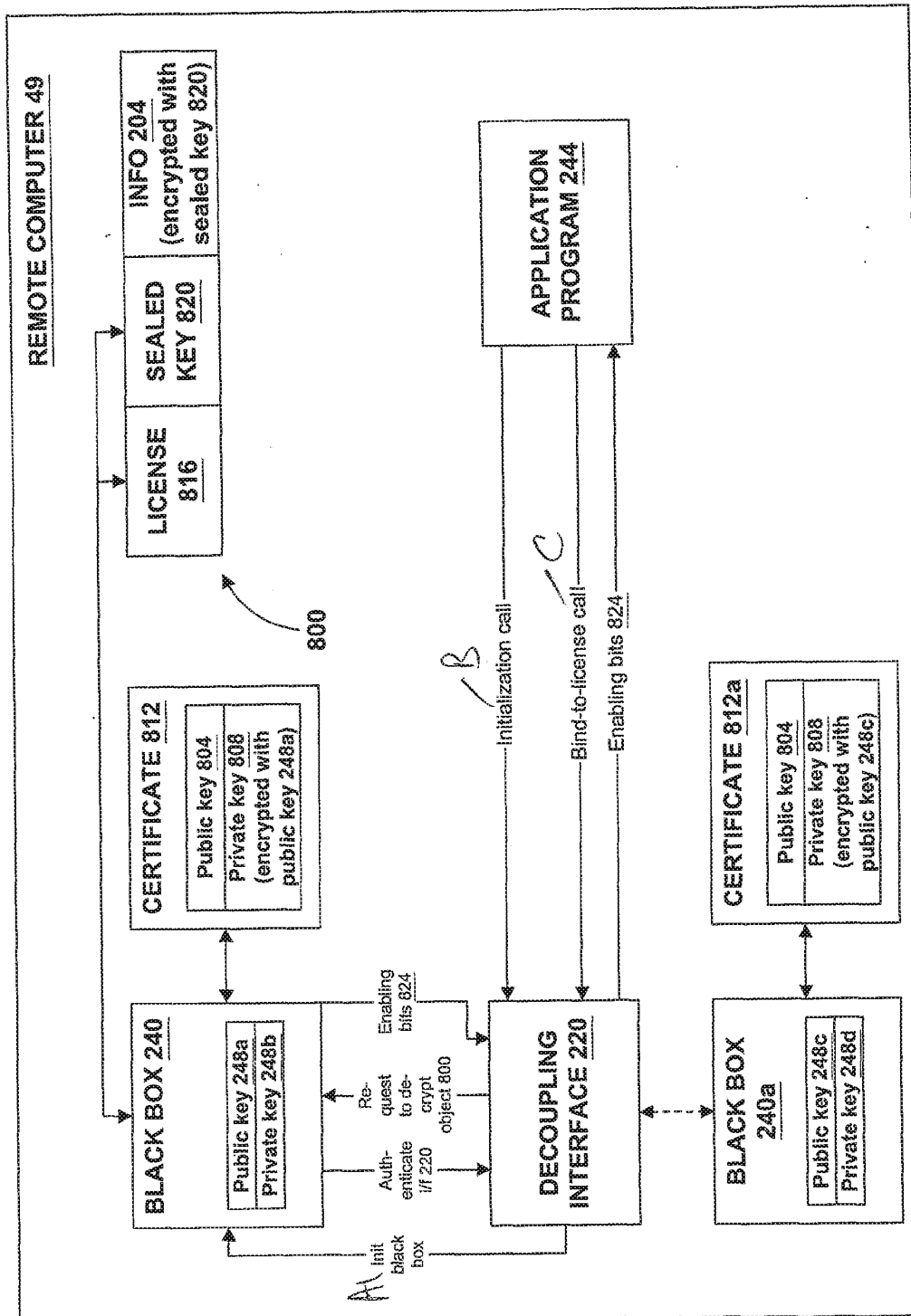
6/9

**FIG. 6**

7/9

**FIG. 7**

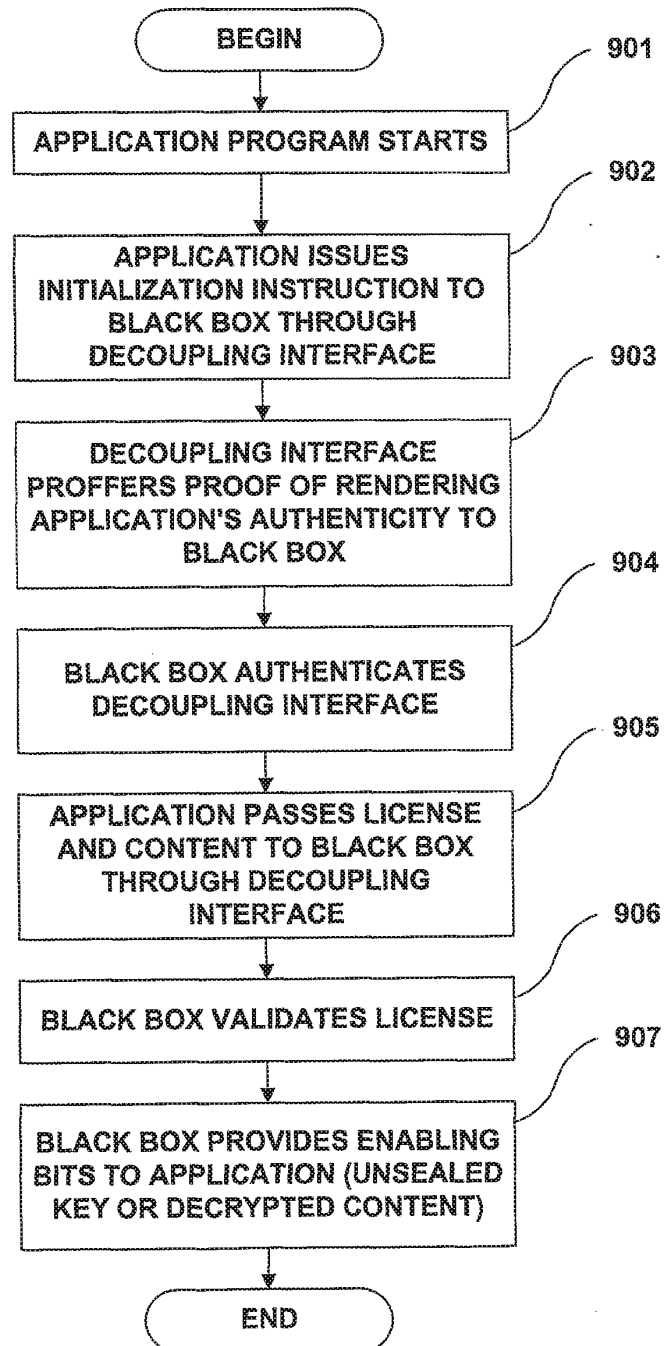
8/9



**FIG. 8**



9/9

**FIG. 9**

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
28 June 2001 (28.06.2001)

PCT

(10) International Publication Number  
WO 01/46918 A2

- (51) International Patent Classification<sup>7</sup>: G07F
- (21) International Application Number: PCT/US00/41736
- (22) International Filing Date: 31 October 2000 (31.10.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/437,065 9 November 1999 (09.11.1999) US
- (71) Applicant: ARCOT SYSTEMS, INC. [US/US]; Suite 200, 3200 Patrick Henry Drive, Santa Clara, CA 95054 (US).
- (72) Inventor: KAUSIK, Balas, Natarajan; 18079 Reed Knoll Road, Los Gatos, CA 95030 (US).
- (74) Agents: LAURIE, Ronald, S. et al.; Skadden, Arps, Slate, Meagher & Flom LLP, 525 University Avenue, Palo Alto, CA 94301 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— Without international search report and to be republished upon receipt of that report.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR SECURE AUTHENTICATED PAYMENT ON A COMPUTER NETWORK

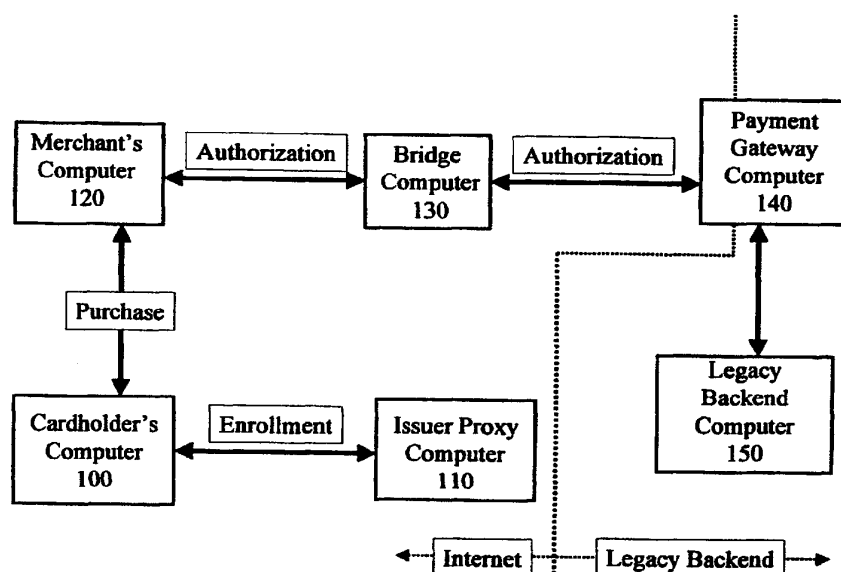


Exhibit E12

EP 3 229 099  
(17000713.2)

BARDEHLE PAGENBERG  
Our ref.: S154820EPT1EIN

(57) Abstract: A simple, secure and easy-to-deploy method and system for authenticating credit and debit cardholders at the point-of-sale on a computer network (e.g. the Internet) is disclosed. Cardholders are authenticated using digital signatures on a sales draft, in a manner that does not necessarily require any changes in the transaction flow of the participating financial institutions.

## **METHOD AND SYSTEM FOR SECURE AUTHENTICATED PAYMENT ON A COMPUTER NETWORK**

### **FIELD OF THE INVENTION**

5       The present invention relates to a method and system for secure authenticated payment at a point-of-sale on a computer network. More particularly, the present invention allows the use of digital signatures on a sales draft to authenticate purchasers in a manner that does not necessarily require any changes in the transaction processing of the financial institutions participating in the transaction.

### **BACKGROUND OF THE INVENTION**

10       In present electronic commerce transactions, buyers may pay for goods and services by presenting the seller with a payment card number, e.g., a conventional credit card number. Because the buyer and seller are connected solely through a computer network (e.g., the Internet), it is not possible for the buyer to authenticate himself as the legitimate cardholder, nor can the buyer sign the sales draft. Thus, the seller honors any  
15       valid credit card number that is presented, creating a large opportunity for fraud.

      Worse yet, other forms of payment such as debit cards are not presently viable on computer networks. Debit cards require the cardholder to enter a personal identification number ("PIN"), which is used to authenticate the transaction to the cardholder's bank. However, entering a simple PIN on a networked computer poses a substantial security  
20       risk—if the PIN and the debit-card number fell into the wrong hands, the cardholder's bank account would be completely compromised.

      Thus, with respect to both conventional credit and debit cards, authenticating a cardholder on the network with a solution that is simple, secure, and easy to deploy remains an important unsolved problem.

25       Digital signature technology offers one means of authenticating the cardholder with a high degree of security. In this technology, each cardholder owns a pair of keys – a signature (private) key and a verification (public) key. The cardholder signs a transaction

with his private key, and then sends the transaction, the digital signature, and (optionally) his public key to the merchant. The merchant forwards these items to the bank (or other financial institution), and the bank honors the transaction if the cardholder's public key verifies the cardholder's digital signature.

5           One security advantage of digital signatures is that the private key of the cardholder typically remains in possession (or at least control) of the cardholder. Thus, there is no inherent risk associated with a transaction that would compromise future transactions. One disadvantage of the digital signature method described above is that banks and transaction processors would have to change their existing infrastructure to allow digital  
10           signatures to flow through their networks. This infrastructure change would basically require a substantial overhaul of the present electronic banking and transaction processing system, which is costly and difficult to achieve.

          Thus, there is a need for a method and system that offers the security advantages of digital signatures without necessarily requiring significant changes in the banking and  
15           processing network.

## SUMMARY OF THE INVENTION

          One embodiment of the present invention includes a simple, secure and easy-to-deploy method and system for authenticating credit and/or debit cardholders at a point-of-sale on a computer network (e.g., the Internet). Cardholders are authenticated using digital  
20           signatures on a sales draft, in a manner that does not necessarily require any changes in the transaction process of the financial institutions participating in the transaction.

          In this embodiment of the system, the cardholder enrolls for an electronic payment card (either an electronic debit or credit card) at a participating financial institution by visiting its issuer proxy enrollment site, e.g., a web site hosted by an issuer proxy com-  
25           puter associated with the financial institution. At the enrollment site, the cardholder types in his particulars, such as his conventional payment card number, conventional payment card PIN, name, address, etc. The cardholder also (optionally) selects a password (access code) for his electronic payment card that is preferably unrelated to the PIN for his conventional payment card. The issuer proxy generates a public key-private key pair for

use by the cardholder if the cardholder does not already have such a pair. The issuer proxy binds the cardholder's public key and some or all of the cardholder's payment particulars in a digital certificate using an encryption key (called a domain key) that is shared between the issuer proxy and a bridge computer. Such a domain key will allow the bridge computer to confirm the issuer's certification during a subsequent authorization stage, described below. The cardholder then receives a piece of software that is downloaded to his computer containing his particulars in encrypted form. This piece of software constitutes the cardholder's electronic payment card. It comprises (or is configured to obtain and use) the cardholder's private key, which is (optionally) protected by the password, and the corresponding public-key digital certificate containing the cardholder's payment particulars.

Thenceforth, as the cardholder shops online, he can elect to pay via electronic payment. To do so, the cardholder activates his electronic payment card with the previously selected password. The cardholder's electronic payment card software interacts with corresponding software at the online merchant to digitally sign the sales draft created during the transaction with the cardholder's private key. The merchant then sends the signed sales draft and the cardholder's digital certificate to the bridge computer for processing. The bridge computer uses the cardholder's digital certificate to check the digital signature on the sales draft. If the signature is valid, the bridge computer creates a conventional debit or credit transaction to be processed by the banking and transaction network. The particulars needed for creating the conventional transaction, such as the conventional card number and PIN, are extracted and decrypted from the cardholder's digital certificate using the private key associated with the domain key (if the digital certificate was asymmetrically encrypted) or the domain key itself (if the digital certificate was symmetrically encrypted). The embodiment of the invention described above provides one or more of the following advantages:

- (1) Additional hardware at the cardholder's computer is not necessarily required for deployment. This is in marked contrast to hardware tokens such as smart cards, where cards and card readers are required. Of course, the software comprising the cardholder's electronic payment card can be stored on smart cards, as well as

virtually any other storage medium, including, without limitation, floppy disks, hard drives, and magnetic stripe cards;

- (2) Changes are not necessarily required in the existing banking network;
- (3) Administrative overhead is low. The cardholder can enroll at any participating financial institution that offers the service, not necessarily the one that issued the cardholder's conventional payment card. Furthermore, enrollment can be on a self-serve basis and does not necessarily require activation mailings by the financial institutions;
- (4) Electronic payment cards can be deployed rapidly, because they are intuitive to use and require little user or administrator training; and/or
- (5) Security can be enhanced via special techniques such as "cryptographic camouflaging," which is commercially available from Arcot Systems, Inc.

The foregoing and other embodiments and aspects of the present invention will become apparent to those skilled in the art in view of the subsequent detailed description of the invention taken together with the accompanying figures and appended claims.

### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating an exemplary computer system for secure authenticated payment on a computer network.

FIG. 2 is a flow chart illustrating an exemplary method for cardholder enrollment for an electronic payment card.

FIG. 2A illustrates an exemplary electronic payment card created using the preferred embodiment of the invention.

FIG. 3 is a flow chart illustrating an exemplary method for point-of-sale interaction between a cardholder and a merchant.

FIG. 4 is a flow chart illustrating an exemplary method for a merchant to obtain authorization for the payment transaction.

## DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 is a block diagram illustrating an exemplary computer system for secure authenticated payment on a computer network (e.g., the Internet). The system contemplates a network of computers including a cardholder's computer **100**, a payment card issuer's proxy computer **110**, a merchant's computer **120**, a bridge computer **130**, a payment gateway computer **140**, and legacy backend computer **150**. In this exemplary embodiment, the network is deployed over the Internet, although those skilled in the art will recognize that any public or private communication network including, without limitation, extranets, intranets, and other telephonic or radio communications networks could also be used. Similarly, as used herein, the term computer refers to any device that processes information using an integrated circuit chip, including without limitation mainframe computers, work stations, servers, desktop computers, portable computers, embedded computers, and hand-held computers.

### Enrollment

Referring now to FIG. 2, at step **200**, a cardholder (user) at computer **100** enrolls for an electronic payment card (either an electronic debit card or an electronic credit card) at the electronic payment card issuer proxy **110**, typically by visiting the website of a participating financial institution on the Internet. At step **210**, the cardholder provides the issuer **110** with particular information used to make a payment (payment particulars), such as his conventional payment card number, conventional payment card PIN, conventional credit card holder verification value 2 ("CVV2"), conventional cardholder name and address, or any other cardholder identification information. The issuer proxy **110** can be operated by any trusted financial institution that participates in the electronic payment system, not necessarily the financial institution that issued the cardholder's conventional payment card.

The issuer proxy **110** can optionally verify the cardholder's payment information by any of the means available for such verification including, without limitation, creating a payment transaction in the conventional payment network. Such a transaction could be

“authorization only” in the sense that it would be used only for verifying the cardholder’s payment particulars, with no money actually transferred.

At step **220**, the issuer **110** generates a public key-private key pair for the cardholder to use in connection with the electronic payment system. If the cardholder already has a public key-private key pair that he wishes to use in connection with the electronic payment system, he provides his public key to the issuer **110**. The cardholder's private key is typically stored on the cardholder's computer **100**, often under the control of a PIN or other form of access code (password). The access code can be protected against unauthorized detection using commercially available software technology such as software smart cards from Arcot Systems, Inc., described in “Software Smart Cards via Cryptographic Camouflage,” Proceedings IEEE Symposium on Security and Privacy, May 1999, and in co-pending US patent application number 08/996,758, “Method and Apparatus for Secure Cryptographic Key Storage Certification and Use,” which is incorporated herein by reference.

The access code may also be protected against unauthorized detection (e.g., so-called “shoulder surfing”) using the technology described in co-pending US patent application number 09/249,043, “Method and Apparatus for Secure Entry of Access Codes in a Computer Environment,” which is incorporated herein by reference.

At step **230**, the issuer **110** binds the cardholder's public key and some or all of the cardholder's payment particulars in a digital certificate, typically by encrypting the cardholder's public key and particular identifying information provided by the cardholder. The encryption key used for encrypting the cardholder’s payment particulars – called the domain key – is typically shared between the issuer proxy **110** and the bridge computer **130**, and may be either a symmetric key or an asymmetric encryption key. In one embodiment, the domain key may be a public key associated with the bridge computer **130**, so that only the bridge computer **130** can decrypt the encrypted cardholder particulars (using a corresponding private key associated with the bridge computer **130**). In another embodiment, the domain key may be a symmetric encryption key that is shared by the issuer proxy **110** and the bridge computer **130**. In either case, the bridge computer will use the domain key (actually, its private key counterpart, if asymmetric; or the domain key itself, if symmetric) to verify the binding, as will be described later in the section entitled



"Authorization." After the issuer proxy **110** combines the cardholder's public key with some or all of the cardholder's payment information and digitally signs the combination to create a digital certificate for the cardholder, the digital certificate for the cardholder is loaded into an electronic payment card for the cardholder. Of course, those skilled in the art will realize that many other types of binding can be used including, without limitation, offloading the signing to a trusted third party, or receiving (rather than creating) the digital certificate from the user (although such binding is less secure).

At step **240**, the issuer **110** sends and the cardholder's computer **100** receives the cardholder's electronic payment card, e.g., a piece of software that is downloaded to the cardholder's computer **100**. The electronic payment card (typically stored in a software wallet) may be further protected against unauthorized access via a PIN (preferably different from the PIN associated with the cardholder's conventional payment card) or other form of user access code. The access code may be protected against unauthorized detection by the above-mentioned procedures used to protect the private key PIN. (Indeed, if the two PINs are the same, private key access for digitally signing and electronic payment card access for transaction execution could be accessed via a single protocol.) Setting the access code (PIN) for the electronic payment card is preferably done when the electronic payment card is being created by the issuer **110**, but can also be done separately, e.g., when the cardholder first accesses his electronic payment card on the cardholder computer **100**.

Alternatively, if the cardholder wishes to be able to perform electronic transactions from a variety of locations, the cardholder's private key and/or electronic payment card may be stored at a credential server and downloaded on the fly by a roaming cardholder using a shared secret or challenge-response protocol. In the latter case, commercially available software such as Arcot WebFort from Arcot Systems, Inc., described at <http://www.arcot.com/products.html> and in co-pending U.S. patent application number 09/196,430, "Method and Apparatus for Secure Distribution of Authentication Credentials to Roaming Users," which is hereby incorporated by reference, may be used to effect the roaming functionality.

One advantage of this enrollment process is that the issuer's participation can be passive, in that the issuer proxy **110** can be operated by any trusted financial institution

that participates in the electronic payment system, and is not necessarily the bank or financial institution that issued the conventional payment card to the cardholder. This is important because it suffices that one well-recognized financial institution participates in the system. Furthermore, even the participation of this financial institution can be limited to establishing the issuer proxy 110 on the network for self-service access by the cardholder, and does not require mailings to the cardholder, or other physical interaction with the cardholder.

FIG. 2A illustrates an exemplary electronic payment card created using the preferred embodiment of the invention, in which the card contains: (a) the cardholder's digital certificate, comprising the cardholder's payment particulars, and his public key, portions of which are encrypted under the domain key; and (b) the cardholder's private key.

#### Point-of-sale Transaction between a Cardholder and a Merchant on the Computer Network

A cardholder uses his computer 100 to shop at a merchant's website at merchant's computer 120. Referring now to FIG. 3, at step 300, when the cardholder decides what goods or services he wants to buy, the merchant presents the cardholder with an electronic sales draft.

At step 310, the cardholder elects to pay the sales draft using the cardholder's electronic payment card. At step 320, a representation of the cardholder's electronic payment card may be displayed on the cardholder's computer 100. If the cardholder chose to protect his electronic payment card with an access code, then at step 330 the cardholder unlocks and activates his electronic payment card. If the electronic payment card is protected with an access code, then the electronic payment card cannot be activated unless the correct access code is entered. The access code can be stored in a variety of locations including, without limitation, the cardholder's own memory, or a floppy disk, magnetic stripe card, smart card, or disk drive coupled to the cardholder's computer 100. At step 340, the cardholder's (activated) electronic payment card digitally signs the electronic sales draft that was presented to the cardholder in step 300 using the cardholder's private key. Optionally, the cardholder's electronic payment card can automatically fill in the information used by the sales draft. At step 350, the cardholder's computer 100 sends the

digitally signed sales draft and the cardholder's digital certificate to the merchant's computer **120**, where it is received by the merchant's computer **120**.

#### Authorization

Referring now to FIG. 4, at step **400**, the merchant's computer **120** sends, and the  
5 bridge computer **130** receives, an authorization request from the merchant (seller). The authorization request includes the electronic sales draft with the cardholder's (buyer's) electronic signature and the cardholder's digital certificate. As mentioned above, in one embodiment of the invention, the cardholder's digital certificate includes the cardholder's verification key (public key) and an encrypted version of the cardholder's PIN for his  
10 conventional payment card.

At step **410**, the bridge computer **130** uses the cardholder's verification key to confirm (verify) that the cardholder's electronic signature on the sales draft was authorized by the cardholder (buyer). If the electronic signature is confirmed, then at step **420** the bridge computer **130** extracts the encrypted version of the cardholder's PIN for his  
15 conventional payment card from the cardholder's digital certificate and decrypts the PIN using the private key associated with the domain key (if the PIN was asymmetrically encrypted) or the domain key itself (if the PIN was symmetrically encrypted). In this (or in some equivalent) fashion, the bridge computer **130** can verify the binding (of the payment particulars and the user's public key) that was performed by the issuer **110**. The  
20 bridge computer **130** uses the decrypted PIN to generate a conventional authorization request as is well-known to those skilled in the art of payment card transaction processing (see, e.g., Visa International Acquirer Services External Interface Specification, April 1 1999, EIS 1080 Version 5.8, available from Visa). The decrypted PIN may be re-encrypted with a key that is shared by the bridge computer **130** and the transaction processor  
25 at payment gateway **140**. Certain other particulars that are typically used for creating a conventional authorization request, such as the conventional payment card number, conventional credit card holder verification value 2 ("CVV2"), conventional cardholder name and address, or any other cardholder identification information, may also be extracted and decrypted from the cardholder's digital certificate.

Note that some types of conventional payment transactions do not necessarily use PINs, e.g., some conventional credit card transactions. For these transactions, after the bridge computer **130** verifies the cardholder's digital signature on the sales draft at step **410**, the bridge computer **130** generates a conventional authorization request at step **420** without performing the PIN extraction and PIN decryption steps.

At step **430**, the bridge computer **130** sends the conventional authorization request to the transaction processor at payment gateway **140**. Using the information provided in the authorization request, the payment gateway **140** approves or denies the request and sends its authorization response back to the bridge computer **130**.

In an alternative embodiment of the invention, the bridge computer **130** can be integrated into the payment gateway **140**. Indeed, any combination of issuer proxy **110**, bridge computer **130**, and/or payment gateway **140** can be integrated together.

The bridge computer **130** receives from the payment gateway **140** either an approval or a disapproval of the authorization request. In either event, at step **440**, the bridge computer **130** forwards the authorization response (approval or disapproval) to the merchant (seller) at the merchant's computer **120**.

If the cardholder is making a debit transaction, then at step **450** the merchant's computer **120** sends a confirmation to the payment gateway **140** via the bridge computer **130**.

One advantage of this authorization process is that there is minimal impact on the merchant. Another advantage is that the payment gateway **140** can interact with the legacy back-end systems **150** using conventional transaction processing methods. In other words, no changes are necessarily required to the back-end infrastructure.

In an alternate embodiment of the system, the bridge computer **130** can act in "stand-in" mode. Specifically, some financial institutions may choose not to receive the decrypted PIN from the cardholder's digital certificate, relying instead on the bridge computer's assertion that the cardholder's signature verified correctly. If the cardholder PIN was also verified at the issuer proxy **110** during enrollment, the risk of a fraudulent transaction may be deemed low. In such situations, the bridge computer **130** would assemble and transmit an authorization request without a PIN to the transaction processor at payment gateway **140**.

In yet another embodiment of the system, the merchant can store a copy of the digital signature of the cardholder along with the sales draft. The bridge computer 130 would process the transaction assuming that the digital signature of the cardholder is valid. In the event that the cardholder disputes the transaction, the merchant must present the  
5 stored copy of the sales draft and the cardholder's digital signature. The bridge computer 130 will verify the digital signature and, on the basis of the verification, determine whether the merchant should refund the amount of the transaction. An advantage of this embodiment is that the computational processing required at the bridge computer 130 is reduced. However, the merchant faces an increased risk of fraud.

10 In yet another embodiment of the system, a user who does not have a conventional credit or debit card (or who wants to get additional conventional payment cards), can be given the option of signing up for a conventional payment card during the electronic payment card enrollment process. The conventional payment card number that is given to this user can then be incorporated into the user's electronic payment card.

15 In yet another embodiment of the system, a user may choose to enroll his checking account to an electronic payment credential, rather than a debit or credit card. The user would identify himself via a variety of means at enrollment time, or may be given an activation code by his bank that he would use to identify himself for enrollment.

20 Although the preferred embodiments of this invention create an electronic payment card for conventional debit or credit cards or conventional checking accounts, the present invention enables a bridge to network payment for almost any conventional transaction system. For example, the present invention could also be used for secure electronic bill payment, person-to-person transactions, and electronic auction settlements.

25 The software described herein, for use by the various computers, is conveniently implemented using C, C++, Java, Javascript, HTML, or XML, running on Windows, Windows NT, Solaris, Unix, Linux, or Macintosh operating systems on virtually any computer platform. Moreover, those skilled in the art will readily appreciate that such software can be implemented using virtually any programming language, running on virtually any operating system on any computer platform.

30 The various embodiments described above should be considered as merely illustrative of the present invention. They are not intended to be exhaustive or to limit the

invention to the forms disclosed. Those skilled in the art will readily appreciate that still other variations and modifications may be practiced without departing from the general spirit of the invention set forth herein. Therefore, it is intended that the present invention be defined by the claims that follow.

What is claimed is:

1. A method for authenticating an electronic payment comprising:
  - (a) receiving from a seller an electronic sales draft including an electronic signature;
  - 5 (b) receiving from said seller a digital certificate associated with a buyer, said digital certificate including a verification key and an encrypted version of a personal identification number (PIN);
  - (c) using said verification key to verify that said electronic signature was authorized by said buyer;
  - 10 (d) extracting said encrypted version of said PIN from said digital certificate;
  - (e) decrypting said encrypted version of said PIN;
  - (f) generating, using said PIN, an authorization request;
  - (g) sending said authorization request for a PIN to a financial institution;
  - (h) receiving an approval of said authorization request from said financial  
15 institution; and
  - (i) sending said approval to said seller.
2. A method for authorizing an electronic purchase in a networked computer environment, comprising the steps of:
  - 20 (a) receiving, from a merchant, a transaction authorization request including a digital certificate passed through said merchant from a user involved in said transaction,
    - (i) said digital certificate including a financial account datum associated with said user, at least a portion of which is confidential from  
25 said merchant,
    - (ii) said digital certificate conveying a binding between at least a portion of said financial account datum and a public key of said user;
  - (b) verifying said binding using a cryptographic verification key associated with a trusted party performing said binding; and

(c) using said financial account datum to authorize a transaction order digitally signed by said user with a private key corresponding to said public key.

3. The method of claim 2 where said digital certificate constitutes said binding.

4. The method of claim 2 where said binding is embedded in said digital certificate.

5 5. The method of claim 2 where said financial account datum includes a credit card number.

6. The method of claim 2 where said financial account datum includes a debit card number.

7. The method of claim 2 where said financial account datum includes a PIN.

10 8. The method of claim 2 where said financial account datum includes a card verification value 2.

9. The method of claim 2 where said financial account datum includes checking account information.

15 10. The method of claim 2 where said binding is performed with a symmetric key shared between said trusted party and a party performing said verification step.

11. The method of claim 2 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.

12. The method of claim 2 where said binding is performed by an issuer of said digital certificate.



13. The method of claim 2 where said binding is performed by an issuer of said financial account datum.
14. The method of claim 2 where said digital certificate is protected with an access code known to said user.
- 5 15. A method for providing electronic payment capabilities to a user in a networked computer environment, comprising the steps of:
- (a) obtaining a financial account datum associated with said user;
  - (b) obtaining a public key associated with said user;
  - 10 (c) obtaining a cryptographically assured binding of said public key to at least a portion of said financial account datum,
    - (i) said binding being conveyed in a digital certificate for said user,
    - (ii) said digital certificate being usable by said user to conduct an electronic transaction involving said financial account datum; and
  - 15 (d) transmitting said digital certificate to said user, enabling said user to conduct said electronic transaction involving (i) a merchant from whom at least a portion of said financial account datum is kept confidential, and (ii) a transaction processor capable of verifying said binding using a cryptographic verification key associated with a trusted party performing said binding.
- 20 16. The method of claim 15 where said digital certificate constitutes said binding.
17. The method of claim 15 where said binding is embedded in said digital certificate.
18. The method of claim 15 where said financial account datum includes a credit card number.
- 25 19. The method of claim 15 where said financial account datum includes a debit card number.

20. The method of claim 15 where said financial account datum includes a PIN.
21. The method of claim 15 where said financial account datum includes a card verification value 2.
22. The method of claim 15 where said financial account datum includes checking account information.
23. The method of claim 15 where said binding is performed with a symmetric key shared between said trusted party and said transaction processor.
24. The method of claim 15 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.
25. The method of claim 15 where said binding is performed by an issuer of said digital certificate.
26. The method of claim 15 where said binding is performed by an issuer of said financial account information.
27. The method of claim 15 further comprising the step, after step (a), of verifying said financial account datum.
28. The method of claim 15 where said digital certificate is protected with an access code known to said user.
29. The method of claim 15 where said digital certificate is stored at a credential server accessible to said user.
30. An apparatus for authorizing an electronic purchase in a networked computer environment, comprising:

- (a) a computer processor;
  - (b) a memory connected to said processor storing a program to control the operation of said processor;
  - (c) the processor operable with said program in said memory to:
    - 5 (i) receive, from a merchant, a transaction authorization request, said request including a digital certificate passed through said merchant from a user involved in said transaction,
      - (1) said digital certificate including financial account datum associated with said user, at least a portion of which datum is confidential from said merchant,
      - 10 (2) said digital certificate conveying a binding between at least a portion of said financial account datum and a public key of said user;
    - (ii) verify said binding using a cryptographic verification key associated with a trusted party performing said binding; and
    - 15 (iii) use said financial account datum to authorize a transaction order digitally signed by said user with a private key corresponding to said public key.
31. The apparatus of claim 30 where said financial account datum includes a PIN.
- 20 32. The apparatus of claim 30 where said financial account datum includes a card verification value 2.
33. The apparatus of claim 30 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.
- 25 34. An apparatus for providing electronic payment capabilities to a user in a networked computer environment, comprising:
- (a) a processor;

- (b) a memory connected to said processor storing a program to control the operation of said processor;
- (c) the processor operable with said program in said memory to:
- (i) obtain a financial account datum regarding said user,
  - 5 (ii) obtain a public key associated with said user,
  - (iii) obtain a cryptographically assured binding of said public key to at least a portion of said financial account datum,
    - (1) said binding being conveyed in a digital certificate for said user,
    - 10 (2) said digital certificate being usable by said user to conduct an electronic transaction involving said financial account datum, and
  - (iv) transmit said digital certificate to said user, enabling said user to conduct said electronic transaction involving (1) a merchant from  
15 whom at least a portion of said financial account datum is kept confidential, and (2) a transaction processor capable of verifying said binding using a cryptographic verification key associated with a trusted party performing said binding.
35. The apparatus of claim 34 where said financial account datum includes a PIN.
- 20 36. The apparatus of claim 34 where said financial account datum includes a card verification value 2.
37. The apparatus of claim 34 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.
- 25 38. A computer-readable storage medium encoded with processing instructions for implementing a method for authorizing an electronic purchase in a networked computer environment, said processing instructions for directing a computer to perform the steps of:

- (a) receiving, from a merchant, a transaction authorization request, said request including a digital certificate passed through said merchant from a user involved in said transaction,
- 5 (i) said digital certificate including a financial account datum associated with said user, at least a portion of which datum is confidential from said merchant,
- (ii) said digital certificate conveying a binding between at least a portion of said financial account datum and a public key of said user;
- 10 (b) verifying said binding using a cryptographic verification key associated with a trusted party performing said binding; and
- (c) using said financial account datum to authorize a transaction order digitally signed by said user with a private key corresponding to said public key.
39. The computer-readable medium of claim 38 where said financial account datum includes a PIN.
- 15 40. The computer-readable medium of claim 38 where said financial account datum includes a card verification value 2.
41. The computer-readable medium of claim 38 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.
- 20 42. A computer-readable storage medium encoded with processing instructions for implementing a method for providing electronic payment capabilities to a user in a networked computer environment, said processing instructions for directing a computer to perform the steps of:
- (a) obtaining a financial account datum regarding said user;
- (b) obtaining a public key associated with said user;
- 25 (c) obtaining a cryptographically assured binding of said public key to at least a portion of said financial account datum,
- (i) said binding being conveyed in a digital certificate for said user,

- (ii) said digital certificate being usable by said user to conduct an electronic transaction involving said financial account datum; and
- (d) transmitting said digital certificate to said user, enabling said user to conduct said electronic transaction involving (i) a merchant from whom at least a portion of said financial account datum is kept confidential, and (ii) a transaction processor capable of verifying said binding using a cryptographic verification key associated with a trusted party performing the said binding.
43. The computer-readable medium of claim 42 where said financial account datum includes a PIN.
44. The computer-readable medium of claim 42 where said financial account datum includes a card verification value 2.
45. The computer-readable medium of claim 42 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.
46. A digital certificate for use in an electronic payment transaction in a networked computer environment, comprising:
- (a) a financial account datum associated with a user, at least a portion of which datum is confidential from a merchant involved in said payment transaction;
- (b) a cryptographically assured binding of a public key associated with said user to at least a portion of said financial account datum, said binding having been generated with a cryptographic verification key associated with a trusted party performing said binding;
- (c) said digital certificate configured for use by a transaction processor to:
- (i) verify said binding using a cryptographic verification key associated with said trusted party, and

- (ii) access said financial account datum to authorize a transaction order digitally signed with said user's private key corresponding to said public key.

5

- 47. The digital certificate of claim 46 where said digital certificate constitutes said binding.
- 48. The digital certificate of claim 46 where said binding is embedded in said digital certificate.
- 49. The digital certificate of claim 46 where said financial account datum includes a credit card number.
- 10 50. The digital certificate of claim 46 where said financial account datum includes a debit card number.
- 51. The digital certificate of claim 46 where said financial account datum includes a PIN.
- 15 52. The digital certificate of claim 46 where said financial account datum includes a card verification value 2.
- 53. The digital certificate of claim 46 where said financial account datum includes checking account information.
- 54. The digital certificate of claim 46 where said binding is performed with a symmetric key shared between said trusted party and said transaction processor.
- 20 55. The digital certificate of claim 46 where said binding is performed with an asymmetric key corresponding to said cryptographic verification key.

56. The digital certificate of claim 46 where said binding is performed by an issuer of said digital certificate.
57. The digital certificate of claim 46 where said binding is performed by an issuer of said financial account datum.
- 5 58. The digital certificate of claim 46 where said digital certificate is protected with an access code known to said user.



1/5

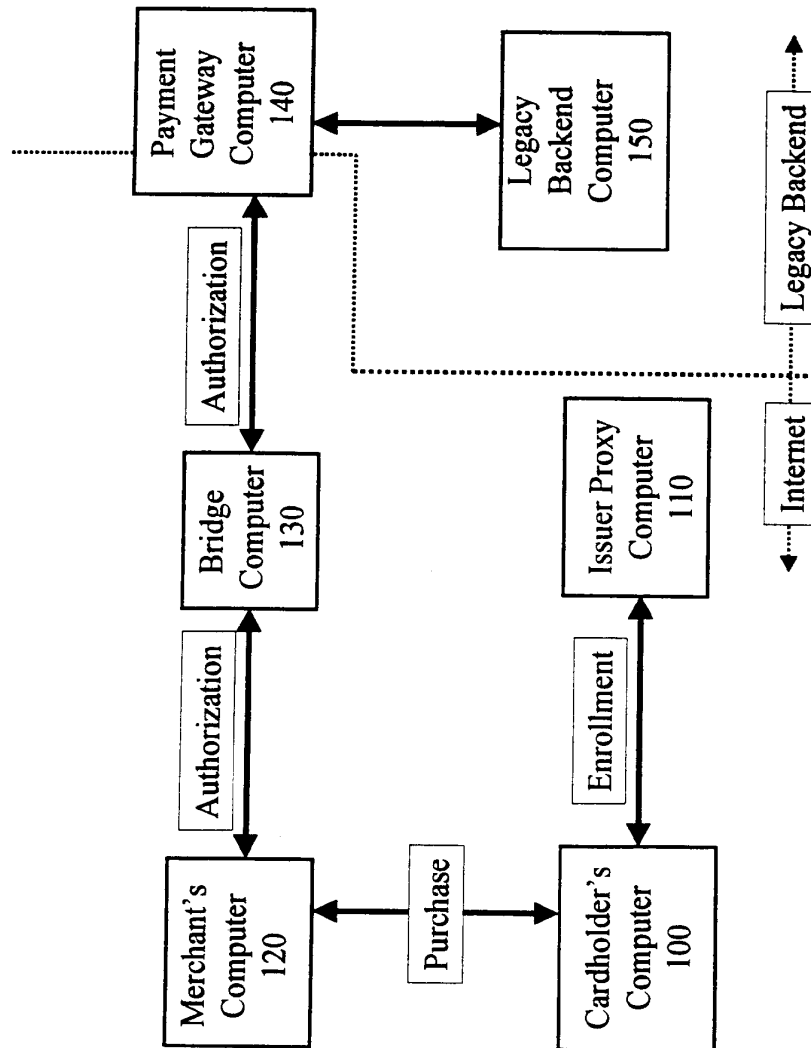


FIG. 1

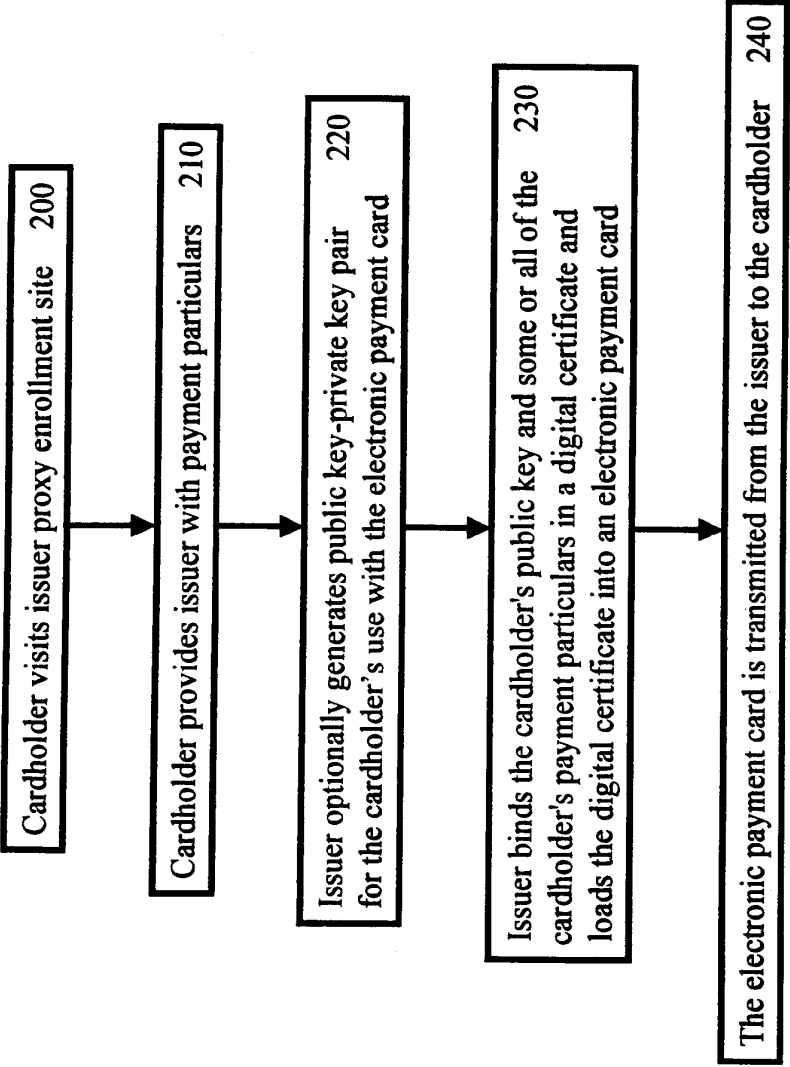


FIG. 2

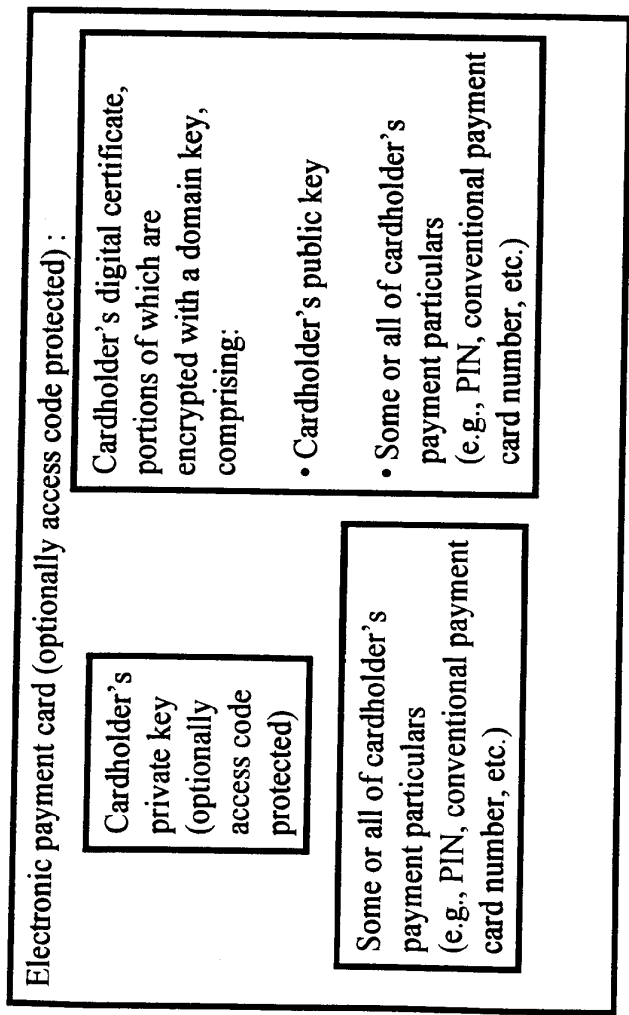


FIG. 2A

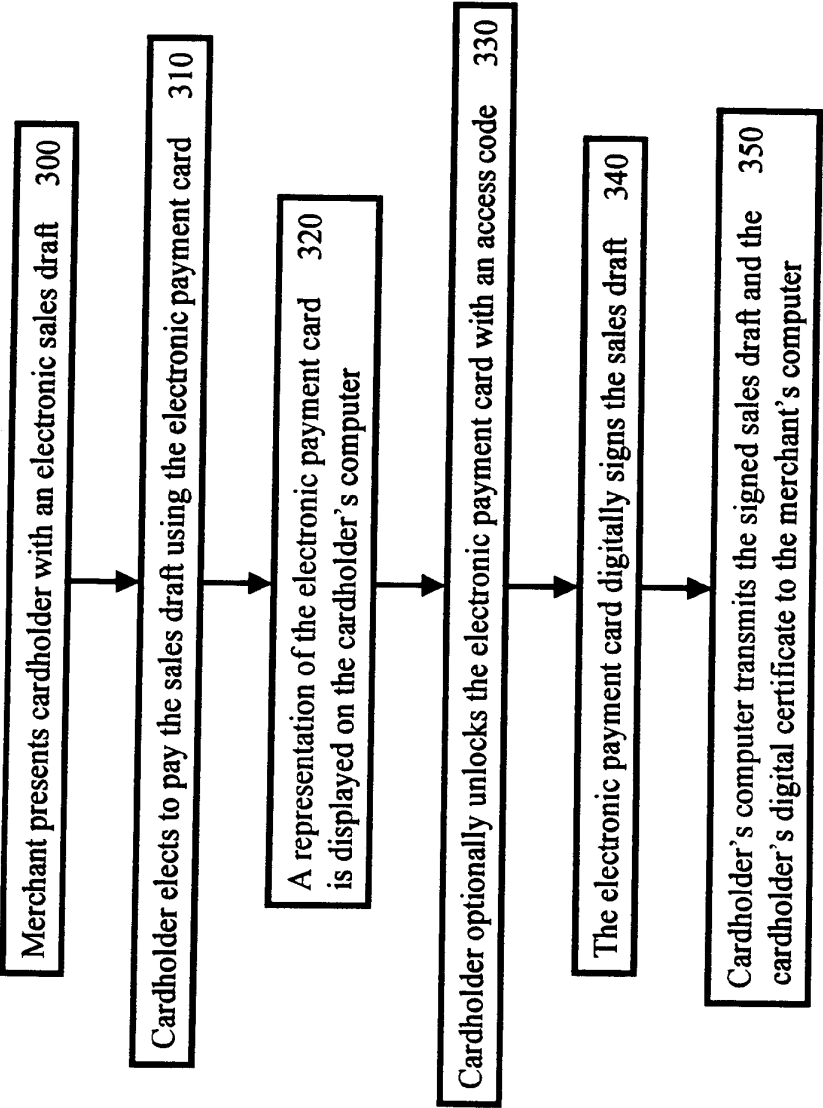


FIG. 3

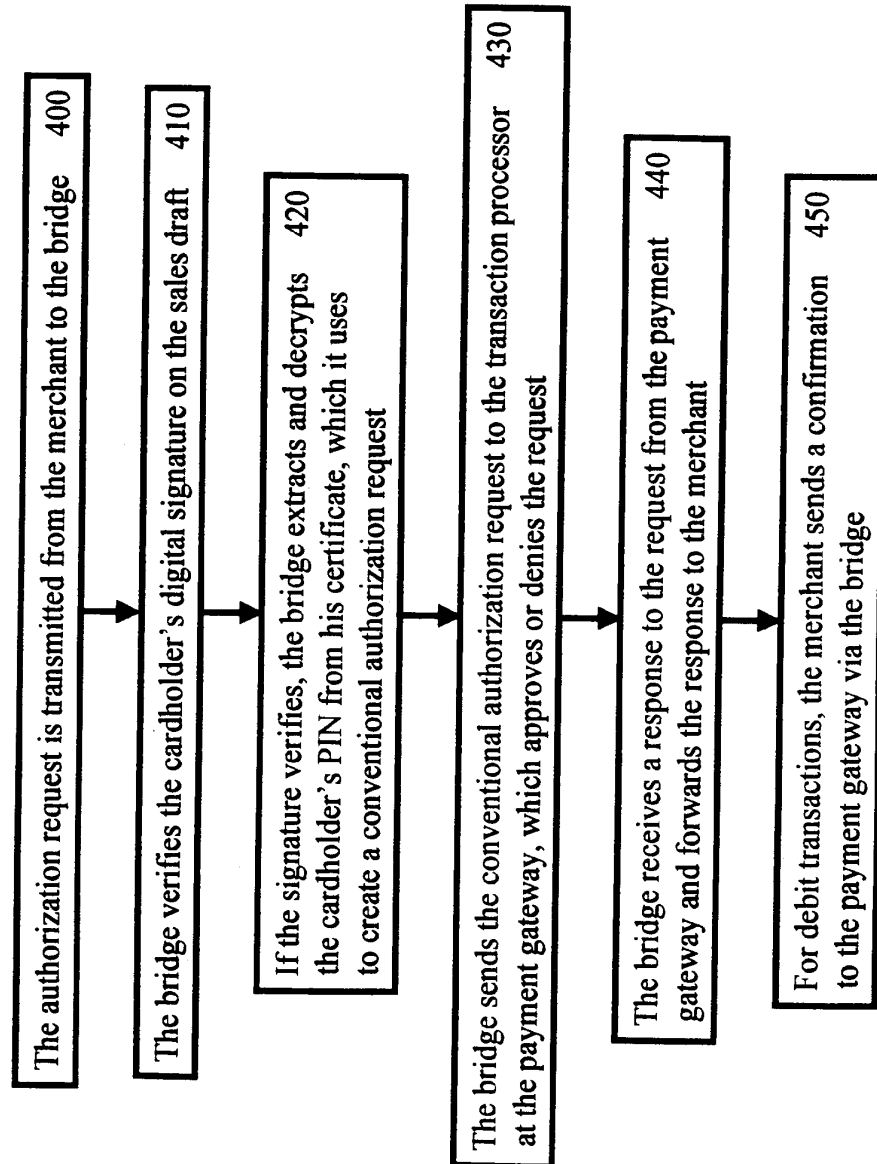


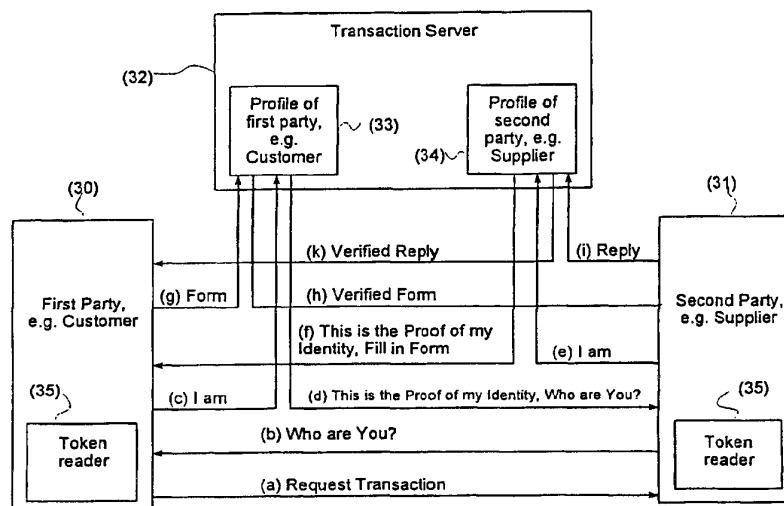
FIG. 4



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

(51) International Patent Classification <sup>7</sup> : <b>G06F 17/00</b>		<b>A2</b>	(11) International Publication Number: <b>WO 00/67143</b>
			(43) International Publication Date: 9 November 2000 (09.11.00)
(21) International Application Number: PCT/NL00/00278		(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 28 April 2000 (28.04.00)			
(30) Priority Data: 99201334.2 28 April 1999 (28.04.99) EP 09/543,602 5 April 2000 (05.04.00) US			
(71) Applicant (for all designated States except US): UNICATE B.V. [NL/NL]; Gooimeer 3-15, NL-1411 DC Naarden (NL).			
(72) Inventors; and			
(75) Inventors/Applicants (for US only): SIPMAN, Wilhelmus, Hendrikus, Maria [NL/NL]; Smidsweg 1A, NL-7433 BM Schalkhaar (NL). WARD, Scott, MacDonald [NL/NL]; Zuidlaan 31, NL-2111 GB Aerdenhout (NL).			
(74) Agent: H.V. MERTENS; Exter Polak & Charlouis B.V., P.O. Box 3241, NL-2280 GE Rijswijk (NL).			
		Published Without international search report and to be republished upon receipt of that report.	
		<b>Exhibit E13</b>  <b>EP 3 229 099 (17000713.2)</b>  BARDEHLE PAGENBERG Our ref.: S15482oEPT1EIN	

(54) Title: TRANSACTION METHOD AND SYSTEM FOR DATA NETWORKS, LIKE INTERNET



## (57) Abstract

A method and a system for performing a transaction between at least one first party and at least one second party are disclosed. A data network connects data input/output terminals of the parties. In the data network a secure and trusted transaction server is provided, in which a profile of the parties is registered, having a party identifier identifying a particular party, and authentication data for authenticating the party and data sent by the party. The parties communicate with each other through the transaction server by means of various transaction messages, which are digitally signed using a table of random numbers and a hashing operation, wherein the table of random numbers is generated by reading a token.

**FOR THE PURPOSES OF INFORMATION ONLY**

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

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

## TRANSACTION METHOD AND SYSTEM FOR DATA NETWORKS, LIKE INTERNET

## FIELD OF THE INVENTION

5

The present invention relates to a transaction system for use with data networks, like intranets, extranets, and Internet. Transactions to be performed may include e-commerce, such as shopping and business-to-business transactions, electronic banking, protected  
10 emailing, consulting and amending databases, brokerage, and the like. The invention also relates to the authentication of parties and the transactions exchanged. The invention further relates to the preparation of a confidential message. The invention still further relates to the generation of a digital signature as for use in the  
15 transmission of secure information.

## BACKGROUND OF THE INVENTION

When using data networks, like Internet, for transactions between  
20 parties, that want to exchange information, that want to buy goods, enjoy services or receive information and other parties offering those goods, services or information, one of the problems observed is the lack of secure and at the same time simple transaction methods. Secure systems, such as SET, SSL, etcetera exist, but are  
25 felt as being cumbersome to use, involving heavy message traffic, (complex) cryptographic procedures, and key management including key recovery.

## SUMMARY OF THE INVENTION

30

An object of the invention is to provide a secure data network transaction system without the need for cryptography, whilst fulfilling demands for security, particularly privacy, authentication and irrefutability of the messages that constitute a  
35 transaction.



The privacy requirements are that no information is disclosed to any party, unless needed to perform the basic objective, which within the scope of this invention comprises the execution of a transaction, with or without a payment that might result from the transaction.

The authentication requirements are to be able to verify the authenticity of transmitted data and the sender of those data. Within the scope of this invention these data, as well as an identification of the parties concerned, form part of transaction messages.

The basis of the invention is to use a secure and trusted computer environment, referred hereinafter as the transaction server. Parties wishing to use the services of the transaction server, are registered at the server with a so called profile. Such profile contains at least the data necessary to provide data integrity, data authentication, authentication of the parties, confidentiality of sensitive data (privacy) and irrefutability.

In such an environment, there will be two types of parties: a party that demands a service, a product, or information, further referred to as a "customer"; and a party that offers such services, products, or information, further referred to as a "supplier".

In case the transaction involves a payment, a third type of party may occur: one or more financial institutions that take care of the payment process, further referred to as a "financial institution". In the system according to the invention, the number of customers, suppliers and/or financial institutions may range from 1 to many. One or more of the objects of the invention are reached by a method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:

- (a) providing a transaction server in the at least one data network;
- (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;
- (c) the first party issuing at least one transaction message to the second party;
- (d) in response to the transaction message, the second party issuing a digitally signed transaction confirmation message to the first party;
- (e) on the basis of the transaction confirmation message, the first party issuing a digitally signed transaction approval message to the transaction server;
- (f) if required by the first or the second party, the transaction server verifying the authenticity of the first party and the second party, and the transaction data, in response to the transaction approval message;
- (g) the transaction server issuing a verified transaction approval message to the second party, in case said authenticity verification is required only if the verification is positive, resulting in a verified transaction approval message; and
- (h) fulfilment of the transaction.

An example of such a transaction is a customer to supplier transaction or a business to business transaction, where a customer or business demands a service, a product, or information.

The invention further discloses a method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:

- (a) providing a transaction server in the at least one data network;

- (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;
- (c) the first party issuing at least one transaction message to the transaction server;
- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party, and the transaction data;
- (e) if the verification is positive, the transaction server issuing a verified transaction message to the second party; and
- (f) the second party and the first party performing the at least one transaction through the transaction server by means of digitally signed messages, the validity of which is verified by the transaction server.

Examples of such a method are brokerage or telebanking, where the first party is a customer, and the second party is a supplier of services.

Use of the invention for sending and retrieving electronic mail is obtained by a method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:

- (a) providing a transaction server in the at least one data network;
- (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;

- (c) the first party issuing at least one digitally signed transaction message to the transaction server;
- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party;
- 5 (e) if the verification is positive, the transaction server linking the transaction message to the profile of a second party;
- (f) if the second party connects to the transaction server, the transaction server verifying the authenticity of the second party; and
- 10 (g) if the verification is positive, the transaction server issuing the transaction message to the second party.

Here, the transaction message is an email.

- 15 The invention further discloses a method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:
- 20 (a) providing a transaction server in the at least one data network;
- (b) in the transaction server, registering a profile of the at least one first party, the profile comprising a party identifier identifying the party, and authentication data for authenticating
- 25 the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation that is enabled for the party, and is meaningful only between the party and the transaction server;
- (c) the first party issuing at least one transaction message to the
- 30 transaction server;
- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party, and the transaction data;
- (e) if the verification is positive, the transaction server issuing
- 35 a verified transaction message to the second party; and
- (f) the second party and the first party performing the at least one transaction through the transaction server.

Such a method is particularly suitable for a party desiring to gain access to a service provider anonymously. Yet the service provider can be sure that the party is entitled to the requested access.

5

Instead of using a table of random data for verifying a digital signature, the profile may comprise operation identifiers each identifying an operation which is enabled for the party, and is meaningful only between the party and the transaction server.

10

In the system according to the invention, the need for encryption is eliminated as there are no sensitive data to be exchanged between customer, supplier and transaction server. The sensitive data are replaced by covert coded data, which are significant only to the sender and receiver of the message, thus respecting the privacy.

15

The authentication problems are solved by non-cryptographic methods, hence avoiding the problems of key management and the computing power to perform cryptographic operations. Even, when cryptographic digital signatures are used, the authenticity can be more easily established, using the transaction server concept according to the invention.

20

Additionally, transaction irrefutability is achieved, since every step in the process can be monitored and verified.

25

If the transaction involves payments, the transaction server does not require any adaptations within the existing payment systems, as it uses transparently the network structures, which are in use today for the processing of electronic payment transactions. This is quite a cost saving factor, as changing anything in these systems is not a trivial operation. At the same time, the use of the transaction server opens new channels for making frequent payments of small amounts of money, as the costs of processing these payments is low.

30

35

FURTHER ASPECTS OF THE INVENTION

Having the secure and trusted transaction server, this server may offer other services, like controlled access to services, payment methods based upon pre-authorisation or electronic cash, loyalty schemes, etc.

- 5 Loyalty schemes can easily be implemented by adding bonus counters to the customer's profile. With the transaction system according to the invention it is possible to add opportunities for 1-1 marketing between various suppliers (like merchants, issuer banks and acquirer banks) and customers. The loyalty scheme is based upon a data file,  
10 added to the customer's profile, which can be accessed e.g. as an Internet HTML page or similar.

- Furthermore, provisions may be made to enable a user to access the facilities, offered through the transaction system, from another  
15 device than his base device (i.e. the device which is standard configured for transactions according to the invention).

- The invention and its advantages will be more readily appreciated as the same becomes better understood by reference to the following  
20 detailed description and considered in connection with the accompanying drawings in which like reference symbols designate like parts.

#### BRIEF DESCRIPTION OF THE FIGURES

25

FIG. 1 is a diagram of a transaction system according to the present invention.

FIG. 2 illustrates how a party applies for participation in the transaction system.

- 30 FIG. 3 illustrates the relation between the various functional entities.

FIG. 4 illustrates the steps in a sample transaction.

FIG. 5 illustrates how a party can obtain a temporary profile where a party uses a device not personalised for him.

- 35 FIG. 6 illustrates the generation of a digital signature used for verifying the authenticity of the data and the sender of those data.

FIG. 7 illustrates how a random structure can be used to generate a table of random numbers.

FIG. 8 illustrates steps in a sample transaction between parties.

FIG. 9 illustrates a transaction, involving a direct payment.

- 5 FIG. 10 illustrates a supplier controlled environment, e.g. for electronic banking or brokerage, using the transaction server concept.

FIG. 11 illustrates steps in exchanging marketing information between parties.

- 10 FIG. 12 illustrates steps in accessing a service provider anonymously.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

- 15 FIG. 1 is a diagram of a transaction system in accordance with the invention. Centrally there is a secure and trusted data processing system, hereinafter to be referred to as a transaction server 1, to which participating parties 2, 3 are connected through a data network 4, e.g. Internet. If the transaction involves a payment, the transaction server 1 connects to one or more financial institutions 6, using the existing payment networks 5. The transaction server holds so-called profiles 7 and 8 for the participating parties, as explained below with reference to FIG. 2. "Users" of the transaction system are parties 2 ("Customer") that want to obtain goods, services and/or information from other parties 3 ("Supplier"), who are suppliers of these goods, services and/or information. The parties 2, 3 are selectively coupled to the data network 4.

- 30 Financial institutions 6 involved are Issuers (banks issuing payment cards to clients), Acquirers (banks serving the needs for suppliers of goods, services and/or information), and Schemes (financial institutions managing payment cards, like VISA, Mastercard and the like).

- 35 FIG. 2 illustrates how a party 12, 13 will be registered to be able to participate in the transaction system. In Step A the party 12, 13 fills in an application form, containing his request to participate

in the transaction system, his identification and (if payment is included in his profile) a list of payment methods (which cards, debit or credit, etc.) he wants to use (if he is a payer/consumer 12) or a list of payment accepting methods (which accounts, etc.) he wants to use (if he is a supplier 13) and an authorisation to the transaction server 11 to verify his account/card data at the corresponding bank and receive additional information, required to create the party's profile. The application form then is submitted to the institution, operating the transaction server 11, on paper, by E-mail, by courier, or otherwise.

Next, Step B, the account/card information together with the identification of the applying party 12, 13 is sent to a Registration Authority 10 (R.A.), e.g. his bank relation(s) to verify the authenticity of the party 12, 13 and the information supplied with his application.

The request to participate in the transaction system may also directly be addressed to the Registration Authority 10 (the dashed line in FIG. 2).

If the Registration Authority 10 of the party 12, 13 verifies the party positively, i.e. party's identity and other information match, it sends an approval message (Step C) to the organisation operating the transaction server 11. Using the approval message content of Step C in the transaction server 11 a profile for the party will be built, Step D.

Each profile contains at least the party's identification and those other data, which are required to process the transactions as being used.

The profile further may contain data, relevant to a personalised token, which has been or will be issued to the party.

Although cryptography is not required for the invention, still the profile may contain cryptographic keys, to enhance the security of the communication between the transaction server and the party.

The profile may contain a random data string, which acts as a pseudonym for the party identified. The profile further may contain a payment method list, indicating the various payment methods permitted for that party.



Each payment method in the profile may be identified by a random data string.

The profile may contain further data fields for other applications, e.g. electronic cash, loyalty programs, telebanking, stock  
5 brokerage, etc.

Finally, in Step E, the organisation operating the transaction server 11, which may also be a supplier or a bank, ensures that the party receives a personalised package, required to interface with the transaction system offered, and to be used with the party's data  
10 processing system to be coupled to the data networks. The content of this package (data, program and token) mirrors the profile, as registered in the transaction server. Parts of the package may be distributed to the party, using other channels, including electronic means, like electronic mail, or using other suppliers, e.g. a shop,  
15 the Registration Authority, or the like.

In FIG. 3 various entities are indicated.

Parties 21, 23 that want to make use of the transaction system may use a token reader 22, 24. This token reader is used to read the  
20 token, required for generating a digital signature of (selected parts of) message(s) in the transaction. Parties communicate through a data network 26, e.g. Internet. A secure and trusted transaction server 20 is also connected to the same data network, in order to receive messages from parties 21, 23. The transaction server 20  
25 processes the transactions, after thorough verification of the validity of the transactions, using the party's profiles, as stored in the transaction server 20. If the transaction involves a payment, the payment process (authorisation and settlement) is handled through the payment network 25, using the procedures that apply for  
30 such situations.

When the customer 2 in FIG. 1 wants to perform a transaction from another location than his base location (the location equipped with standard configured/personalized means for performing transactions  
35 according to the invention), or otherwise is requested to authenticate himself, this can be achieved by downloading a temporary set of data and program, as depicted in step E in FIG. 2

at the not at base location. A not at base location could be a workstation at the office, a Cybercafe, a PDA (Personal Digital Assistant) with mobile phone function attached, or otherwise, provided that such a system offers access to the data network and is  
5 equipped with a token read facility.

As soon as an authentication request is due and the not at base system detects an unknown token (i.e. a token, which is not registered with the token reader, data and program, as referred to  
10 in step E of FIG. 2), the user is prompted to enter data, containing the address of the transaction server, his remote ID and, if required, his password. If the token possesses a memory, these data may be obtained from that memory. A message is built, containing the user data entered and provided with a hash and (token based)  
15 signature, similar to the payment order as in (c) of FIG. 9, to be discussed below. The message is now transferred to the transaction server which, after verification of the correctness of the message received and the existence of a profile for that user, will send a temporary set of data and program to the not at base system. Now the  
20 not at base system is able to prepare a transaction message, or to authenticate the user, and proceed as normal. Depending of the agreed conditions, the temporary set of data and program will be erased or maintained for a set period.

25 FIG. 4 depicts a sample flow for a typical transaction.

- (a) A first party 30 and a second party 31 negotiate a transaction. In an Internet environment the first party, e.g. a customer will through his browser visit a site of the second party (e.g. a supplier) and request for a transaction, e.g. purchase goods,  
30 buy and sell stock, obtain information, or the like.
- (b) The second party may require some proof of identity of the first party. He therefore sends a "Who are You" message to the first party.
- (c) The first party adds his identity, as registered in his profile  
35 33 at the transaction server 32 to the "Who are You" message and completes the message with a unique transaction number, a Hash total calculated over the previous data items, and his

digital signature. This message ("I am") is then transferred to the transaction server.

- 5 (d) The transaction server verifies the message, received from the first party. If the message and the first party are verified positively, the server sends a message to the second party, confirming the identity of the first party and requesting the second party to identify himself.
- 10 (e) The second party adds to the message, received from the server, his identity, as registered in his profile 34, a "form" (e.g. a HTML page in case of Internet) in which the first party can fill in the details of his transaction request, a unique transaction number, a Hash total calculated over the previous data items, and his digital signature.
- 15 (f) Similar to step (d), the transaction server verifies the message received from the second party and his identity. If verified positively, the form is transferred to the first party.
- 20 (g) The first party now fills in the form with the transaction details, as requested, adds a unique transaction number, a Hash total and his digital signature, and sends it to the server.
- (h) The server verifies the message and, if verified positively, transfers the filled in form to the second party.
- 25 (i) The second party replies to the transaction details, as filled in in the form with a message, completed with a unique transaction number, a Hash total calculated over the previous data items, and his digital signature.
- (j) The server verifies the message received, and if verified positively, transfers the reply to the first party.

30 Steps (g) to (j) may be repeated, if more transactions are to be performed in the same session.

The verification of the messages as in (d), (f), (h) and (j) may be executed on the basis of a risk profile. If the risk profile does not demand immediate verification, the verification may be postponed until a request of either party is received to verify the messages, e.g. in case of a dispute about a certain transaction.

The creation of the digital signature may be performed as described in FIG. 9 below.

FIG. 5 depicts a sample flow how a party 40 can obtain a temporary profile, if he uses another system than the one, which has been personalised according to step E in FIG. 2.

5 Where in FIG. 4 step (c) should start, the system detects a token, for which the system has not been personalised.

(a) It therefore sends a request to the server 42 to receive a temporary profile. The message contains at least the server address  
10 and a user identification. These data are either collected from the first party's token, using a token reader 41 or entered manually. To enhance the security the message may be provided with a digital signature, generated in the same way as at the payment order. For further security the first party may have to include a Personal  
15 Identification Number (PIN) or any other personal identifier, e.g. a password or biometric quality.

(b) After positive verification of the request for a temporary profile, using the party's profile 43, the server returns a message,  
20 containing a temporary personalised package, which is in essence the same as in step E of FIG. 2, but may contain certain usage restrictions, like validity period or services granted. The validity period may be one transaction, a limited number of transactions, or an agreed time period, whatever has been agreed at the time the  
25 profile has been set up (step D in FIG. 2).

As an additional service, the server may log the full transaction details, to be collected by the first party at a later time, when he connects to the server from the system, which has initially been  
30 personalised for using the transaction method described.

#### POINTER METHOD

With the pointer method, as depicted in FIG. 6, it is possible to  
35 generate an electronic signature over a given set of data.

Assume a table 46 of random numbers. The table 46 may be the read result of a token, e.g. a 3DAS® card. Assume the number of elements in the table 46 equals n. Further assume that each number has a value between 0 and 255.

- 5 Assume the defined length of the signature is m bytes, where m is smaller than n.

In the example, depicted in FIG. 6, n equals 35 and m equals 5. Then an electronic signature over a given set of data D can be calculated:

- 10 compress D, using a hashing technique and/or some other method resulting into d;  
 d has the property that it can be split into at least n digits, each representing a value between 1 and m; preferably m is replaced by m', where m' is the largest power of 2 smaller than  
 15 m. In the example d becomes the string: 1, 9, 20, 16, 30;  
 identify the elements in the table of random numbers with the values 1 to m. In the example the elements are numbered column by column, starting at the top;  
 the digital signature now is the string of values of the  
 20 elements taken from the random table, identified by the values in d. In the example the digital signature becomes the string: 128, 27, 5, 99, 38.

- FIG. 7 illustrates how a table of random numbers may be generated,  
 25 using a material with a random structure.  
 Assume a token (e.g. a credit card), which is provided with a random two dimensional or three dimensional structure (e.g. a 3DAS® card). The token is inserted in a token reader 50. In the token reader 50 an image is taken from the random structure in the token, which is  
 30 in the case of a 3DAS® card an image as the sample image 51. In the image empty spaces are visible between the mapping of the random structure as blank areas. The areas are measured, and the n largest areas are selected (52). For each of the selected areas its Centre of Gravity (CoG) is calculated as a pair of coordinates (53). A  
 35 table 54 now is filled with the sizes (first column of the table) and the corresponding coordinates of the CoG (second and third

columns of the table). As the original material used is random, also the values in the table 54 are random.

Referring again to FIG. 1, the customer is indicated with 2 and the  
5 supplier with 3. The customer profile may contain, as earlier  
indicated, among others the account/card related data and data  
fields for other applications, e.g. electronic cash, loyalty  
programs, electronic banking, stock brokerage, etc. To enable 1-1  
marketing a further data file is added. It should be noted, that all  
10 the information, stored in the customer profile is present with the  
customer's agreement, as he did sign up for the transaction solution  
described. Upon given criteria, any of the parties 2, 3 and 6 may  
leave a message in the data file, destined for one of the other  
parties. This message is stored in the format, applicable for the  
15 network solution chosen, through which the customer communicates  
with the transaction server. In case of Internet this will be an  
HTML page or the like.

As soon as the destination party connects to the transaction server  
20 and the party's authenticity has been established, he will be  
prompted by the transaction server that a message is present for  
him. The party may neglect the prompt or decide to read the message.  
The message itself may link to other messages, stored elsewhere in  
the data network (e.g. Internet).

25  
E.g., if an acquirer wants to make an offer to the customer, the  
message may contain links to the acquirer's website, containing  
further information. As an example, an acquirer may offer extra  
bonus miles if the customer selects Mastercard as payment method for  
30 paying an international flight. The message then could be an HTML  
page: 'Dear customer, you may earn extra bonus miles if you pay your  
international flight, using your Mastercard. If you are interested,  
visit our website at [www.Acquirer.com](http://www.Acquirer.com)'. If the customer is  
interested, he just clicks the website's reference and can access  
35 all the relevant data on how to obtain the extra bonus miles and the  
conditions.

The criteria, used to add a message to the customer profile, may refer to general attributes, belonging to the profile, e.g. which payment methods are enabled for that customer, or refer to specific attributes, e.g. the current transaction, his collected bonus etc.

5

If a first party wishes to send a message to a second party based upon a given selection criterion, applicable to such second party, the first party sends the message together with the selection criterion to the transaction server. The transaction server now  
10 selects all second parties, that match the criterion and adds to their profiles a data file, containing the message concerned, or if such data file already exists, adds the message to that data file. The criterion may include a validity time, i.e. a time within which this criterion should be applied. In that case the transaction  
15 server will monitor changes in the second party's profile. If such a change might result in that the second party meets the criterion, the message is added to his profile as described above. As an example, an issuer may offer a special financing arrangement to any customer buying a car during a given month. The criterion is a  
20 customer buying a car. The validity time is the given month. As soon as the transaction server detects a transaction, in which a customer is buying a car (or might be expected to do so, based upon the supplier's profile, i.e. the supplier is a car seller), the transaction server adds the message to the customer's profile and  
25 prompts the customer that there is a message for him.

When applying this method the privacy of the customer is guaranteed: no information is disclosed to (in the example) the issuer, unless the customer decides to contact the issuer and apply for the special  
30 financing arrangement.

If no direct payment is involved between a first and a second party, FIG. 8 shows how the transaction will be protected, whereby the authentication of both parties, the transaction messages and the  
35 irrefutability of the transaction can be guaranteed, using the same mechanisms as described hereafter with reference to FIG. 9. The first party is the one, requesting a service. The first party may be

a consumer, but also a professional (as in business to business electronic commerce, B2B). The second party is the supplier of the requested services.

In step (50) the first party visits a site of the second party and indicates his desire to perform a certain kind of transaction. The second party sends a signed request for log-in to the first party (51), whereby his signature is generated similarly to the invoice message as in step (b) of FIG. 9 to be described hereafter.

The first party processes the log-in request (52) similar to preparing a payment order, as in step (c) of FIG. 9 and transmits the message to a secure transaction server.

The server verifies the message from step (52), using the profiles of both parties. If verification is positive, the server passes the log-in data to second party (53).

The second party checks the authorisation of the first party to perform the requested transaction and returns a signed application form (54) to the server. The application form is a data file (e.g. an HTML page) the first party can use to enter the transaction data. The server verifies the validity of the application form and transfers it to the first party (55).

The first party fills in the application form and signs it similar to the payment order message as in step (c) of FIG. 9. The form is then transmitted (56) to the server.

After verification by the server (57) the application form is passed to the second party, to be further processed.

Steps (54) to (57) may be repeated, depending on the context of the transaction.

Authenticity of the parties, the transaction and its irrefutability are thus guaranteed by the secure transaction server.

30

FIG. 9 depicts a sample flow for a typical transaction involving a payment.

(a) A first and a second party negotiate a purchase. In an Internet environment the first party, e.g. a consumer will through his browser visit a site of the second party (e.g. a merchant) and pursue a normal electronic shopping process by picking goods and storing them in a so-called shopping basket. When he has



completed the shopping process, he sends a buying order to the second party.

- (b) If the second party accepts the order, he will prepare an invoice message and transmit this to the first party.

5 The invoice message contains: an identification of the second party, a content of the sale, an amount due, a payment accepting method, a unique transaction number, a Hash total (HashM) calculated over the previous data items, and a digital signature of the second party, calculated over HashM, as exemplified below.

10 The identification of the second party may be replaced by his pseudonym, as registered at the transaction server, to enhance privacy as his true identity now is covert, without the need for cryptography.

15 The payment accepting method may be replaced by the random selected key, pointing to the required payment accepting method, as registered in the transaction server, to enhance privacy as his payment accepting method data now are covert, without the need for cryptography.

20 The digital signature may be generated using a token. The digital signature may be generated using the pointer method, as elucidated below. The token used in this case may be a 3DAS® object, such as a card, with an optically scannable three-dimensional pattern of randomly overlying fibres, as disclosed in U.S. Patent Nos. 5,354,097 and 5,719,939, and  
25 Dutch Patent No. 1,000,330, which documents are incorporated herein by reference. The card can be read with an appropriate reading device. This eliminates the need for cryptography.

30 If a 3DAS® card is used as token, any jitter in the reading of the mark of the card may be used to make the transaction number unique and guarantees at the same time, that the token has been read and processed.

- 35 (c) After receiving the invoice, the first party prepares a payment message and transmits this to the transaction server. This message contains: an identification of the first party, a copy

of the invoice message, a payment method, a unique transaction number, a Hash total (HashC) calculated over the previous data items, and a digital signature of the first party, calculated over HashC, as exemplified below.

5 The content of the sale may be deleted from the copy of the invoice message, enhancing privacy by not transmitting this information.

The HashC may be deleted from the payment message.

10 The identification of the first party may be replaced by his pseudonym, as registered at the transaction server, to enhance privacy as his true identity now is covert, without the need for cryptography.

15 The payment method may be replaced by a random selected key, pointing to the required payment method, as registered in the transaction server, to enhance privacy as his payment method data now are covert, without the need for cryptography.

The digital signature may be generated using a token.

20 The digital signature may be generated using the pointer method as elucidated below. The token used in this case may be a card like a 3DAS® card. This eliminates the need for cryptography. If a 3DAS® card is used as token, the jitter in the reading of the card may be used to make the transaction number unique and guarantees at the same time, that the token has been read and processed.

25 The digital signature also may include a personal identification, such as a Personal Identification Number (PIN) as is generally used when one wants to withdraw money through an Automated Teller Machine (ATM), a personal identification code or character string, a biometric feature, or any other  
30 feature which is strictly personal.

(d) After receiving the payment message, the transaction server starts verifying the message received. Depending on first party's payment method and second party's payment accepting method the authenticity of both parties are verified by the  
35 transaction server itself or by a financial institution concerned.

If 3DAS® cards are used as token, the transaction numbers, generated by either party will prove that an actual read has been performed, reducing the chance for counterfeit attacks, such as replay.

- 5 (e) Using the information in the first and second party's profiles the transaction server now builds an authorisation request message and/or settlement messages as agreed with the financial institution concerned. In the example of FIG. 9 an authorisation request message is build and transmitted to an
- 10 acquirer bank (a merchant's bank as indicated by his payment accepting method).
- (f) If the authorisation for the payment is granted, the financial institution concerned (in the example of FIG. 9 the acquirer bank) will prepare for settlement of the payment transaction.
- 15 (g) The financial institution concerned will inform the transaction server whether the payment is authorised.
- (h) Finally the transaction server informs both parties about the result of the authorisation.
- 20 These messages may contain a random number, which then is used to modify the party's pseudonym and the keys, identifying the payment methods. This will reduce the possibilities for possible attackers to replay a transaction or to act as impostor.

25 In FIG. 10 a configuration for telebanking or electronic banking is described. In this case there is only one party, the client of a bank. The party is via an open network, e.g. Internet, connected to the secure transaction server similar as the situation described in FIG. 1. Similarly the client's bank is connected to the server via

30 a closed network.

(60) When the client wants to perform one or more telebanking transactions, he sends a log-in request to the server, signed and protected in the same way as the payment order message (as (c) in FIG. 4).

35 (61) The server verifies the authenticity of the client and checks if a relation with the specified bank exists and/or the client is

allowed to perform a specific transaction. If so, the log-in request is (via the secure closed network) transmitted to the client's bank. The bank will respond with an Application Form (62), being a data file, where the client can enter the data concerning the transaction requested. This data file may be in case of Internet an HTML page. (63) The server signs the Application Form and transfers it to the client.

The client completes the Application Form with the transaction data required (64), applying the same protection and signature mechanisms, as used in the payment order as in (c) in FIG. 4 and sends it to the server.

After inspection of the signed Application Form, the server passes it to the client's bank (65).

If required, steps (62) till (65) are repeated, until all transactions are transmitted.

Authenticity of the party, the transaction(s) and the irrefutability are thus guaranteed by the secure transaction server.

Referring to FIG. 11, parties may exchange marketing information using the profiles (items 7, 8 in FIG. 1), where a party S is the sender and a party R is the receiver of the information. The identity of party R is hidden to party S, unless party R decides to directly contact party S to further investigate the information transmitted to party R. According to a step 70, the party S wants to submit a message to one or more parties R, based upon predefined criteria. For this purpose, party S therefore sets a criteria scheme. This scheme indicates on what basis the one or more parties R have to be addressed, and the time duration of the message to be submitted to the one or more parties R (i.e. the time period during which the message is valid and may be sent). Furthermore, party S defines the message contents. This message may contain a reference to a special offer and/or references to (in case of Internet) a website of party S.

According to a step 71, party S then sends the criteria scheme and the message to the server. According to a step 72, the server analyses the criteria scheme and selects those one or more parties R, that meet the criteria scheme. If a duration is specified, the

server monitors all transaction occurrences of the one or more parties R, to find out if they will fit into the criteria scheme, due to a change in their continuously updated profile. If a transaction occurrence of a party R is found, meeting the criteria  
5 scheme, the message is added to his profile together with a "flag", which is an indicator indicating that the criteria scheme is met. According to a step 73, as soon as an occurrence of a party R contacts the server, e.g. by issuing a payment order, and the "flag" is set, the server informs this party R that a message is present.  
10 The party R may decide to read the message or to ignore it. If the party R reads the message, he may react to the contents of the message or neglect it.

A party may want to access a certain service provider, without  
15 disclosing his true identity. The service provider, in turn, may or may not want to be sure that the party is entitled to the requested access. FIG. 12 shows how a party can get anonymous access to a service provider.

In step (80) the party sends a signed request to the secure  
20 transaction server to access a certain service provider. The server verifies the identity of the party and, if required, its right to access a requested service, using the party's profile. If the party is verified positively, the server transmits an access request message to the service provider, using a pseudonym for the party.  
25 The service provider now knows that the access request is legitimate, since the request originates from the transaction server and starts a dialogue (82) with the (anonymous) party through the server, which transfers the messages to the party (83). At the choice of the party, the party might disclose his true identity to  
30 the service provider during this dialogue.

#### EXAMPLES OF THE USE OF THE INVENTION

A consumer wants to buy a book via Internet at an Internet  
35 bookseller. He selects the book wanted from the inventory of the bookseller and places an order. The bookseller now prepares a digitally signed invoice message. In this message the bookseller's

identity is hidden by the reference to his profile at the transaction server, as is his payment accepting method. As the invoice message arrives at the computer of the consumer, he is prompted with a "pay" request. The consumer now inserts his token in  
5 the token reader and selects his method of payment. Next, the program in his computer prepares a payment message to the transaction server. In this message only the data from the invoice message, which are relevant for the payment process, are copied. The identity and selected method of payment are hidden by referring to  
10 his profile at the transaction server. The payment message finally is provided with the consumer's digital signature.

The transaction server now can process the payment request in basically three ways.

- 15 1. If the payment method selected is a cash payment, the account in the consumer's profile is debited for the amount due, while the account in the bookseller's profile is credited for the same amount. (Note: the cash method is more likely to be used for buying services, like searching a database, or information,  
20 like an audio file of the latest top hit).
2. If the payment method selected is a debit or credit transaction, and the transaction server is entitled by the financial institutions concerned to verify the authenticity of the consumer and the bookseller, then a simple authorisation  
25 request message or even a settlement request message will be sent to the corresponding financial institution.
3. If the payment method selected is a debit or credit transaction, and the transaction server is not entitled by the financial institutions concerned to verify the authenticity of  
30 the consumer and the bookseller, then an authorisation request message is sent to the corresponding financial institution, which then has to verify the authenticity of the consumer and bookseller, and inform the transaction server about the result of the authorisation.

35

Finally, the transaction server informs the consumer and retailer about the result of the payment order and optionally provides them with a random number, to modify the pointers to their profile data.

- 5 The privacy of the payment transaction in the above examples is guaranteed simply by not transmitting privacy sensitive information, but only references to data, already available at the transaction server.

10 The authentication of the payment transaction and its initiators is done by verifying the digital signatures.

Protection against hacking can be offered by changing the pointer values after each payment transaction, at fixed time intervals or after a given number of transactions.

- 15 The software required for implementing the method according to the invention, or one of the features thereof, is recorded on a computer readable medium, such as a floppy disk, a hard disk, a magnetic tape or other suitable media, to make a data processing system execute procedures in accordance with the method or one of the features  
20 thereof.

## CLAIMS

1. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:
- (a) providing a transaction server in the at least one data network;
  - 10 (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data  
15 for verifying a digital signature;
  - (c) the first party issuing at least one transaction message to the second party;
  - (d) in response to the transaction message, the second party issuing a digitally signed transaction confirmation message to the  
20 first party;
  - (e) on the basis of the transaction confirmation message, the first party issuing a digitally signed transaction approval message to the transaction server;
  - (f) if required by the first or the second party, the transaction  
25 server verifying the authenticity of the first party and the second party, and the transaction data, in response to the transaction approval message;
  - (g) the transaction server issuing a transaction approval message to the second party, in case said authenticity verification is  
30 required only if the verification is positive, resulting in a verified transaction approval message; and
  - (h) fulfilment of the transaction.
2. The method of claim 1, wherein the profile further comprises operation identifiers each identifying an operation which is enabled  
35 for the party, and is meaningful only between the party and the transaction server.



3. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:
- (a) providing a transaction server in the at least one data network;
  - (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation which is enabled for the party, and is meaningful only between the party and the transaction server;
  - (c) the first party issuing at least one transaction message to the second party;
  - (d) in response to the transaction message, the second party issuing a digitally signed transaction confirmation message to the first party;
  - (e) on the basis of the transaction confirmation message, the first party issuing a digitally signed transaction approval message to the transaction server;
  - (f) in response to the transaction approval message, the transaction server verifying the authenticity of the first party and the second party, and the transaction data;
  - (g) if the verification is positive, the transaction server issuing a verified transaction approval message to the second party; and
  - (h) fulfilment of the transaction.
4. The method of claim 1 or 3, wherein the profiles further comprise payment method identifiers identifying authorized methods of payment, the method further comprising before fulfilment of the transaction:
- (g1) the transaction server requesting an authorisation to at least one financial institution for authorising a payment from the first party to the second party;

(g2) in response to the authorisation request, the financial institution informing the transaction server whether the payment is authorised; and

(g3) the transaction server informing the first party and the second party about the result of the authorisation.

5. The method of claim 4, wherein step (b) comprises:

- (b1) each party providing payment information to the transaction server about each method of payment to be used in conjunction with the party;
- (b2) the transaction server verifying the payment information with at least one financial institution;
- (b3) the at least one financial institution providing the transaction server with transaction data necessary for the method of payment to be performed;
- (b4) the profiles further comprising the transaction data for each method of payment.

6. The method of claim 1 or 3, wherein the parties each establish an account held by the transaction server, the method further comprising before fulfilment of the transaction:

- (a) for settling a payment from the first party to the second party, the transaction server debiting the account of the first party, and crediting the account of the second party; and
- (b) the transaction server informing the first party and the second party about the result of the payment.

7. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:

- (a) providing a transaction server in the at least one data network;
- (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and

authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;

(c) the first party issuing at least one transaction message to the transaction server;

(d) in response to the transaction message, the transaction server verifying the authenticity of the first party, and the transaction data;

(e) if the verification is positive, the transaction server issuing a verified transaction message to the second party; and

(f) the second party and the first party performing the at least one transaction through the transaction server by means of digitally signed messages, the validity of which is verified by the transaction server.

15

8. The method of claim 7, wherein the profile further comprises operation identifiers each identifying an operation which is enabled for the party, and is meaningful only between the party and the transaction server.

20

9. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the

method comprising:

(a) providing a transaction server in the at least one data network;

(b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation that is enabled for the party, and is meaningful only between the party and the transaction server;

(c) the first party issuing at least one transaction message to the transaction server;

- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party, and the transaction data;
- (e) if the verification is positive, the transaction server issuing a verified transaction message to the second party; and
- (f) the second party and the first party performing the at least one transaction through the transaction server by means of digitally signed messages, the validity of which is verified by the transaction server.
10. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party, the method comprising:
- (a) providing a transaction server in the at least one data network;
- (b) in the transaction server, registering a profile of the at least one first party, the profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation that is enabled for the party, and is meaningful only between the party and the transaction server;
- (c) the first party issuing at least one transaction message to the transaction server;
- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party, and the transaction data;
- (e) if the verification is positive, the transaction server issuing a verified transaction message to the second party; and
- (f) the second party and the first party performing the at least one transaction through the transaction server.
11. A method for performing at least one transaction between at least one first party and at least one second party using at least one data network for interconnecting data input/output devices of

the at least one first party and the at least one second party, the method comprising:

- (a) providing a transaction server in the at least one data network;
- 5 (b) in the transaction server, registering a profile of the at least one first party and the at least one second party, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data  
10 for verifying a digital signature;
- (c) the first party issuing at least one digitally signed transaction message to the transaction server;
- (d) in response to the transaction message, the transaction server verifying the authenticity of the first party;
- 15 (e) if the verification is positive, the transaction server linking the transaction message to the profile of a second party;
- (f) if the second party connects to the transaction server, the transaction server verifying the authenticity of the second party; and
- 20 (g) if the verification is positive, the transaction server issuing the transaction message to the second party.

12. The method of claim 11, wherein, if the verification of the authenticity of the second party is positive, the transaction server  
25 issues a transaction confirmation message to the first party.

13. The method of claim 1, 3, 7, 9, 10 or 11, wherein the at least one first party and the at least one second party are provided with personalized means for assembling electronic messages, including  
30 transaction data associated thereto.

14. The method of claim 13, wherein for each party the means for assembling electronic messages are personalized by:

- (a) the party issuing a personalization message to the transaction  
35 server;
- (b) on the basis of the personalization message, the transaction server verifying the authenticity of the party; and

(c) if the verification is positive, the transaction server issuing a set of data and program to the party.

15. The method of claim 14, wherein the personalization message  
5 comprises:

- an address of the transaction server;
- the party identifier; and
- an optional password code.

10 16. The method of any of claims 1-4 and 7-11, wherein the identifiers are randomly selected.

17. The method of any of claims 1-4 and 7-11, wherein before the last step the transaction server modifies at least one of the  
15 identifiers, and subsequently informs each party concerned about the at least one modified identifier.

18. The method of claim 1, 3, 7, 9, 10 or 11, wherein at least one of the first party identifier and the second party identifier  
20 comprises a random data string.

19. The method of claim 1, 3, 7, 9, 10 or 11, wherein at least one message from a member of the group comprising the at least one first party and the at least one second party to any of the other members  
25 of said group is stored at the transaction server, and transmitted to said any of the other members of said group upon connection of said any of the other members of said group with the transaction server.

30 20. The method of claim 19, wherein the message is associated with at least one criterion, the method further comprising:

- (a) selecting any member of said group matching said at least one criterion, and
- (b) issuing said message to said selected member of said group.

35

21. The method of claim 19, wherein the message is associated with a message validity time period, the method further comprising:

- (a) selecting any member of said group connecting to the transaction server within the message validity time period, and
- (b) issuing said message to said selected member of said group.

5     22. The method of claim 19, wherein the group comprises a financial institution.

23. The method of claim 1, 3, 7, 9, 10 or 11, wherein at least one of the messages is signed by:

- 10    (a) subjecting message data to a hashing operation, resulting in a hashing code;
- (b) providing a digital signature on the basis of the hashing code; and
- (c) adding the digital signature to the message.

15

24. The method of claim 23, wherein the digital signature includes a personal identification.

20    25. The method of claim 23, wherein the digital signature is generated by:

- (a) providing a table of n random numbers, each having a predetermined position in the table;
- (b) dividing the hashing code into m digits, each representing a value between 1 and n, where m is smaller than n; and
- 25    (c) assembling a string of the random numbers of which the position in the table is indicated by the digits.

26. The method of claim 25, wherein the table of random numbers is generated by reading a token.

30

27. The method of claim 25 or 26, wherein the table of random numbers is generated by optically scanning geometrical configurations of a randomly shaped twodimensional or threedimensional mark.

35

28. A method for signing a message containing data, comprising:

- (a) subjecting the message data to a hashing operation resulting in a hashing code;
- (b) providing a digital signature on the basis of the hashing code; and
- 5 (c) adding the digital signature to the message.

29. The method of claim 28, wherein the digital signature is generated by:

- (a) providing a table of n random numbers, each having a predetermined position in the table;
- 10 (b) dividing the hashing code into m digits, each representing a value between 1 and m, where m is smaller than n; and
- (c) assembling a string of the random numbers of which the position in the table is indicated by the digits.

15

30. A method for generating a digital signature over a message, comprising:

- (a) providing a table of n random numbers, each having a predetermined position in the table;
- 20 (b) splitting the message into m digits, each representing a value between 1 and n, where m is smaller than n; and
- (c) assembling a string of the random numbers of which the position in the table is indicated by the digits.

25 31. A data processing system for performing at least one transaction between at least one first party and at least one second party, the system comprising:

- (a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second
- 30 party;
- (b) a transaction server in the at least one data network;
- (c) means for registering a profile of the at least one first party and the at least one second party in the transaction server, each profile comprising a party identifier identifying the party, and
- 35 authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;



(d) means for assembling electronic messages by the first party and the second party;

(e) means for issuing at least one transaction message by the first party to the second party;

5 (f) means for issuing a digitally signed transaction confirmation message by the second party to the first party in response to the transaction message;

(g) means for issuing a digitally signed transaction approval message by the first party to the transaction server on the basis of  
10 the transaction confirmation message;

(h) means for verifying the authenticity of the first party and the second party, and the transaction data by the transaction server in response to the transaction approval message;

(i) means for issuing a verified transaction approval message by  
15 the transaction server to the second party, if the verification is positive.

32. The system of claim 31, wherein the profile further comprises operation identifiers each identifying an operation which is enabled  
20 for the party, and is meaningful only between the party and the transaction server.

33. A data processing system for performing at least one transaction between at least one first party and at least one second  
25 party, the system comprising:

(a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party;

(b) a transaction server in the at least one data network;

30 (c) means for registering a profile of the at least one first party and the at least one second party in the transaction server, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each  
35 identifying an operation which is enabled for the party, and is meaningful only between the party and the transaction server;

(d) means for assembling electronic messages by the first party and the second party;

(e) means for issuing at least one transaction message by the first party to the second party;

5 (f) means for issuing a digitally signed transaction confirmation message by the second party to the first party in response to the transaction message;

(g) means for issuing a digitally signed transaction approval message by the first party to the transaction server on the basis of  
10 the transaction confirmation message;

(h) means for verifying the authenticity of the first party and the second party, and the transaction data by the transaction server in response to the transaction approval message;

(i) means for issuing a verified transaction approval message by  
15 the transaction server to the second party, if the verification is positive.

34. The system of claim 31 or 33, wherein the profiles further comprise payment method identifiers identifying authorized methods  
20 of payment, the system further comprising:

(a) means for requesting an authorisation by the transaction server to at least one financial institution for authorising a payment from the first party to the second party;

(b) means for informing the transaction server by the financial  
25 institution whether the payment is authorised, in response to the authorisation request; and

(c) means for informing the first party and the second party by the transaction server about the result of the authorisation.

30 35. The system of claim 34, wherein the profile registering means comprises:

(b1) means for providing payment information by each party to the transaction server about each method of payment to be used in conjunction with the party;

35 (b2) means for verifying the payment information by the transaction server with at least one financial institution;

(b3) means for providing the transaction server by the at least one financial institution with transaction data necessary for the method of payment to be performed, wherein the profiles further comprise the transaction data for each method of payment.

5

36. The system of claim 31 or 33, further comprising:

(a) means for establishing an account held by the transaction server by each party;

(b) means for debiting the account of the first party, and crediting  
10 the account of the second party by the transaction server, for settling a payment from the first party to the second party; and

(c) means for informing the first party and the second party by the transaction server about the result of the payment.

15 37. A data processing system for performing at least one transaction between at least one first party and at least one second party, the system comprising:

(a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second

20 party;

(b) a transaction server in the at least one data network;

(c) means for registering a profile of the at least one first party and the at least one second party in the transaction server, each profile comprising a party identifier identifying the party, and

25 authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data for verifying a digital signature;

(d) means for assembling electronic messages by the first party and the second party;

30 (e) means for issuing at least one transaction message by the first party to the transaction server;

(f) means for verifying the authenticity of the first party, and the transaction data by the transaction server in response to the transaction message;

35 (g) means for issuing a verified transaction message by the transaction server to the second party, if the verification is positive;

(h) means for performing the at least one transaction by the second party and the first party through the transaction server by means of digitally signed messages; and

(i) means for verifying the validity of the digitally signed  
5 messages by the transaction server.

38. The system of claim 37, wherein the profile further comprises operation identifiers each identifying an operation which is enabled for the party, and is meaningful only between the party and the  
10 transaction server.

39. A data processing system for performing at least one transaction between at least one first party and at least one second party, the system comprising:

15 (a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party;

(b) a transaction server in the at least one data network;

(c) means for registering a profile of the at least one first party  
20 and the at least one second party in the transaction server, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation that is enabled for the party, and is  
25 meaningful only between the party and the transaction server;

(d) means for assembling electronic messages by the first party and the second party;

(e) means for issuing at least one transaction message by the first party to the transaction server;

30 (f) means for verifying the authenticity of the first party, and the transaction data by the transaction server in response to the transaction message;

(g) means for issuing a verified transaction message by the transaction server to the second party, if the verification is  
35 positive;

(h) means for performing the at least one transaction by the second party and the first party through the transaction server by means of digitally signed messages; and

(i) means for verifying the validity of the digitally signed  
5 messages by the transaction server.

40. A data processing system for performing at least one transaction between at least one first party and at least one second party, the system comprising:

10 (a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party;

(b) a transaction server in the at least one data network;

(c) means for registering a profile of the at least one first party  
15 in the transaction server, the profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the profile further comprising operation identifiers each identifying an operation that is enabled for the party, and is meaningful only between the party and the  
20 transaction server;

(d) means for assembling electronic messages by the first party and the second party;

(e) means for issuing at least one transaction message by the first party to the transaction server;

25 (f) means for verifying the authenticity of the first party, and the transaction data by the transaction server in response to the transaction message;

(g) means for issuing a verified transaction message by the transaction server to the second party, if the verification is  
30 positive; and

(h) means for performing the at least one transaction by the second party and the first party through the transaction server.

41. A data processing system for performing at least one  
35 transaction between at least one first party and at least one second party, the system comprising:

- (a) at least one data network for interconnecting data input/output devices of the at least one first party and the at least one second party;
- (b) a transaction server in the at least one data network;
- 5 (c) means for registering a profile of the at least one first party and the at least one second party in the transaction server, each profile comprising a party identifier identifying the party, and authentication data for authenticating the party and data sent by the party, the authentication data comprising a table of random data
- 10 for verifying a digital signature;
- (d) means for assembling electronic messages by the first party and the second party;
- (e) means for issuing at least one digitally signed transaction message by the first party to the transaction server;
- 15 (f) means for verifying the authenticity of the first party by the transaction server in response to the transaction message;
- (g) means for linking the transaction message by the transaction server to the profile of a second party, if the verification is positive;
- 20 (h) means for verifying the authenticity of the second party by the transaction server, if the second party connects to the transaction server; and
- (i) means for issuing the transaction message by the transaction server to the second party, if the verification is positive.
- 25
42. The system of claim 41, further comprising means for issuing a transaction confirmation message by the transaction server to the first party, if the verification of the authenticity of the second party is positive.
- 30
43. The system of any of claims 31-34 and 37-41, comprising means for randomly selecting the identifiers.
44. The system of any of claims 31-34 and 37-41, further
- 35 comprising:
- (a) means for modifying at least one of the identifiers by the transaction server; and

(b) means for informing each party concerned about the at least one modified identifier.

45. The system of claim 31, 33, 37, 39, 40 or 41, wherein at least  
5 one of the first party identifier and the second party identifier  
comprises a random data string.

46. The system of claim 31, 33, 37, 39, 40 or 41, wherein the  
transaction server comprises a server message section for storing at  
10 least one message from a member of the group consisting of the at  
least one first party and the at least one second party to any of  
the other members of said group.

47. The system of claim 46, wherein the public data network has a  
15 network message section stored on it, and the message contains a  
link to said network message section.

48. The system of claim 46, wherein the message is associated with  
at least one criterion, the system further comprising:  
20 (a) means for selecting any member of said group matching said at  
least one criterion; and  
(b) means for issuing said message to said selected member of said  
group.

25 49. The system of claim 46, wherein the message is associated with  
a message validity time period, the system further comprising:  
(a) means for selecting any member of said group connecting to the  
transaction server within the message validity time period; and  
(b) means for issuing said message to said selected member of said  
30 group.

50. The system of claim 31, 33, 37, 39, 40 or 41, wherein the means  
for assembling electronic messages comprise:  
(a) means for issuing a personalization message by the first party  
35 and the second party to the transaction server;

(b) means for verifying the authenticity of the first party and the second party by the transaction server on the basis of the personalization message from the first party and the second party;

(c) means for issuing a set of data and program by the transaction  
5 server to the first party and the second party.

51. The system of claim 31, 33, 37, 39, 40 or 41, comprising means for signing at least one message by:

(a) subjecting message data to a hashing operation, resulting in a  
10 hashing code;

(b) providing a digital signature on the basis of the hashing code;  
and

(c) adding the digital signature to the message.

15 52. The system of claim 51, comprising means for generating the digital signature by:

(a) providing a table of n random numbers, each having a predetermined position in the table;

(b) dividing the hashing code into m digits, each representing a  
20 value between 1 and n, where m is smaller than n; and

(c) assembling a string of the random numbers of which the position in the table is indicated by the digits.

53. The system of claim 52, comprising a token and a token reader  
25 for generating the table of random numbers.

54. The system of claim 53, wherein the token is an object provided with an optically scannable randomly shaped twodimensional of threedimensional mark.

30

55. A data processing system for signing an message containing data, comprising:

(a) means for subjecting the message data to a hashing operation resulting in a hashing code;

35 (b) means for providing a digital signature on the basis of the hashing code; and

(c) means for adding the digital signature to the message.



56. The system of claim 55, comprising means for generating the digital signature having:

- (a) means for providing a table of n random numbers, each having a predetermined position in the table;
- (b) means for dividing the hashing code into m digits, each representing a value between 1 and m, where m is smaller than n; and
- (c) means for assembling a string of the random numbers of which the position in the table is indicated by the digits.

10

57. A data processing system for generating the digital signature, comprising:

- (a) means for providing a table of n random numbers, each having a predetermined position in the table;
- (b) means for dividing the hashing code into m digits, each representing a value between 1 and m, where m is smaller than n; and
- (c) means for assembling a string of the random numbers of which the position in the table is indicated by the digits.

58. A computer program product comprising at least one computer readable medium, having thereon computer program code means, when said program is loaded, to make a data processing system execute procedure according to claim 1, 3, 7, 9, 10 or 11.

59. A computer program element comprising computer program code means to make a data processing system execute procedure according to claim 1, 3, 7, 9, 10 or 11.

60. The computer program element of claim 59, embodied on a computer readable medium.

61. A computer readable medium having a program recorded thereon, where the program is to make a data processing system execute procedure according to claim 1, 3, 7, 9, 10 or 11.

35

FIG. 1

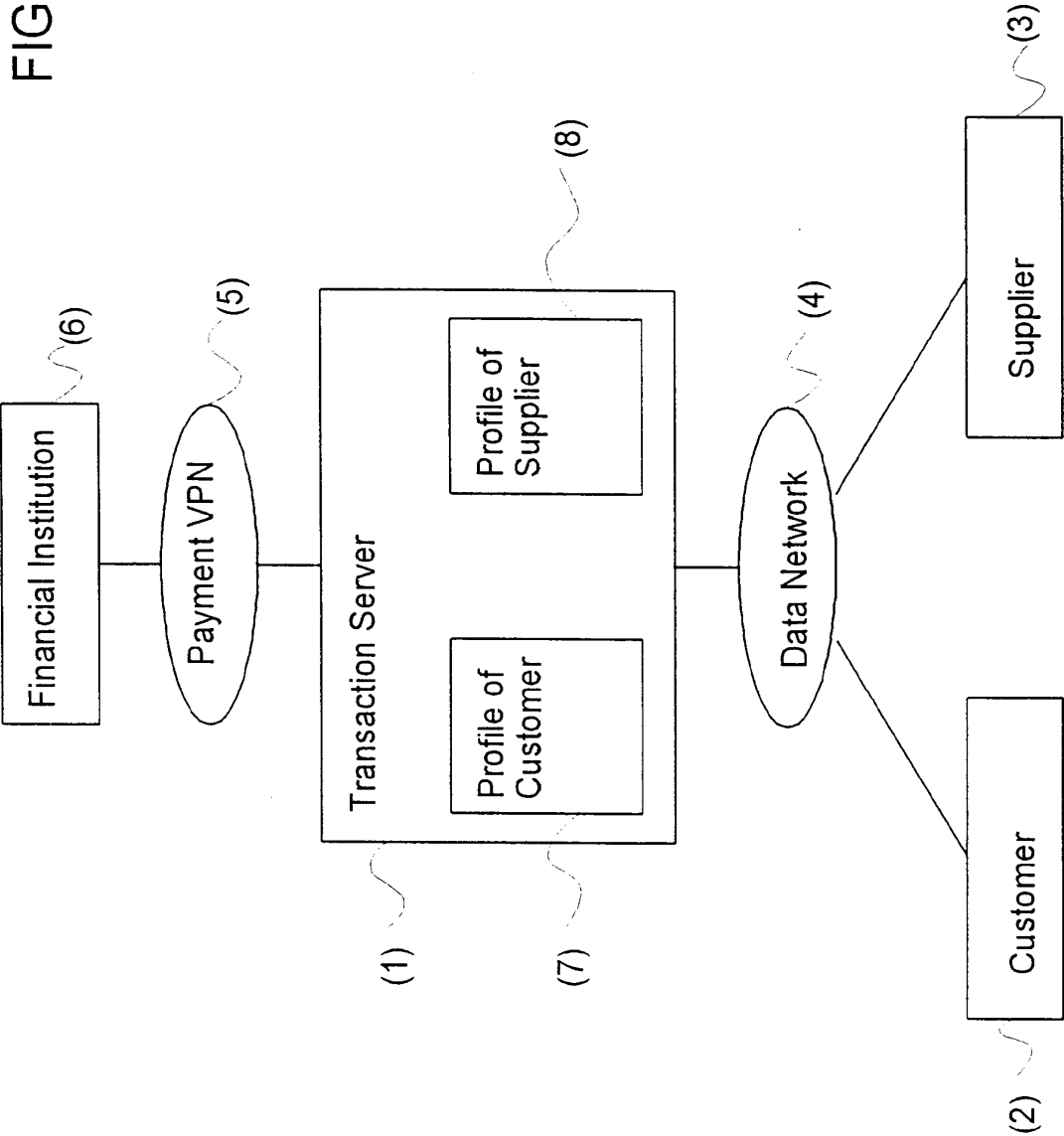


FIG. 2

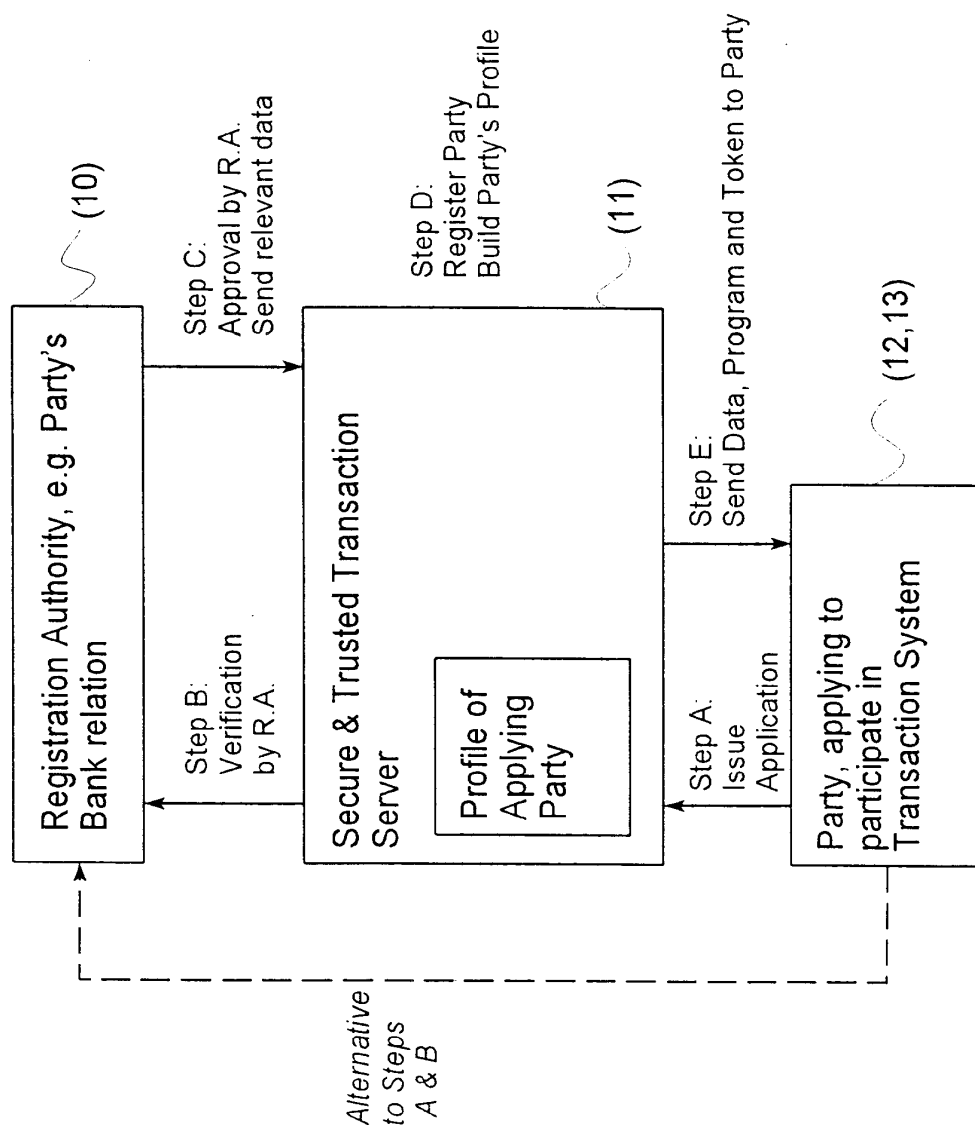


FIG. 3

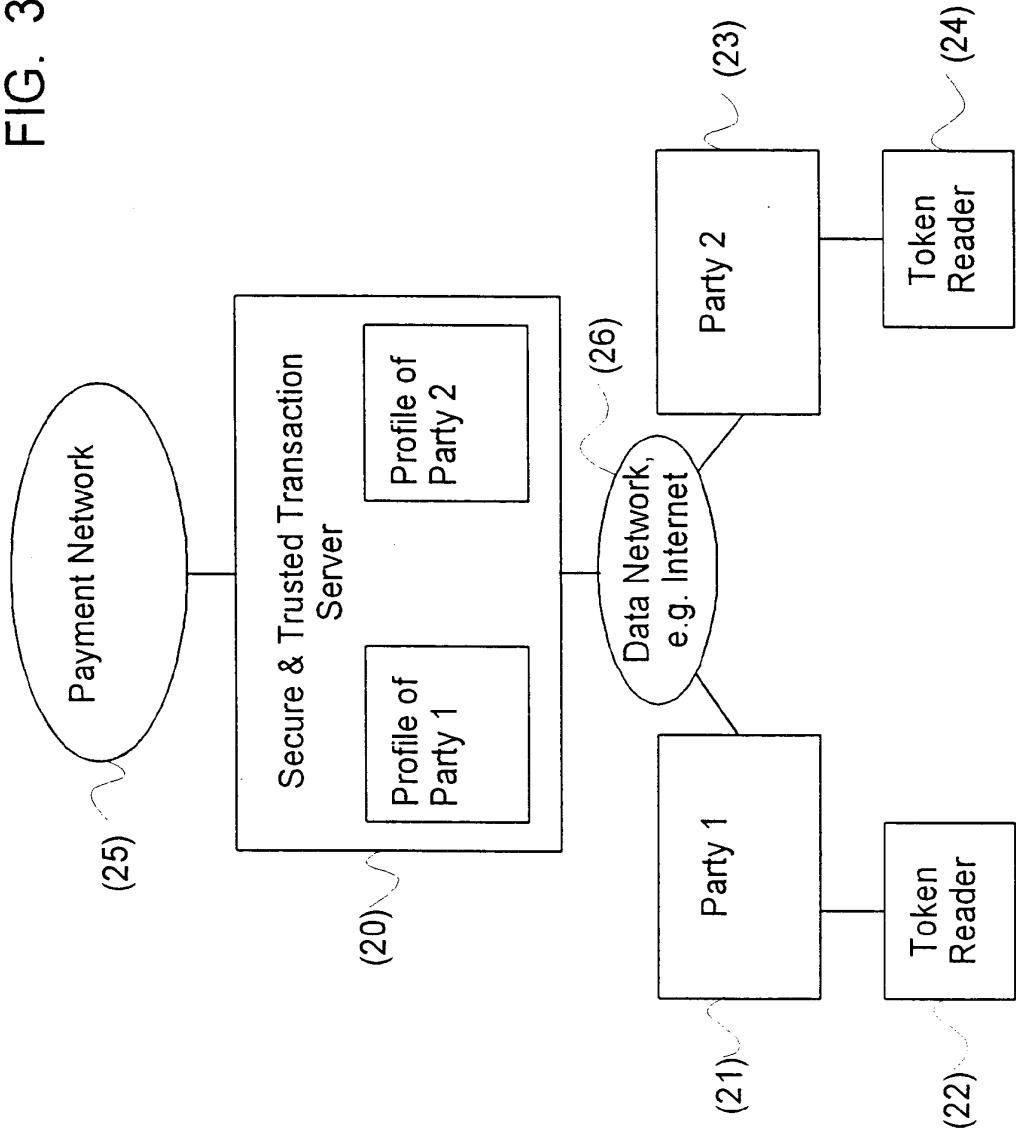


FIG. 4

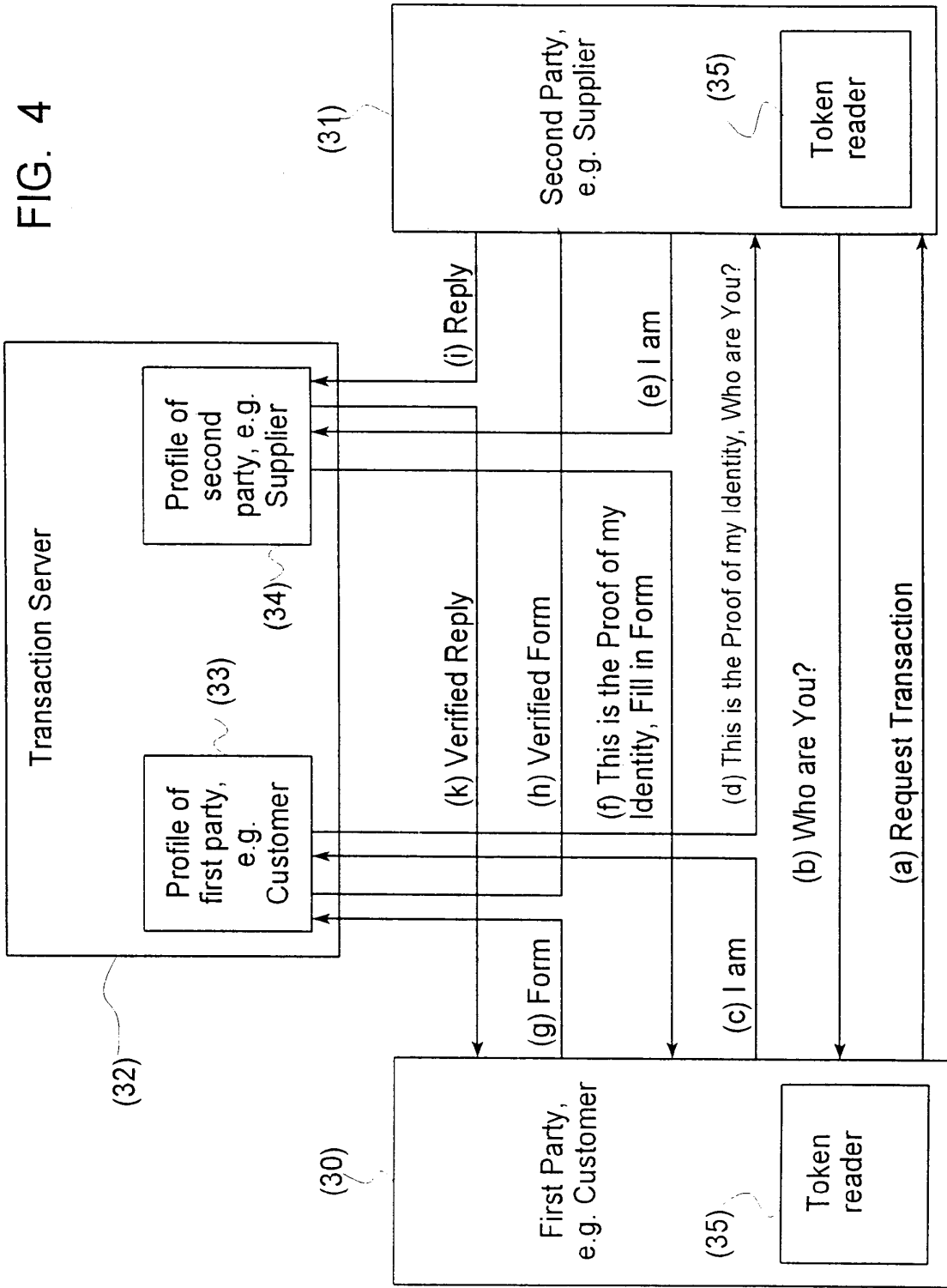
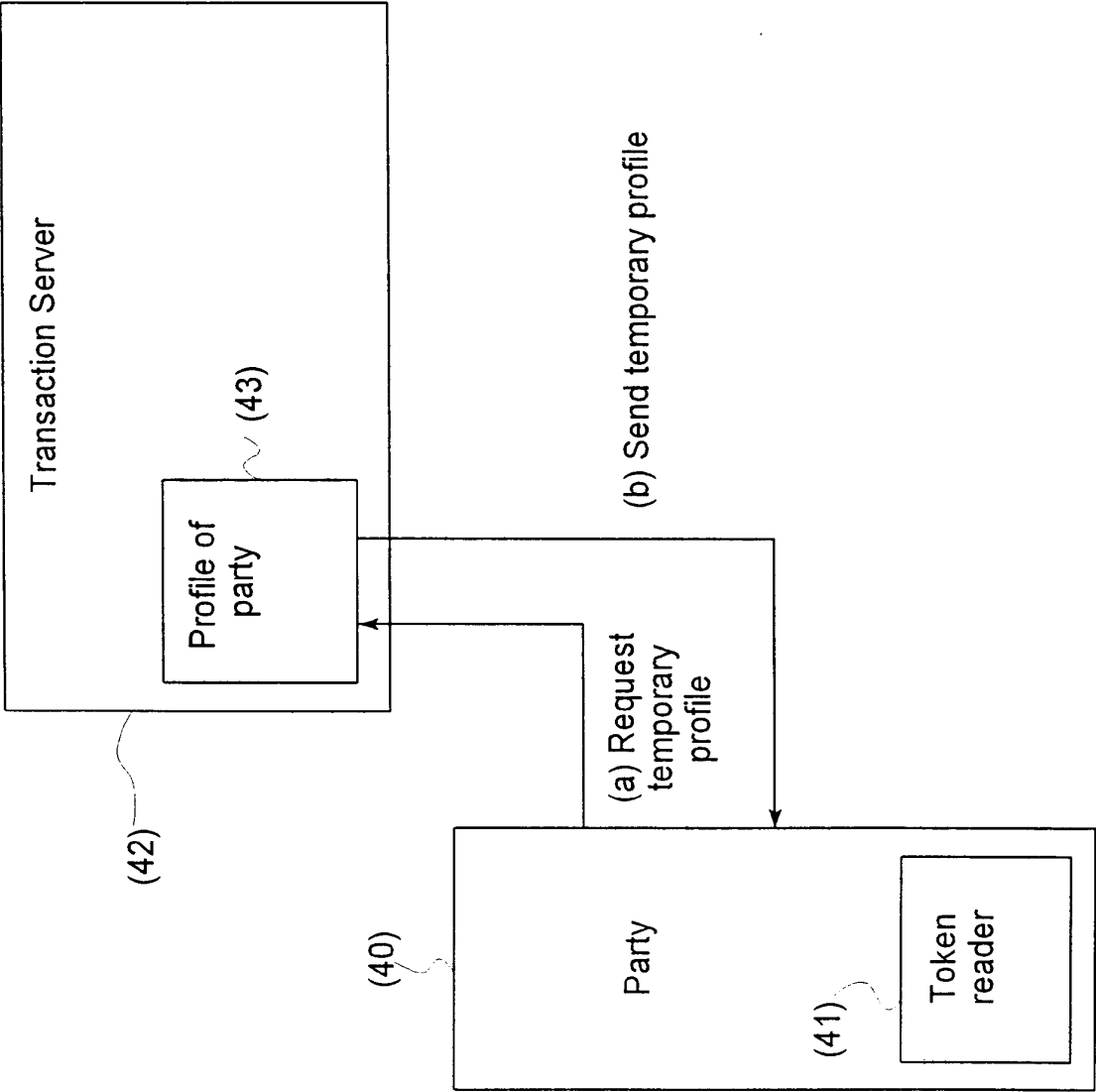


FIG. 5



Pointer Method  
FIG. 6

Table with Random Numbers

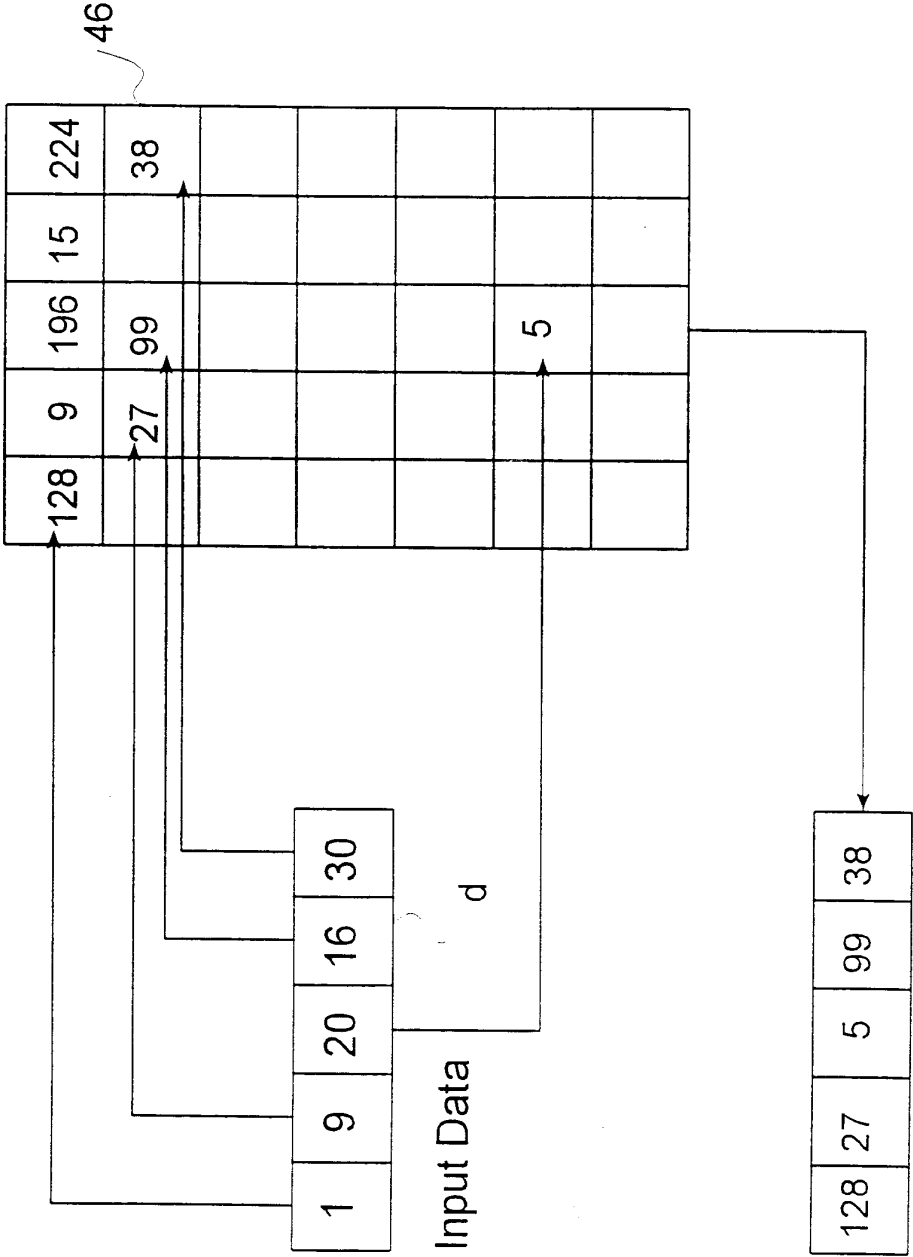


FIG. 7

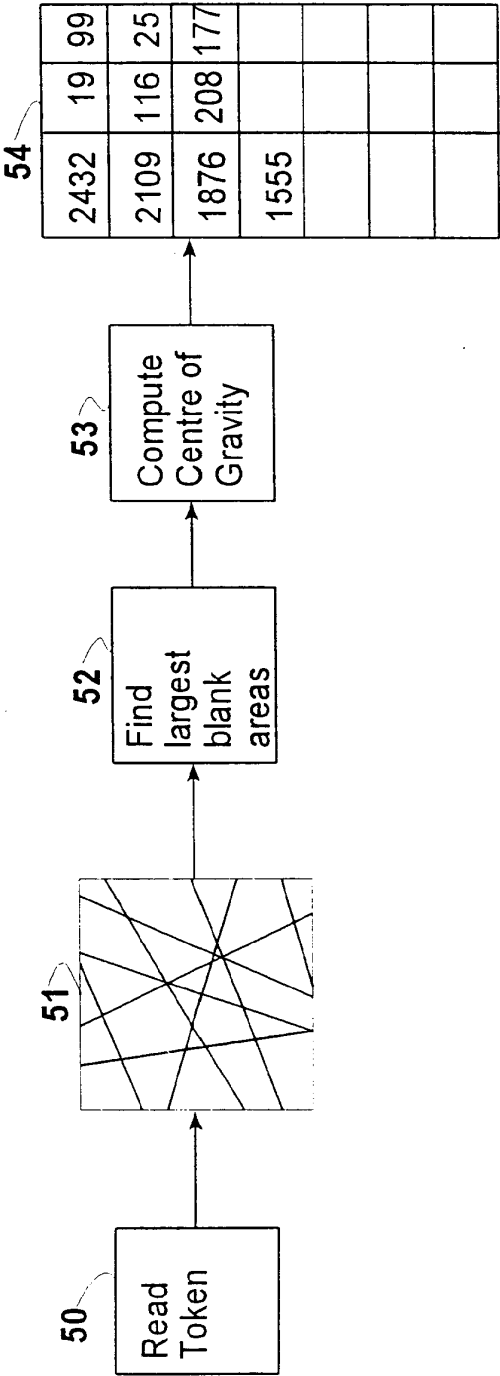
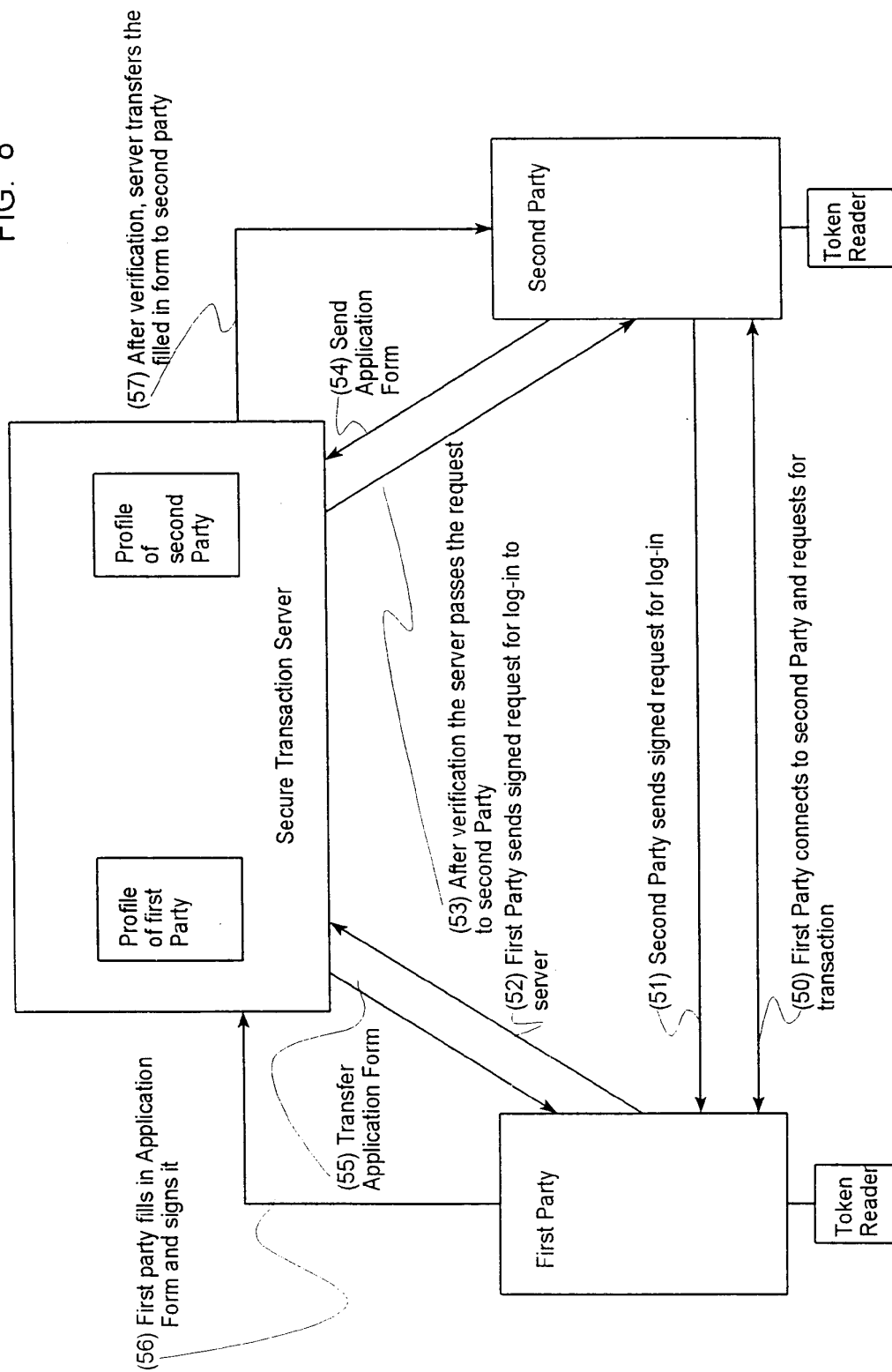




FIG. 8



9/12

FIG. 9  
Sample Transaction Flow

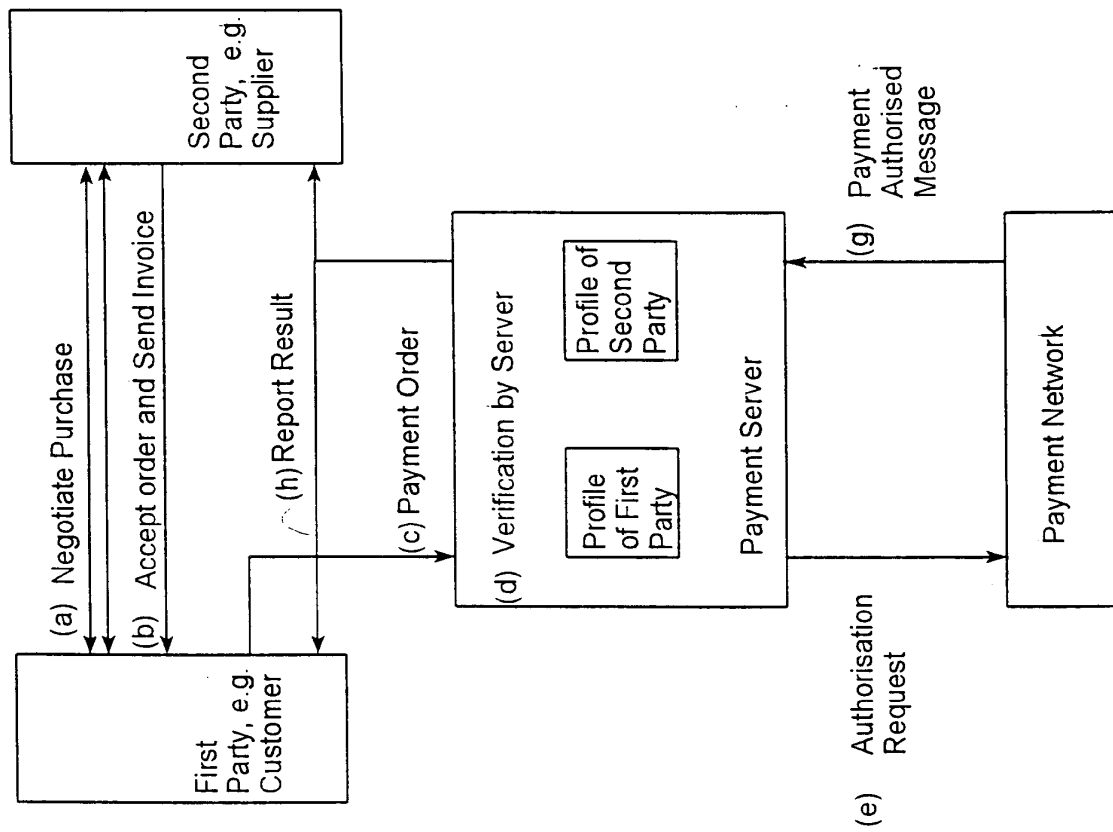


FIG. 10

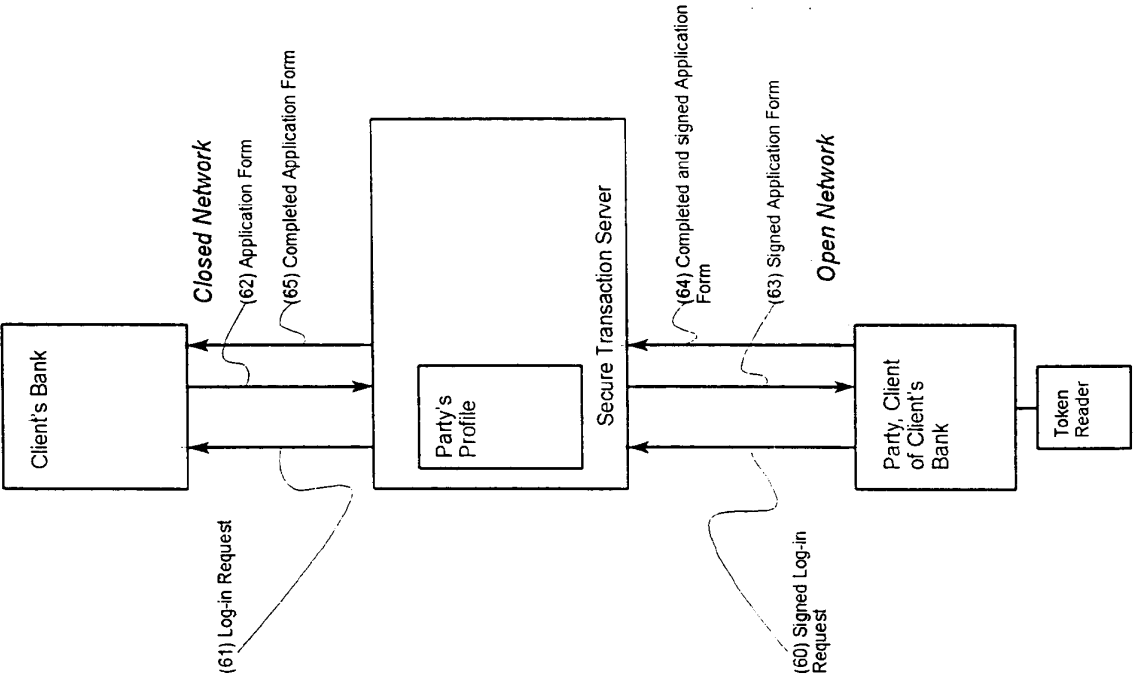


FIG. 11

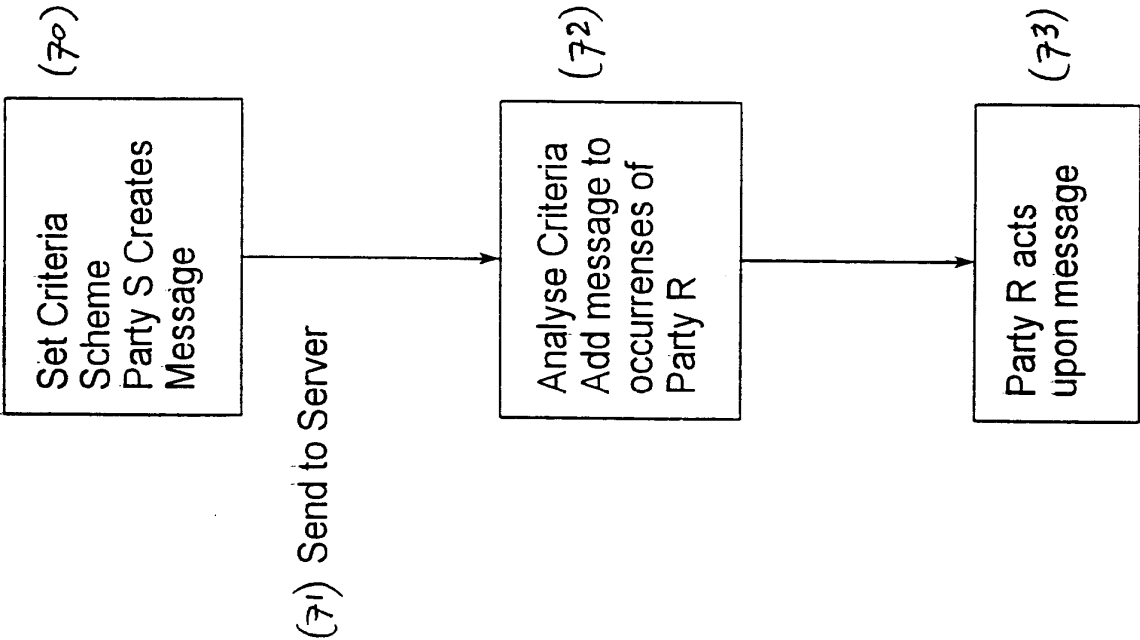
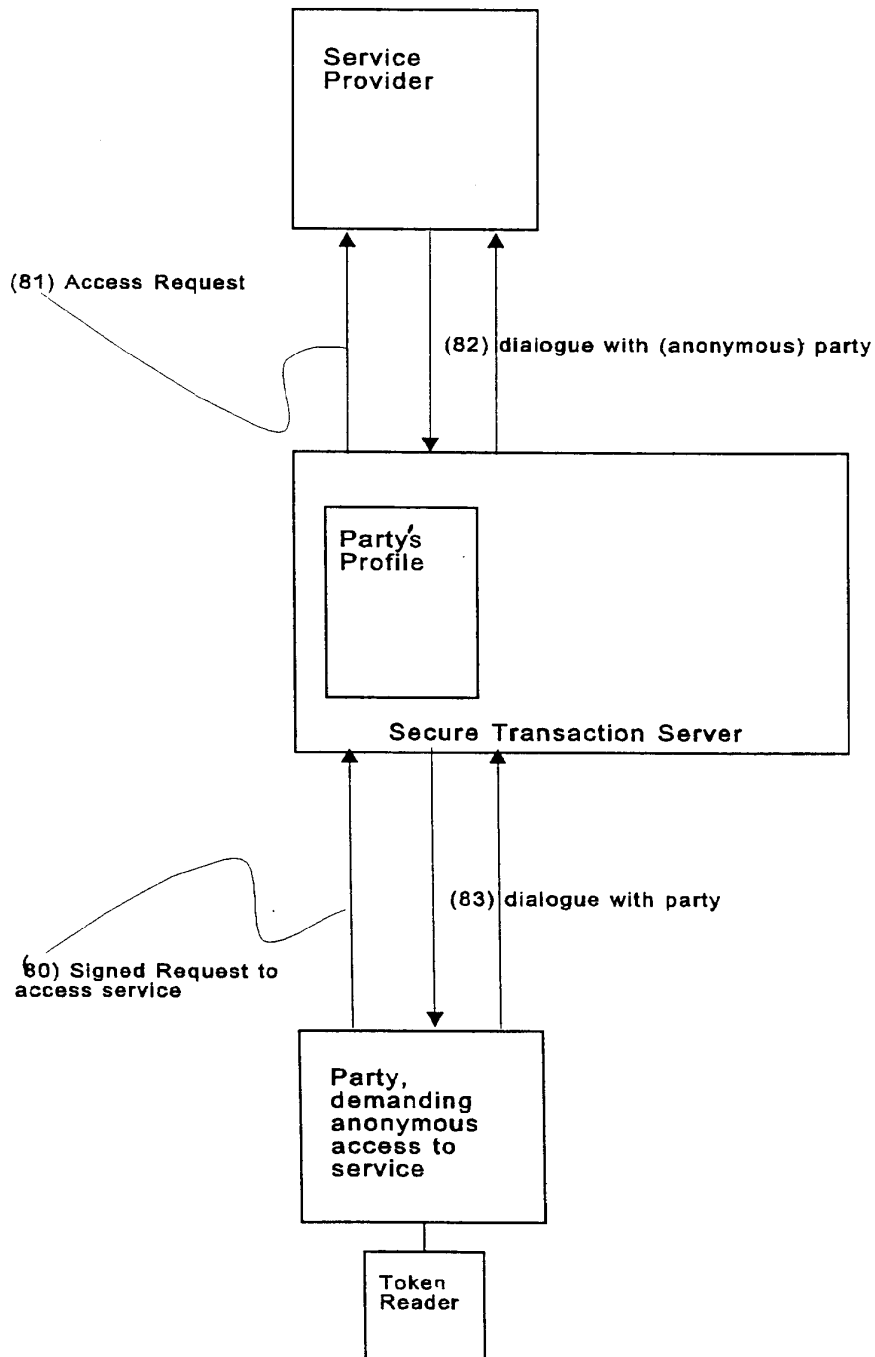


FIG. 12



Electronic Acknowledgement Receipt	
<b>EFS ID:</b>	37821551
<b>Application Number:</b>	16597773
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	8373
<b>Title of Invention:</b>	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA
<b>First Named Inventor/Applicant Name:</b>	Scott MacDonald WARD
<b>Customer Number:</b>	39083
<b>Filer:</b>	David Joseph Kenealy/Jill Tuncay
<b>Filer Authorized By:</b>	David Joseph Kenealy
<b>Attorney Docket Number:</b>	5061-0041CON
<b>Receipt Date:</b>	21-NOV-2019
<b>Filing Date:</b>	09-OCT-2019
<b>Time Stamp:</b>	15:51:07
<b>Application Type:</b>	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)	11-21-19_IDS.pdf	1036178	no	5
			cfe479c59bbb8a9baaf5f4a08dcf2a30f8330ea4		

### Warnings:

Information:					
2	Foreign Reference	WO2004015579A1.pdf	5271127	no	48
			6802053c4f68bc2aee59c774b97fb834ffa59a9		
Warnings:					
Information:					
3	Foreign Reference	EP1076279A1.pdf	5451904	no	46
			2e3fff3207852ddc51a5272c165b6be07cce42bf		
Warnings:					
Information:					
4	Foreign Reference	WO2001017296A1.pdf	748517	no	20
			10c46d31a3f44a2bffeec39e0ed16cd6f31e8298		
Warnings:					
Information:					
5	Foreign Reference	WO2002076127A1.pdf	4087987	no	38
			2f4a8ce9e2a27389bcbf6ef995871f322a4a2f1f6		
Warnings:					
Information:					
6	Foreign Reference	GB2338381A1.pdf	3399956	no	45
			489a888bb7ab13cd0360a996576a958f90b7cb3f		
Warnings:					
Information:					
7	Foreign Reference	EP1237324A1.pdf	6118966	no	46
			cfcd698a16880fe50fb3ff2899f56438b40fcd658		
Warnings:					
Information:					
8	Foreign Reference	WO0201334A2.pdf	9197760	no	65
			a7d3f2464bd0c6b1ff814ebb83b316e1298496fa		
Warnings:					
Information:					

9	Foreign Reference	WO200146918A2.pdf	2927862	no	28
			637beba2509d0e8add27a6b8e5233623b106e3e		
Warnings:					
Information:					
10	Foreign Reference	WO2000067143A2.pdf	5175120	no	56
			110eec80fc11eea3f139b713e9ae6ce901f8441e		
Warnings:					
Information:					
11	Non Patent Literature	NPL_Doc_1.pdf	7285178	no	52
			f80aab00b7a24b8f095066b5aa80f3ac3dd5bd7		
Warnings:					
Information:					
12	Non Patent Literature	NPL_Doc_2.pdf	399661	no	2
			d36a6b1faf6f9b4dd793043e04ab353880bd451bd		
Warnings:					
Information:					
13	Non Patent Literature	NPL_Doc_3.pdf	1131816	no	6
			ab963f846147ee73368dabdcbb7f8f290459ecb7		
Warnings:					
Information:					
14	Non Patent Literature	NPL_Doc_4.pdf	8369543	no	53
			9c4e72acc56e6f749ef71f72f53928fe28bfdb3a3		
Warnings:					
Information:					
15	Non Patent Literature	NPL_Doc_5.pdf	128090	no	1
			b4629061b4f3b2b9c3fe32d515e62bbb41cfa05c		
Warnings:					
Information:					
Total Files Size (in bytes):			60729665		



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 NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
16/597,773	10/09/2019	2491	1720	5061-0041CON	25	3

**CONFIRMATION NO. 8373**

## FILING RECEIPT



0000000112182203

39083  
KENEALY VAIDYA LLP  
3000 K Street, N.W.  
Suite 200  
Washington, DC 20007

Date Mailed: 10/29/2019

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

**Please verify the accuracy of the data presented on this receipt.** If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

### Inventor(s)

Scott MacDonald WARD, Aerdenhout, NETHERLANDS;  
Teunis TEL, Groningen, NETHERLANDS;

### Applicant(s)

WARD PARTICIPATIONS B.V., Aerdenhout, NETHERLANDS;

**Power of Attorney:** The patent practitioners associated with Customer Number 39083

### Domestic Priority data as claimed by applicant

This application is a CON of 10/560,579 04/23/2007  
which is a 371 of PCT/NL2004/000422 06/14/2004

**Foreign Applications** (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)

NETHERLANDS PCT/NL03/00436 06/13/2003 No Access Code Provided

**Permission to Access Application via Priority Document Exchange:** Yes

**Permission to Access Search Results:** Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

**If Required, Foreign Filing License Granted:** 10/25/2019

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 16/597,773**

**Projected Publication Date:** 02/06/2020

**Non-Publication Request:** No

**Early Publication Request:** No  
**Title**

METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A  
VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA

**Preliminary Class**

713

**Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications:** No

## **PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES**

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

**LICENSE FOR FOREIGN FILING UNDER**  
**Title 35, United States Code, Section 184**  
**Title 37, Code of Federal Regulations, 5.11 & 5.15**

**GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

**NOT GRANTED**

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

---

***SelectUSA***

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
16/597,773	10/09/2019	Scott MacDonald WARD	5061-0041CON

CONFIRMATION NO. 8373

FORMALITIES LETTER



39083  
KENEALY VAIDYA LLP  
3000 K Street, N.W.  
Suite 200  
Washington, DC 20007

Date Mailed: 10/29/2019

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

*Filing Date Granted*

**Items Required To Avoid Abandonment:**

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing.

Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- Additional claim fees of \$ **500** as an undiscounted entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

**SUMMARY OF FEES DUE:**

The fee(s) required within **TWO MONTHS** from the date of this Notice to avoid abandonment is/are itemized below. No entity status discount is in effect. If applicant is qualified for small entity status, a written assertion of small entity status must be submitted to establish small entity status. (See 37 CFR 1.27). If applicant is qualified for micro entity status, an acceptable Certification of Micro Entity Status must be submitted to establish micro entity status. (See 37 CFR 1.29 and forms PTO/SB/15A and 15B.)

- \$ **500** for **5** total claims over 20.
- \$( **0** ) previous unapplied payment amount.
- \$ **500** TOTAL FEE BALANCE DUE.

**Items Required To Avoid Processing Delays:**

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

**A new inventor's oath or declaration that identifies this application (e.g., by Application Number and filing date) is required. The inventor's oath or declaration does not comply with 37 CFR 1.63 in that it:**

- does not state that the above-identified application was made or authorized to be made by the person executing the oath or declaration.

Scott MacDonald WARD  
Teunis TEL

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts  
Commissioner for Patents  
P.O. Box 1450  
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".

<https://portal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/hchin/

---

<b>PATENT APPLICATION FEE DETERMINATION RECORD</b> Substitute for Form PTO-875						Application or Docket Number <b>16/597,773</b>				
<b>APPLICATION AS FILED - PART I</b>										
(Column 1)		(Column 2)		SMALL ENTITY		OTHER THAN SMALL ENTITY				
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)			
BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>	N/A	N/A	N/A			N/A	300			
SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>	N/A	N/A	N/A			N/A	660			
EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>	N/A	N/A	N/A			N/A	760			
TOTAL CLAIMS <small>(37 CFR 1.16(j))</small>	25	minus 20 = * 5				x 100 =	500			
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>	3	minus 3 = *				x 460 =	0.00			
APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00			
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL		TOTAL	2220			
<b>APPLICATION AS AMENDED - PART II</b>										
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OTHER THAN SMALL ENTITY		
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=	x	=		x	=
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=	x	=		x	=
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))									
TOTAL ADD'L FEE									TOTAL ADD'L FEE	
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=	x	=		x	=
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=	x	=		x	=
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))									
TOTAL ADD'L FEE									TOTAL ADD'L FEE	
<div style="font-size: x-small;">             * If the entry in column 1 is less than the entry in column 2, write "0" in column 3.              ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".              *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".              The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.           </div>										

<b>REQUEST FOR PARTICIPATION IN THE GLOBAL/IP5 PATENT PROSECUTION HIGHWAY (PPH) PILOT PROGRAM IN THE USPTO</b>			
Application No.:		First Named Inventor:	Scott MacDonald WARD
Filing Date:	October 9, 2019	Attorney Docket No.:	5061-0041CON
Title of the Invention:	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		
<b>THIS REQUEST FOR PARTICIPATION IN THE PPH PILOT PROGRAM ALONG WITH THE REQUIRED DOCUMENTS MUST BE SUBMITTED VIA EFS-WEB. INFORMATION REGARDING EFS-WEB IS AVAILABLE AT <a href="http://www.uspto.gov/patents-application-process/applying-online/about-efs-web">HTTP://WWW.USPTO.GOV/PATENTS-APPLICATION-PROCESS/APPLYING-ONLINE/ABOUT-EFS-WEB</a></b>			
<b>APPLICANT HEREBY REQUESTS PARTICIPATION IN THE PATENT PROSECUTION HIGHWAY (PPH) PILOT PROGRAM AND PETITIONS TO MAKE THE ABOVE-IDENTIFIED APPLICATION SPECIAL UNDER THE PPH PILOT PROGRAM.</b>			
<b>Office of earlier examination (OEE):</b> European Patent Office  <b>OEE application number:</b> <u>20170000713.2</u>  <b>Both the OEE application and the above-identified U.S. application have the following earliest date (filing or priority date):</b> <u>June 13, 2003</u>  <b>Type of OEE work product relied upon:</b> Decision to grant a patent  <b>Mailing date of OEE work product:</b> <u>August 2, 2018</u>			
<b>Supporting Documents</b>  <b>1. OEE Work Product and Translation</b>  A copy of the OEE work product and translation if not already in English:  <input checked="" type="checkbox"/> Attached <input type="checkbox"/> Previously submitted <input type="checkbox"/> Not required because the decision to grant a patent was the first office action  <input type="checkbox"/> Applicant requests the USPTO to attempt to obtain the OEE work product from the Dossier Access System or PATENTSCOPE  NOTE: If the applicant requests the USPTO to obtain the OEE work product electronically and such attempt is unsuccessful, the applicant will be required to supply the document. Accordingly, to avoid dismissal of the initial PPH request and potential denial of participation in the PPH program, the applicant should verify that the OEE work product is actually available via the Dossier Access System or PATENTSCOPE before requesting retrieval. If the applicant is unable to verify availability, then the applicant should submit the document with the PPH request.			
<b>2. References Cited in OEE Work Product</b>  An information disclosure statement (IDS) listing the references cited in the OEE work product and document copies (except U.S. patents and U.S. published patent applications):  <input type="checkbox"/> Attached <input type="checkbox"/> Previously Submitted <input checked="" type="checkbox"/> Not required because no references were cited in the OEE work product			

[Page 1 of 2]

This collection of information is required by 35 U.S.C. 119, 37 CFR 1.55, and 37 CFR 1.102(d). The information is required to obtain or retain a benefit by the public, which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS



**REQUEST FOR PARTICIPATION IN THE GLOBAL/IP5 PPH PILOT PROGRAM IN THE USPTO**  
(continued)

## 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 in 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.

### POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in the attached transmittal letter.

I hereby appoint:

- ☒ Practitioners associated with the Customer Number identified in the box at right as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A or equivalent):

**39083**

Please recognize or change the correspondence address for the application identified in the attached transmittal letter to:

- ☒ The address associated with the above-mentioned Customer Number.

I am the:

- ☐ Applicant.
- ☒ Assignee or Person to Whom the Inventor is Under an Obligation to Assign and am authorized to act on behalf of all assignees.
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor.
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document).

#### SIGNATURE of Applicant for Patent

Signature:



Date:

Oct 5, 2018

Name (printed):

Scott M. Ward

Telephone:

+31 6 53 848479

Title and Company:

WARD PARTICIPATIONS B.V.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		
<p>The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.</p> <p>This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.</p>			

**Secrecy Order 37 CFR 5.2:**

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

**Inventor Information:**

Inventor	1				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Scott	MacDonald	WARD		
Residence Information (Select One)    US Residency    • Non US Residency    Active US Military Service					
City	Aerdenhout		Country of Residence <sup>i</sup>	NL	
Mailing Address of Inventor:					
Address 1	31 Zuidlaan				
Address 2					
City	Aerdenhout		State/Province		
Postal Code	NL-2111 GB		Country <sup>i</sup>	NL	
Inventor	2				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Teunis		TEL		
Residence Information (Select One)    US Residency    • Non US Residency    Active US Military Service					
City	Groningen		Country of Residence <sup>i</sup>	NL	
Mailing Address of Inventor:					
Address 1	13 Esserlaan				
Address 2					
City	Groningen		State/Province		
Postal Code	NL-9722 SK		Country <sup>i</sup>	NL	
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button.					
Add					

**Correspondence Information:**

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON	
		Application Number		
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA			
Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).				
<input type="checkbox"/> An Address is being provided for the correspondence information of this application.				
Customer Number	39083			
Email Address	IP@KVIPLAW.COM	Add Email	Remove Email	

**Application Information:**

Title of the Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		
Attorney Docket Number	5061-0041CON	Small Entity Status Claimed	<input type="checkbox"/>
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	7	Suggested Figure for Publication (if any)	

**Filing By Reference:**

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

**Publication Information:**

<input type="checkbox"/> Request Early Publication (Fee required at time of Request 37 CFR 1.219)
<input type="checkbox"/> <b>Request Not to Publish.</b> I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application <b>has not and will not</b> be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

**Representative Information:**

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.			
Please Select One:	<input checked="" type="radio"/> Customer Number	<input type="radio"/> US Patent Practitioner	<input type="radio"/> Limited Recognition (37 CFR 11.9)
Customer Number	39083		

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		

## Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending	<a href="#">Remove</a>	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation of	10/560579	2007-04-23
Prior Application Status		<a href="#">Remove</a>	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
10/560579	a 371 of international	PCT/NL2004/000422	2004-06-14
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the <b>Add</b> button.			<a href="#">Add</a>

## Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)<sup>i</sup> the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Application Number	Country <sup>i</sup>	Filing Date (YYYY-MM-DD)	<a href="#">Remove</a>
PCT/NL03/00436	WO	2003-06-13	
Additional Foreign Priority Data may be generated within this form by selecting the <b>Add</b> button.			<a href="#">Add</a>

## Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.

NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		

## Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

**NOTE:** This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

### 1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

**A. Priority Document Exchange (PDX)** - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

**B. Search Results from U.S. Application to EPO** - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

### 2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

**NOTE:** Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		

## Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.			
<b>Applicant</b>	1	<input type="button" value="Remove"/>	
<p>If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.</p>			
		<input type="button" value="Clear"/>	
<input type="radio"/> Assignee	Legal Representative under 35 U.S.C. 117		Joint Inventor
Person to whom the inventor is obligated to assign.		Person who shows sufficient proprietary interest	
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:			
<div> <div></div> <div></div> </div>			
Name of the Deceased or Legally Incapacitated Inventor: <input type="text"/>			
If the Applicant is an Organization check here. <input checked="" type="checkbox"/>			
Organization Name	WARD PARTICIPATIONS B.V.		
<b>Mailing Address Information For Applicant:</b>			
Address 1	31, Zuidlaan		
Address 2			
City	Aerdenhout	State/Province	
Country	NL	Postal Code	2111 GB
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button. <input type="button" value="Add"/>			

## Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.
---



<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		

<b>Assignee</b>   1				
Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.				
				<input type="button" value="Remove"/>
If the Assignee or Non-Applicant Assignee is an Organization check here.				<input type="checkbox"/>
Prefix	Given Name	Middle Name	Family Name	Suffix
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<b>Mailing Address Information For Assignee including Non-Applicant Assignee:</b>				
Address 1		<input type="text"/>		
Address 2		<input type="text"/>		
City	<input type="text"/>	State/Province	<input type="text"/>	
Country i	<input type="text"/>	Postal Code	<input type="text"/>	
Phone Number	<input type="text"/>	Fax Number	<input type="text"/>	
Email Address	<input type="text"/>			
Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.				<input type="button" value="Add"/>

**Signature:**

**NOTE:** This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

<b>Signature</b>	/Eric D. Morehouse/		Date (YYYY-MM-DD)	2019-10-08
First Name	Eric	Last Name	Morehouse	Registration Number
				38565
Additional Signature may be generated within this form by selecting the Add button.				<input type="button" value="Add"/>

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	5061-0041CON
		Application Number	
Title of Invention	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

## Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

**Combined Declaration for Patent Application**

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; and that I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method and system for performing a transaction and for performing a verification of legitimate access to, or use of digital data  
the specification of which (check one)

- [ ] is attached hereto;  
 [ ] was filed in the United States under 35 U.S.C. §111 on \_\_\_\_\_, as  
 U.S. Appl. No. \_\_\_\_\_\*; or  
 [X] was/will be filed in the U.S. under 35 U.S.C. §371 by entry into the U.S. national stage of an international (PCT)  
 application, PCT/NL2004/000422; filed 14 June 2004, entry requested on \_\_\_\_\_\*; national stage  
 application received U.S. Appl. No. 10/560,579; §371/§102(e) date December 13, 2005  
 (\* if known)

and was amended on \_\_\_\_\_ (if applicable).  
 (include dates of amendments under PCT Art. 19 and 34 if PCT)

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; and I acknowledge the duty to disclose to the Patent and Trademark Office (PTO) all information known by me to be material to patentability as defined in 37 C.F.R. §1.56.

I hereby claim foreign priority benefits under 35 U.S.C. §§ 119 (a)-(d) and 365 (b) of any prior foreign application(s) for patent or inventor's certificate, or §365(a) of any prior PCT application(s) designating a country other than the U.S., listed below with the "Yes" box checked, and have also identified below, by checking the "No" box, any foreign application for patent or inventor's certificate or PCT international application having a filing date before that of the application on which priority is claimed:

PCT/NL03/00436	PCT Application	June 13, 2003	[X]	[X]
(Number)	(Country)	(Day Month Year Filed)	YES	NO
PCT/NL2004/000422	PCT Application	June 14, 2004	[X]	[X]
(Number)	(Country)	(Day Month Year Filed)	YES	NO

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional applications listed below:

_____ (Application No.)	_____ (Day Month Year Filed)
_____ (Application No.)	_____ (Day Month Year Filed)

I hereby claim the benefit under 35 U.S.C. §120 of any prior U.S. non-provisional application(s) or under §365(c) of any prior PCT international application(s) designating the U.S., listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in such U.S. or PCT international application in the manner provided by the first paragraph of 35 U.S.C. §112, I acknowledge the duty to disclose to the PTO all information which is material to patentability as defined in 37 C.F.R. §1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:


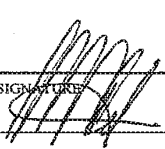
_____ (Application No.)	_____ (Day Month Year Filed)	_____ (Status: patented, pending, abandoned)
_____ (Application No.)	_____ (Day Month Year Filed)	_____ (Status: patented, pending, abandoned)

Title: Method and system for performing a transaction and for performing a verification of legitimate access to, or use of digital data

U.S. Application filed December 13, 2005, Serial No. 10/560,579

PCT Application filed June 14, 2004, Serial No. PCT/NL2004/000422

I hereby further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF FIRST INVENTOR WARD, Scott MacDonald	INVENTOR'S SIGNATURE 	DATE March 1, 2006
RESIDENCE Aerdenhout, the Netherlands	CITIZENSHIP Netherlands/ nat. U.S.	
POST OFFICE ADDRESS 31, Zuidlaan 2111 GB AERDENHOUT the Netherlands		
FULL NAME OF SECOND JOINT INVENTOR TEL, Teunis	INVENTOR'S SIGNATURE 	DATE March 1, 2006
RESIDENCE Groningen, the Netherlands	CITIZENSHIP Netherlands/nat. NL	
POST OFFICE ADDRESS 13, Esserlaan 9722 SK GRONINGEN the Netherlands		
FULL NAME OF THIRD JOINT INVENTOR	INVENTOR'S SIGNATURE	DATE
RESIDENCE	CITIZENSHIP	
POST OFFICE ADDRESS		
FULL NAME OF FOURTH JOINT INVENTOR	INVENTOR'S SIGNATURE	DATE
RESIDENCE	CITIZENSHIP	
POST OFFICE ADDRESS		
FULL NAME OF FIFTH JOINT INVENTOR	INVENTOR'S SIGNATURE	DATE
RESIDENCE	CITIZENSHIP	
POST OFFICE ADDRESS		
FULL NAME OF SIXTH JOINT INVENTOR	INVENTOR'S SIGNATURE	DATE
RESIDENCE	CITIZENSHIP	
POST OFFICE ADDRESS		
FULL NAME OF SEVENTH JOINT INVENTOR	INVENTOR'S SIGNATURE	DATE
RESIDENCE	CITIZENSHIP	
POST OFFICE ADDRESS		

ALL INVENTORS MUST REVIEW APPLICATION AND DECLARATION BEFORE SIGNING. ALL ALTERATIONS MUST BE INITIALED AND DATED BY ALL INVENTORS PRIOR TO EXECUTION. NO ALTERATIONS CAN BE MADE AFTER THE DECLARATION IS SIGNED. ALL PAGES OF DECLARATION MUST BE SEEN BY ALL INVENTORS.

Method and system for performing a transaction and for performing a  
verification of legitimate access to, or use of digital  
5 data

The present invention relates to a method and system for  
performing an electronic transaction and for performing a  
verification of legitimate access to, or use of digital data.

10 At present, numerous transactions are being handled by  
electronic means in digital format. Digital networks have evolved  
which enable parties of different kind across the world to  
communicate with each other and to exchange data and information to  
reach desired transactions.

15 The data and information exchanged in said transactions may be  
legally privileged or protected by copyright, for example. However,  
digital information may be very easily copied and spread without a  
trace of who illegally copied and spread the data.

Further, in particular in transactions involving private  
20 network access, financial commitments, settlements and/or payments,  
each party involved in such a transaction wants to identify any  
other party, or at least, to be able to track any other party, if  
after completion of the transaction a problem arises. For such  
identification purposes, it is known to use personal identifiers,  
25 such as passwords, Personal Identification Numbers (PIN), and the  
like, which are only known to a specific user. However, using  
personal identifiers over public networks like the Internet, there  
is a possibility that the personal identifier becomes known to  
another person, enabling this other person to do transactions or  
30 gain access to digital data presenting himself as somebody else. If  
a problem arises after completion of the transaction, it is not  
possible to track the real transaction partner, as its personal  
identifier may have been used by a malicious user of the public  
network.

35 For a more secure transaction, it has been proposed in European  
Patent Application No. 1 219 088 to use a trusted third party  
transaction server comprising profiles of the transaction parties.

The transaction server verifies the identity of the transaction parties by using authentication data comprising a table of random data for verifying a digital signature. The digital signature is generated from a random token using a token reader. The table of random data corresponds to data collected from said random token. Thus, a digital signature originating from the random token and being different for every subsequent transaction is virtually impossible to forge and therefore uniquely identifies the transaction party.

10 A disadvantage of this system and other systems employing additional hardware is that the additional hardware, e.g. a token and a token reader, should be supplied to every possible transaction party.

15 It is therefore an object of the present invention to provide a method and system for performing an electronic transaction or electronic verification or identification without requiring additional hardware.

At least this object is achieved in the present invention by a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party. The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party. In a preferred embodiment, the second transaction party provides digital data to the first transaction party.

30 In a further aspect, the present invention provides a method for performing a verification of legitimate use of digital data on an electronic device. The method comprises providing authentication data in a memory of said electronic device which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication

software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to an application which accesses digital data having a digital signature embedded therein; and comparing the  
5 digital signature embedded in the digital data with the provided digital signature.

In another aspect, the present invention provides a method for encrypting digital data on an electronic device using an encryption key, the method comprising gathering session specific data; hashing  
10 said session specific data to obtain reference numbers referring to positions in an authorization table stored in said electronic device; generating said encryption key from the characters stored in the authorization table at said positions; and encrypting said digital data using said encryption key.

15 In a further aspect, the present invention provides systems for performing said methods.

Without use of any additional hardware, a transaction party in an electronic transaction may be identified with virtually no possibility for fraudulent use of the method. In a private network  
20 access transaction, the first transaction party is uniquely identifiable by its digital signature. Said digital signature is provided to the second transaction party that may store the digital signature. If a problem arises later, the first transaction party may be traced and identified by the digital signature provided to  
25 the second transaction party.

If digital data are provided to the first transaction party, e.g. copyright protected files such as music and the like, that are digitally signed according to the present invention, i.e. a digital signature is embedded in the digital data, said digital data may be  
30 traced, if they are later found to be illegally copied or spread. The embedded digital signature is uniquely traceable to the original first transaction party that received said digitally signed digital data.

Digital data digitally signed according to the present  
35 invention are stored in a storage medium of a device having the authentication software installed. The signature has been generated in accordance with the authentication data stored in said device and



thereafter embedded in the digital data to protect the data and to be able to trace a malicious user.

The signature embedded in the data may also be employed to prevent that the data are illegally used, since the signature may be regenerated by the device at any time. As a regenerated signature should be identical to the one embedded in the digital data, a comparison of the embedded signature with the regenerated signature provides information whether the digital data is rightfully installed on the device. If the comparison shows that the signatures are identical, the data may be accessed by the device, and, for example, an application comprised in said digital data may be run or said digital data may be accessed by any other application, for example for playing music represented by said digital data. When the signatures are not identical, the digital data are illegally installed, e.g. copied from another device, and they may not be accessed and read by the device and an error signal may be generated.

Further aspects and features of the present invention are disclosed in the dependent claims.

The invention and its aspects, features, and advantages will be more readily appreciated as the same becomes better understood by reference to the following detailed description and considered in connection with the accompanying drawings provided by way of non-limiting examples, in which drawings like reference symbols designate like actions or parts.

Fig. 1 illustrates a chart of an installation method for a new device according to the present invention.

Fig. 2 illustrates a chart of an installation method for an existing device according to the present invention.

Fig. 3 schematically illustrates a computer configuration having authentication software and an authentication table installed in the BIOS according to the present invention.

Fig. 4 illustrates a chart of another installation method for an existing device according to the present invention.

- Fig. 5A schematically illustrates a computer configuration having authentication software and an authentication table installed in a user accessible memory.
- Fig. 5B schematically illustrates a memory location secured by two encryption layers.
- Fig. 6 illustrates a chart of a method to generate a digital signature from a random table.
- Fig. 7 illustrates a chart of a method to generate a personalized traceable digital file according to the present invention.
- Fig. 8 illustrates a chart of an off-line authentication of digital rights according to the present invention.
- Fig. 9 illustrates a chart of an on-line authentication or transaction method according to the present invention.
- Fig. 10 illustrates a chart of an off-line authentication of digital rights of digital data stored in a removable memory according to the present invention.

Referring to Figs. 1, 2, 4 and 6 - 9, in the methods illustrated in the respective Figures actions of different actors are shown in a number of columns. In the vertical direction, actions of the actors have been arranged in an essentially chronological order from top to bottom. Thus, actions described in one row of a chart essentially are performed prior to actions in a subsequent (lower) row of the chart, although this may not always be necessary, as indicated in some instances. The arrows in the Figures indicate a flow of data, which may be transferred through a suitable connection, such as a hard-wired connection, or a wireless connection, or a combination thereof, using an appropriate protocol in a network connection.

In this context, an installation method is to be understood as a method for installing software at some actors involved in the present method for enabling tracking and tracing of at least one transaction party. Further in this context, a new device relates to any electronic device containing a Basic In Out System ("BIOS", "Boot agent" etc.) with any associated secure storage/memory location, e.g. a computer, server, printer, personal digital

assistant, mobile telephone, which is being manufactured, or has been manufactured, but has not yet been delivered to an end-user, whereas an existing device relates to any electronic device containing a Basic In Out System which has been delivered to an end-user, and may or may not have been used by the end-user. The Basic In Out System is only referred to as a system for accessing a memory location in a memory that is directly or indirectly connected to the electronic device. However, as is described hereinafter, the method according to the present invention may employ such a BIOS system to securely store certain digital data and/or such a BIOS system may be provided with an encryption system. A secure storage location and/or an encryption system are not essential to the BIOS system with respect to the present invention.

Fig. 1 illustrates a first preferred embodiment of the present invention. Referring to Fig. 1, an installation method for a new device is illustrated. The method of Fig. 1 includes five actors: a random table supplier, a trusted third party, a BIOS manufacturer, a BIOS and authentication software. The first, second and third actors are physical entities, e.g. persons or companies, whereas the fourth and fifth actors are software embedded in a device.

In the first column of Fig. 1 actions of the random table supplier are shown. This supplier generates and supplies a random table or, essentially, random data to another actor.

A random table is defined as a table comprising random numbers. Such a table may be created by suitable software as such. Alternatively, U.S. Patent Application No. 5,354,097 discloses another method of producing random numbers by optically scanning a randomly shaped material, such as a non-woven material. Likewise, a two-dimensional pattern may be used, which pattern has been formed by transformation of a bit string as disclosed in European Patent Application No. 1,219,088. From data collected during the reading of the randomly shaped material or pattern, random numbers can be derived, e.g. by using a predetermined algorithm.

The second column of Fig. 1 relates to actions of a trusted third party (TTP), which may be selected by the random table supplier, a BIOS manufacturer and/or the owner of any application or service. The TTP generates a bit string from the random table and

other data, to be elucidated below. Further, the TTP stores the bit string and other data used to generate it. The generated bit string is sent to the BIOS manufacturer.

It is noted that in another embodiment of the present invention, the bit string may be generated and stored by another actor than the TTP. For example, in case of network access identification, the network administrator and/or a network server may generate and store the bit string such that when a user attempts to access the private network, the network administrator or server may regenerate the digital signature of said user. Thus, the user may unambiguously be identified by the network administrator or server before the user is granted access to the network.

The actions of the BIOS manufacturer are represented in the third column of Fig. 1. The BIOS manufacturer installs authentication software in a BIOS and stores the bit string supplied by the TTP in the BIOS.

The BIOS which is supplied by the BIOS manufacturer, is also an actor in the method, and its actions are represented by the fourth column of Fig. 1. The BIOS may be coupled to the TTP to receive a decryption code and enable the authentication software to decrypt an authentication table, which will then be encoded again and stored in a secure part of the device, only accessible to the BIOS.

The fifth column of Fig. 1 represents the actions of the authentication software, which may be supplied by the random table supplier or by any other third party and is stored in the BIOS. The authentication software may be coupled to the TTP to decrypt the bit string in the BIOS into an authentication table, encrypts the authentication table again and stores it in the secure part of the device, only accessible to the BIOS. Further, the authentication software stores and activates a digital-signing algorithm to generate a session- or transaction-specific digital signature. The authentication software preferably runs in a separate operating environment in the BIOS or in a console and is independent from and inaccessible to the operating system (OS) on the device.

As indicated in Fig. 1, the installation method for a new device for use with the tracking and tracing method according to the present invention starts with the generation of a random table, e.g.

a table comprising randomly chosen characters or numbers, according to cell 2. The numbers may be generated using software employing an algorithm. Alternatively, as disclosed in U.S. Patent Application No. 5,354,097, random numbers may be deduced from a randomly shaped material, such as a non-woven material. For example, in a non-woven material, fibers are arranged in a random order due to the randomness of the production process. From the geometry of the fibers of the non-woven material numbers may be calculated. For example, a number that may be calculated, may be the angle between two fibers in an area of the material, or the number of fibers crossing an imaginary line. The calculated numbers will be random, as the geometry of the material is random. Another alternative manner, as disclosed in European Patent Application No. 1,219,088, to collect random numbers is by transforming a bit string into a two-dimensional pattern using an algorithm. As described above from this two-dimensional pattern a table of random numbers may be calculated based on the geometry of the pattern.

The random table supplier may generate the random table and supply it to the TTP. Alternatively, the random table supplier may only supply a piece of non-woven material or software to generate two dimensional patterns according to European Patent Application No. 1,219,088, from which the TTP may then generate one or more random tables.

According to cell 4, a BIOS manufacturer embeds or installs authentication software in the BIOS. The authentication software may be supplied by the random table supplier, generated by the BIOS manufacturer or supplied by a third party.

The authentication software preferably has its own unique serial number, software ID and/or private encryption key. Said unique number may be advantageously employed in the installation method and in a track and trace method, as explained below.

Further, according to cells 6A-6E, the TTP generates a unique bit string, in the following way. First, according to cell 6A, the TTP collects fixed data, such as a unique serial number from a hardware device, a processor for example, or the unique software ID embedded in the authentication software in the previous step.

Preferably, the fixed data refer to a number or other character string stored in or related to the specific BIOS. The bit string, which will be generated from the fixed data, among others, is preferred to have a strong relation with the BIOS in which it is stored. Such a strong relation makes forging the bit string difficult.

In the next step 6B, the TTP selects a random table previously received from the random table supplier, or it generates a random table using a marker, such as a piece of non-woven material or using a random two dimensional pattern.

The TTP further generates a data string according to cell 6C. The three data sets, e.g. the fixed data, the random table and the data string, are used to generate a bit string using a predetermined algorithm according to cell 6D. The bit string is stored in a database according to cell 6E together with the fixed data and the data string, which were used to generate it.

The bit string is supplied to the BIOS manufacturer, preferably in an encrypted way. The BIOS manufacturer embeds the bit string, like the authentication software, in the BIOS according to cell 8. At this point, a computer may be assembled having a BIOS which comprises authentication software and a bit string. Thereafter, the computer may be sold, installed and connected to a network, such as the Internet.

At the first start-up of the computer according to cell 10 the BIOS or the authentication software uses a connection to a network, such as the Internet, to establish a connection with the TTP. The connection is being made to collect the data string, which is needed to decrypt the bit string and generate an authentication table. The BIOS or the authentication software may be identified by the TTP using the fixed data, which is a unique number as mentioned above.

The connection with the TTP does not need to be initiated by the BIOS, but may also be initiated by the authentication software, which is embedded in a secure part of the BIOS, or by a third party application. The connection preferably is made without a possibility for a user or the operating system of the computer to interrupt the procedure.

When a connection with the TTP is established, the BIOS or authentication software sends its identification number upon request of the TTP according to cell 12. The TTP returns, possibly upon request of the BIOS, the decrypt key, and the authentication software uses the decrypt key according to cell 14 to generate an authentication table from the bit string which was embedded in the BIOS according to cell 8.

Next, according to cell 16, the authentication table is encoded by the authentication software to prevent that the authentication table may be uncovered (e.g. by a person). Uncovering of the authentication table would weaken the protection conferred by the method and should therefore be virtually impossible.

The encoding is performed using a number, which is specific for the BIOS. When the authentication table is needed later, it may then be decoded by the authentication software using that BIOS-specific number. The BIOS-specific number may be a unique serial number or any ordinary encryption key incorporated in or generated by the BIOS, for example. The BIOS-specific number is sent to the authentication software according to cell 18 after a request from the authentication software according to cell 16.

The installation in a new device is completed according to cell 20. The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS may also be any other secure location in the device. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS and/or authentication software. Storing the authentication table in a secure location further decreases the possibility of retrieval of the authentication table.

The device is now set-up. Authentication software and an encoded authentication table are installed in the BIOS. Further, in a database of a trusted third party (TTP), data are stored which enable the TTP to generate the same authentication table when later needed.

Fig. 2 illustrates a second preferred embodiment. In the second preferred embodiment, the installation method is performed on an

existing computer device. The existing device does not have authentication software installed in its BIOS, nor does it have an encrypted bit string stored in its BIOS at the time a digital signature according to the present invention is needed for any track and trace purpose, including a particular electronic session, e.g. a network access procedure, or an on-line transaction. Thus, compared to a new computer device, further steps are necessary in the above-described installation method to install the necessary software and authentication table in a secure memory location of the device. Such a secure memory location may be a part of a secure memory or it may be an encrypted part of a non-secure memory, such as a user-accessible memory.

In Fig. 2, there are five columns representing five actors: a random table supplier, a TTP, a BIOS, authentication software and a third party application. Compared to Fig. 1 the BIOS manufacturer is no actor in the installation method for an existing device, but a third party application is introduced as an actor. The third party application is an on-line application, usually intended to do on-line logical access, or on-line transactions. The third party application requests a digital signature from the existing device, which does not have the authentication software and the authentication table. The third party application therefore requests that the device installs the authentication software first before continuing the transaction.

According to cell 22, the random table supplier generates and supplies a random table, similar to the action according to cell 2 in Fig. 1.

According to cell 24, a device having a BIOS without an authentication table and/or authentication software initiates an on-line transaction with a third party application. The third party application, however, requires an authentication according to the present invention. Therefore, the third party application redirects the requesting device to the TTP to obtain the authentication software and an authentication table.

The TTP generates and stores a bit string according to cells 26A-26E, similar to cells 6A-6E of Fig. 1. According to cell 26A the TTP collects fixed data, according to cell 26B a random table or a



random marker is selected, and according to cell 26C the TTP selects a data string. From these three data sets, the TTP generates a bit string according to cell 26D and stores the bit string, the fixed data and the data string according to cell 26E. Further, the bit  
5 string is sent to the requesting device together with the authentication software, both encrypted.

The authentication software and the bit string are put in a part of the BIOS which is accessible to the operating system according to cell 28A. Next, according to cell 28B, the device needs  
10 to reboot to store the authentication software and the bit string in a secure part of the BIOS which is not accessible for the operating system.

At this point, the method continues as the method illustrated in Fig. 1 from cell 10 and further. According to cell 30 the device  
15 having newly stored authentication software and a bit string in its BIOS contacts the TTP again and requests a decrypt key, for example the data string, according to cell 32.

According to cell 34, the decrypt key is used to decrypt the authentication software and to decrypt the bit string and generate  
20 the corresponding authentication table. Next, according to cell 36, the authentication software verifies the authentication table and requests BIOS-specific data from the BIOS to encrypt the authentication table again.

According to cell 38 the BIOS sends BIOS-specific data, for  
25 example any common encryption key, to the authentication software in reply. The method is completed according to cell 40, wherein the authentication table is encrypted and stored in a secure part of the BIOS. Any data remaining in the operating system from the transfer of the authentication data and authentication software is deleted by  
30 the BIOS.

Using this installation method every device having a BIOS and capable of communicating with external, third party applications may have the authentication software and an authentication table installed, for example a personal computer, a cellular phone, a  
35 hand-held personal digital assistant with wireless communication capabilities or any other device containing a BIOS, as mentioned above.

When the software and the table are correctly installed in the BIOS, particular sessions, track and trace or on-line transactions may be initiated and performed using the authentication and digital signing method.

5        Fig. 3 schematically illustrates a possible configuration of a device having the authentication software and an authentication table installed. The device consists of several components, commonly referred to as hardware 42. Exemplary hardware components are a processor, a hard drive, or other memory and a keyboard.

10       All the hardware devices are controlled by the BIOS 44. The BIOS 44 is a software package stored in a read-only memory (ROM), usually located on a main board, which is one of the hardware components of an electronic device.

15       The BIOS 44 controls most of the instructions and data flowing from one component to another. Data or instructions coming from one component and addressed to another need to pass through the BIOS 44. The BIOS 44 may check whether the instructions to the other component are allowed or not, as not every component is accessible for any other component.

20       The BIOS 44 may comprise an encryption key 46. Such an encryption key 46 may be used to encrypt data and only another party who knows the encryption key 46 may be able to decrypt the data.

25       The operating system 48 creates a run-time environment for any user applications. Therefore, it may be stated that the operating system 48 is an interface between a user and the hardware 42. A user may instruct the operating system 48 to run an application 50. If the application 50 needs data that are stored on the hard drive, the application 50 requests the data from the operating system 48. The operating system 48 in turn requests the data through the BIOS 44 from the hardware 42, i.e. the hard drive. The data coming from the hardware 42 pass through the BIOS 44 and the operating system 48 before they arrive at the requesting application 50. Via this protocol the BIOS 44 may control the accessibility of certain data or hardware devices 42 and thus the use of particular software. It may prohibit, for example, the operating system 48 to access certain parts of a hard drive or start certain applications.

A console 52 is an environment, which runs all the processes in the computer without user interruption. The console 52 plays an important role at start-up of the computer, instructing the hardware devices 42 via the BIOS 44 to start and report their status. In case of errors, not only at start-up but also during operation, the BIOS 44 sends the error messages coming from the hardware 42 to the console 52. The console 52 then handles and corrects the errors. Thus, the console 52 runs more or less stand-alone the computer. A user may not interrupt or influence the console 52. Any instructions coming from the operating system 48 intended for the console 52 may therefore be blocked by the BIOS 44.

In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48. That means that the operating system 48 does not know that the applications and data in the storage locations are present and maybe even running in a separate environment. For example, when a hardware device 42 reports an error, the console 52 may run a recovery application 56 to handle the error such that the hardware device 42 recovers from the error and is able to continue its operation. The operating system 48 has probably not even noticed that there was an error.

An authentication table may be securely stored in the secure storage location 60. In such a part of the computer, commonly seen as a part of the BIOS 44, the authentication table is unreachable for the operating system 48 and thus for a user.

With the authentication table securely stored in the secure storage location 60, the authentication software 54 should run in an environment in the BIOS 44, thus preventing that the authentication table is accessible to the operating system 48 at any time. Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device.

The secure area 62 may further comprise other applications and/or memory locations. For example, another memory location 58 may store a log file of all actions performed by the BIOS 44 in the

secure area 62, or a log file storing any other actions by any other application or all network data exchange, for example all data exchanges over the Internet.

Further, the digital signing algorithm is preferably stored in  
5 a secure part of the BIOS 44. This algorithm may be protected like the authentication table in the secure storage location 60, because if both become known to a user, the user may be able to forge a digital signature. By locking the authentication table and the authentication software 54 including the digital signing algorithm  
10 in the secure area 62 they are secured.

In a third preferred installation method illustrated in Fig. 4, the authentication data, e.g. authentication table, is stored in a memory that is accessible to an operating system of the device and possibly to a user of said device. To prevent that the user may  
15 obtain the authentication data, which should be prevented as described above, the authentication data are encrypted. Thus, the user may only obtain encrypted authentication data. To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing  
20 environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Further, a decryption algorithm and/or a  
25 decryption key should not be obtainable to any user.

It is preferred to encrypt the authentication data twice, i.e. using at least two encryption layers. If a malicious user succeeds in decrypting the authentication data once, thus succeeds in decrypting a first encryption layer, a second encryption layer still  
30 protects the authentication data.

Preferably, a first encryption layer uses an encryption algorithm employing a device specific encryption key, for example an encryption key associated with part numbers of one or more parts of the device. A second encryption algorithm employs preferably an  
35 encryption layer associated with supplied data, such as a supplied bit string or a supplied encryption algorithm. Therefore, the third preferred embodiment is especially suitable for use with an

electronic device previously provided with an encryption system for encrypting data using a device specific encryption key. Such devices are known in the art and are commonly and commercially available. In Fig. 4, the third preferred embodiment is illustrated in relation to such a device.

Fig. 4 shows two actors in the installation method. A first actor is a third party application. The third party application may be an application for performing an electronic transaction that needs authentication data, and if no authentication data are found on the device, initiates an installation of the authentication software. Also, it may be an application or an operating system controlled by a user to initiate an installation of the authentication software. The second actor in the installation method is the authentication software. The method may be extended with a third actor. For example, the BIOS may be enabled to encrypt data using a device specific encryption key. In such a case, the BIOS may generate an encryption layer according to cell 210 or 212, as will be described hereinafter.

Now referring to Fig. 4, the third preferred installation method according to the present invention starts with obtaining and installing an authentication software package, for example obtained through the Internet, as indicated in cell 200.

After installation of the authentication software, the authentication software obtains a master bit string (MBS) according to cell 202. The master bit string (MBS) may be a large array of characters, for example 2048 numbers. The MBS may be embedded in the authentication software package, and may in that case not need to be obtained separately as indicated in Fig. 4.

From said master bit string (MBS), according to cell 204, the authentication software selects a bit string. Said bit string thus comprises a number of said characters, for example 128 characters randomly selected from the MBS.

The authentication software generates authentication data, e.g. an authentication table, from the bit string in accordance with cell 206. The authentication software runs in the above-mentioned secure processing environment. Thus, the algorithm generating the authentication data from the bit string is protected against

malicious users who want to obtain the authentication data or said algorithm.

The authentication data are generated from a bit string that is generated at the device of the first transaction party and the bit string and is therefore not known to a second transaction party or a trusted third party (TTP). Thus, the second transaction party cannot generate and store (a copy of) the authentication data. For identification of the first transaction party in a transaction, the authentication data need to be known to the second transaction party or a TTP. Therefore, if the authentication data are to be used for identification in a transaction, a copy of the bit string is returned to the supplier of the authentication software package or to another party, which party may be a second transaction party or a TTP in accordance with cell 208.

The bit string may be encrypted or not, while it is returned to the other party. If it is ensured that the algorithm for generating the authentication data from said bit string is not known to any user, the actual bit string used for generating the authentication data may become known to any user.

The generated authentication data may be stored in a user accessible memory, such as a hard drive of a computer device, or on a removable memory such as a floppy disk or a flash memory. As mentioned above, the authentication data need to be protected from any user, especially from a malicious user. Thereto, the authentication data are encrypted by the authentication software, preferably running in a secure processing environment, before the authentication data are stored in said user accessible memory.

According to cell 210, the authentication data are encrypted by a first encryption layer, e.g. by an encryption algorithm which is part of the authentication software package. The algorithm is secured such that it does not become known to any user. According to cell 212, the authentication data are further encrypted by a second encryption layer, e.g. using an encryption key which is associated with a hardware device serial number, or the like, thereby preventing illegal copying to another device. If the authentication data are stored in a removable memory, for example a flash memory, the encryption key may be associated with a serial number of the

removable memory. Preferably, the encryption key is (also) associated with a user identifying number, e.g. a personal identifying number (PIN) or a number or a template associated with a fingerprint of the user, or the like.

5       According to cell 214, the encrypted authentication data are stored in the memory. Thus, the device is provided with the authentication software and the authentication data, the authentication data being secured by encryption from becoming known to a user. The device is installed for performing the transaction  
10   method and the legitimate use verification method according to the present invention.

      Fig. 5A illustrates schematically an embodiment of an electronic device for performing the installation method according to the third embodiment of the present invention. The schematic  
15   diagram of Fig. 5A is similar to the diagram of Fig. 3. The illustrated electronic device comprises hardware 42, a BIOS 44, an operating system 48, a user application 50, and a console 52. From the below description, it will become apparent to those skilled in the art that the secure area 62 (as indicated in Fig. 3) is not  
20   essential for performing the third embodiment of the installation method according to the present invention.

      After installation of the authentication software package according to the method illustrated in Fig. 4, an encrypted data package is stored in a memory location 63 that is a part of the  
25   hardware 42. The memory location 64 is accessible to a user via a user application 50. The application 50 may request data from the memory location 63 by sending a request to the operating system 48. The operating system 48 may obtain the data stored at the memory location 63 possibly via the BIOS 44. The authentication software is  
30   also installed and stored in a user accessible memory location as indicated. The authentication software may be encrypted, or not.

      The data stored at the memory location 63 are encrypted authentication data. The encryption of the authentication data renders the data unusable to the user application 50. Fig. 5B  
35   illustrates the encryption of authentication data 63A stored in memory location 63. The authentication data 63A are encrypted twice as illustrated in Fig. 4. A first encryption layer 63B and a second

encryption layer 63C protect the authentication data 63A. Said encryption layers 63B and 63C may be removed by decryption by the corresponding application.

One encryption layer may be decrypted by the authentication software 54. The other encryption layer may be an encryption/decryption application stored in the electronic device using a device specific encryption key. Such an application may be stored in the BIOS or in a secure area 58 (as indicated in Fig. 3) and may use the encryption key 46. In other embodiments, there may be only one encryption layer or any other encryption/decryption application may be employed.

Referring to Fig. 5A again, the application 50 may require a digital signature for an electronic transaction or for verification of legitimate use of digital data. Thereto, the authentication software 54 is executed to initiate decryption of the authentication data stored in memory location 63 and for generating said digital signature. From one authentication table, numerous digital signs may be generated. Fig. 6 illustrates a method to generate a certain digital signature from an authentication table and illustrates how numerous different digital signs may be generated. Using different digital signs for every action minimizes the chance of infringements or forgery and maximizes the ability to trace the origin of an illegal copy.

Fig. 6 shows two actors, a BIOS and the authentication software. The authentication software starts by collecting data according to cell 64. There are multiple components required. First a fixed component, which is identical for each instance a digital signature is generated. Further, a variable component is used, which enables the method to generate a numerous amount of different, but traceable digital signs. A system trace component, e.g. a transaction ID, also depends on the instance. Two variable components make it virtually impossible to derive the authentication table from a number of generated digital signs. Optionally, a personal identification number (PIN) or password may be used to identify the user as well as the device in which the authentication software is embedded.



When all the required data is collected, the authentication software hashes the collected data into a bit string and translates the bit string into a number of pointers according to cell 66. Using the encryption key, the authentication table is decrypted by the BIOS according to cell 68. The pointers generated according to cell 66 point to positions in the authentication table. Using the pointers, the authentication software gathers a series of random numbers from the authentication table according to cell 70. Thereafter, according to cell 72, the BIOS encrypts the authentication table again. Finally, the digital signature accompanied by the variable data components that were used to generate the digital signature are transferred to the requesting application according to cell 74.

A digital signature generated according to the present invention may be used in various ways. First, it may be used to track and trace digital files. Second, it may be used off-line to prevent illegally copied software from being used and third, the digital signature may be used in on-line transactions and authentication, for example network access authentication.

Fig. 7 illustrates the track and trace method for digital files. Information contained in a digital file may be protected by copyrights, for example. Software, films or music may be bought and downloaded from the Internet. However, such digital files containing protected information may be very easily illegally copied and spread without a trace of whom copied and spread the file. Adding a trace in the digital file makes it possible to trace to origin of the illegal copy. Rightful owners of such a digital file, e.g. music file, will therefore not spread the file, but they may use the file on different locations, for instance on their computer at home and on their portable digital player, e.g. MP3-player.

According to cell 80, a third party application receives a request for a digital file, e.g. software or a music file. The third party application is intended to provide protected or traceable digital files and may be any application adapted therefor. It may be an on-line application, for example.

Then, according to cell 82, the third party application asks the requesting device if the authentication software is installed

and running. According to cell 84, the BIOS responds. If the authentication software is not installed or running, the device is rerouted to a trusted third party to download and install the software according to cell 24 of Fig. 2.

5        If the software is installed and running, the third party application starts the transaction by transferring a transaction ID to the requesting device's BIOS according to cell 86. According to cell 88, the BIOS and authentication software generate a digital signature in accordance with the method described in relation to  
10       Fig. 6 using the transaction ID they received.

      The generated digital signature together with the variable data component, e.g. date/time, is then sent to the third party application. According to cell 90, the third party application stores the digital signature, the variable data component,  
15       transaction ID, which it provided itself, and some data that identifies the requesting device, e.g. an IP-number, Ethernet number, serial number or any other unique identifying number, in a database.

      Further, the digital signature is embedded in the requested  
20       digital file according to cell 92. This does not necessarily need to be conducted after the storage according to cell 90, but may also be done before or at the same time.

      The digital signature is preferably unique for the requested digital file. When an illegal copy of the digital file is found, the  
25       third party application will be able to check in its database to which device it sent that particular file after reading the digital signature embedded in the file.

      The file with the digital signature embedded therein is sent to the requesting device according to cell 94, which completes the  
30       method.

      In a further embodiment, the authentication software may be provided with a system for signing a digital signature according to the present invention using at least a part of a software identification number (software ID) or any other application  
35       identifying character string, hereinafter referred to as a private key. The private key is registered at a third party, for example the authentication software provider which supplied said private key.

The private key may be used to uniquely sign the digital signature and append the signed digital signature to the digital signature, which is sent to the second transaction party. The second transaction party is not capable of generating the signed digital signature without the private key. The second transaction party only stores the signed digital signature.

The first transaction party may later claim not to have performed the transaction and not to have generated the digital signature, and may thus claim that another party has maliciously presented itself as the first transaction party while being able to generate a corresponding digital signature. The second transaction party may then verify the signed digital signature by sending it to the third party, which may verify the signed digital signature. Based on the signed digital signature it may be determined whether a malicious user has performed the transaction or the first transaction party is maliciously claiming not to have performed the transaction.

Fig. 8 illustrates a method to protect digital files from being used on other devices than the originally intended device. For illustrative purposes, an application that is digitally signed according to the present invention is described in relation to Fig. 6. However, the method may also be used for any other signed digital file such as a file containing music or video.

According to cell 100, an application having a digital signature embedded therein is being started on a device having the authentication software installed in its BIOS. The embedded digital signature has been generated on said device and is therefore device specific. For example, the application may have been obtained using the method discussed in relation to Fig. 7. In any case, the application has been signed and programmed to be used on the specific device having the authentication software installed.

The application is programmed to start with a verification of its embedded digital signature. Therefor, the application requests the BIOS to verify its digital signature according to cell 102. For the BIOS to be able to verify the digital signature, it needs the data components that were used to generate the digital signature. So, according to cell 104, the application transfers the data

components to the BIOS. After receipt of all necessary data from the application, the BIOS regenerates the digital signature according to cell 106.

For verification, it is needed to compare the embedded digital signature with the regenerated digital signature. This comparison may be conducted by the authentication software in the BIOS or the BIOS. Thus, according to cell 108, the BIOS receives and compares the embedded digital signature with the regenerated digital signature.

If the two digital signs are identical, the application starts according to cell 110, otherwise the BIOS prevents the application being started, possibly informing the user that it is trying to use an illegal copy of the application.

Fig. 9 illustrates an on-line transaction method. An on-line application is accessed with a request for a transaction, possibly a financial transaction. For example, the transaction may be a customer who wishes to purchase an item on-line. Usually, such transactions are performed using a credit card. However, only a credit card number and some additional data are supplied by the customer. If a customer later states he did not purchase or use his credit card, there is no way of verifying by checking a signature, like in common credit card transactions.

Referring to Fig. 9, according to cell 120, the on-line application receives a request for a transaction. Upon this request, the on-line application asks the BIOS of the requesting device if the authentication software according to the present invention is installed and running. If not, the requesting device is rerouted to a TTP to download and install the authentication software and authentication table according to the method discussed in relation to Fig. 2 or Fig. 4.

If the authentication software is installed and running, the BIOS responds accordingly according to cell 124 and the on-line application hands over a transaction ID according to cell 126. The requesting device may then generate a digital signature according to the method described in relation to Fig. 6, according to cell 128. The variable data components used for generating the digital signature and the digital signature itself are then transferred to

the on-line application, which may then complete the transaction according to cell 130.

The digital signature functions as a signature in this case. If the customer later claims not to have used its credit card, for example, the on-line application owner may verify the digital signature. The verification of the digital signature may be conducted by the trusted third party, which supplied the authentication table to the customer and is able to regenerate said authentication table. It may also be conducted by using the customer device regenerating a digital signature using the variable data components stored by the on-line application.

Optionally, a certain on-line third party application may use an authentication table in transactions with a specific device, which authentication table is specifically intended for use in transactions with said certain on-line third party application. Such a specific authentication table has the advantage that the third party application may verify the digital signature already before completing the transactions and so preventing that problems may later arise.

The specific authentication table may be derived from the authentication table stored in the BIOS of said device using an algorithm known to the device. The specific authentication table derived from the original authentication table would then be supplied by the TTP to the on-line application. Thus, the device knows how to derive the specific authentication table and the on-line third party application knows a certain table, but not the original table stored in the BIOS of the device. Otherwise, the on-line third party may provide a separate authentication table to the requesting device. The specific authentication table is encrypted before it is stored using a BIOS specific encryption key. This ensures that the supplying third party does not know the stored authentication data and therefore the third party can not use the data in a fraudulent way.

If the on-line application knows the authentication table used by the requesting device to generate a digital signature, the requesting device may be a device connecting to a network, e.g. a corporate private network. The requesting device needs to be

identified by the corporate network before access to the corporate network is granted. Upon connection, the requesting device sends a digital signature that was generated using fixed data components and variable data components. The fixed data components may comprise a password or a personal identification number (PIN). The fixed data components are known to the corporate network and do not need be sent over the unsecured network connection; only the variable data components used are sent over the unsecured connection. Thus, the corporate network may regenerate the digital signature of the connecting device after receipt of the used variable data components. This method has the advantage that no password or PIN is sent over an unsecured connection and that the digital signature identifies both the device (authentication table) and the user (PIN or password).

Apart from securing a transaction or network authentication as illustrated above, also the hardware used in the operating environment may be verified. For example, if a first computer communicates with a second computer, the second computer may identify itself in a similar way, using similar data, as used in a transaction, described above. If the authentication table of the second computer is known at the first computer, the identification of the second computer may be verified by the first computer. A similar procedure is feasible when one of the computers communicates with a printer, as long as the authentication tables are stored securely in the devices.

Fig. 10 illustrates a method for authenticating a user to access digital data that are stored in a removable and portable memory such as a flash memory. Digital data may be stored in the removable memory using the encryption method for protecting (authentication) data as illustrated in and described in relation to Fig. 4, i.e. data are encrypted using at least one, preferably at least two encryption layers, of which one layer is related to the authentication software installed according to Fig. 4. The removable memory is to be connected with an electronic device on which the authentication software has been installed according to said method illustrated in Fig. 4 in order to decrypt the encryption layer related to said authentication software.

Referring to Fig. 10, digital data are stored in a removable and portable memory. The digital data are encrypted on another electronic device, for example a computer device at work, while the electronic device attempting to access the digital data is a computer device at home.

The authentication software for accessing the digital data is installed on both computer devices, e.g. at home and at work, according to the installation method of Fig. 4. The digital data are to be read by a user application such as a text editor, a spreadsheet application, an audio or video application, or the like. In another case, the digital data may be a software application to be executed by the electronic device.

After connecting the portable and removable memory with the electronic device, a user starts the user application. From the user application, an attempt is made to access the digital data according to cell 220. The digital data are however encrypted.

Then, according to cell 222, the authentication software is started to decrypt the digital data. The digital data are transferred to the authentication software. In accordance with cell 224, the authentication software decrypts a first encryption layer using a decryption algorithm, for example embedded in the authentication software. Said decryption algorithm is identical on each device that is provided with the authentication software.

To prevent that any device being provided with the authentication software may decrypt the digital data, the digital data may be protected with a second encryption layer. For decryption of the second encryption layer, a decryption key is required. Such a decryption key may be associated with a serial number of the portable memory, ensuring that the digital data are only decryptable if the digital data are stored in said portable memory. A copy of the digital data in any other memory is thereby rendered undecryptable. Likewise, the decryption key may be associated with a personal identification number (PIN) rendering the digital data unusable without the PIN. In yet another embodiment, the decryption key may be associated with a fingerprint of a user, rendering the digital data unusable without the presence of that specific user.

The decryption key may be associated with any identifying characteristic such as a serial number, a PIN or a fingerprint by a protected hashing algorithm or it may be identical to the serial number or PIN. Hashing the characteristic provides however an  
5 additional security layer, especially when the hashing is performed in the secure processing environment.

According to cell 226, the data for generating the decryption key are collected and the decryption key is generated. Then, in accordance with cell 228, the second encryption layer is decrypted.  
10 The digital data are now suitable for access by the user application. Therefore, in cell 230, the digital data are provided to the user application and in cell 232, the user application accesses and uses the digital data. After use and possibly alteration of the digital data, the digital data are again stored in  
15 the removable memory using the encryption algorithm of the authentication software and using the decryption key used to access the digital data.

Above it is described that the decryption algorithm is identical on each device that is provided with the authentication  
20 software. However, the decryption/encryption algorithm may also be different for each device or it may be identical for each device that is installed using an authentication software package obtained by a specific user. Thus, the digital data may only be accessible on a device that is previously installed by said specific user.

25 In such a case, it is necessary to provide certain authentication data to the transaction party supplying said authentication software, since the user specific algorithm needs to be embedded in the software package before the software package is supplied to the specific user in order to protect the algorithm.

30 In an even further embodiment, digital data may be encrypted using an encryption key that is generated according to the present invention, i.e. identical to the method for generating a digital signature. Fig. 6 illustrates how a digital signature may be generated according to the present invention by gathering session  
35 specific data, such as fixed data, variable data, personal data and/or device specific data. Hashing said session specific data may generate reference numbers, referring to positions in the



authorization table securely stored according to the present invention. Gathering the characters stored in the authorization table at said positions generates a bit string comprising a number of characters. Said bit string may comprise any number of characters and said number may be dependent on the session specific data. Data encrypted with an encryption key having an unknown number of bits is virtually impossible to be cracked by another person not entitled to access said data.

To decrypt the data, the encryption key is required. To obtain the encryption key, the authorization table and the session specific data are needed. Having an authorization table stored and installed in a device according to the present invention, data may be securely stored in a memory of said device. The data may easily be decrypted when being accessed using said device. However, a copy of said encrypted data on any other device is rendered virtually inaccessible.

If identical authorization tables are stored on two or more separate devices, encrypted data may be exchanged between said two or more devices. If the encrypted data are transferred from one device to another, together with the session specific data, the other device may regenerate the encryption key and decrypt the data. Such an encryption and decryption method is especially useful for secure communication between said two or more devices over a publicly accessible network such as the Internet.

In an embodiment, a network server is provided with all authorization tables of client devices connected to said server. A client attempting to access the server and the network of said server is authenticated by its digital signature, and thereafter all exchanged data may be encrypted using a digital signature. If a client sends data to another client, the data may be encrypted and transferred to the server together with the session specific data. The server decrypts the data using the session specific data and the authorization table of the first client. Then, the server encrypts the data again, now using the authorization table of the other client using the same or other session specific data. Next, the encrypted data are transferred to the other client together with the

corresponding session specific data, which decrypts the data using its authorization table.

Now referring to Fig. 10 again, the digital data securely stored on the portable and removable memory device may be an authorization table according to the present invention. Thus,  
5 digital data encrypted using the authorization table may be decrypted on any device when the portable and removable memory device is connected to said device and said device is provided with the authorization software according to Fig. 10.

## CLAIMS

1. A method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party, the electronic device having an operating system creating a run-time environment for user applications, the method comprising:

providing authentication data in a memory of said electronic device, said memory being a secure part of a Basic In Out System or any other secure location in said electronic device, which authentication data are inaccessible to a user of said electronic device, wherein the authentication data are encrypted when the authentication data are stored in said memory, a decryption key for decrypting the authentication data being incorporated in the electronic device, said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device, thereby rendering the authentication data inaccessible to said user;

providing authentication software in said electronic device, the authentication data being accessible to said authentication software, wherein authentication software is stored in said secure memory inaccessible to said operating system;

activating the authentication software to generate a digital signature from the authentication data, wherein the authentication software is run in a secure processing environment inaccessible to said operating system;

providing the digital signature to the second transaction party.

2. The method according to claim 1 , wherein the authentication data are provided by the second transaction party, which stores the authentication data together with data identifying the first transaction party.

3. The method according to claim 2, wherein the second transaction party uses the stored authentication data to obtain transaction specific authentication data according to a specific algorithm.

4. The method according to claim 3, wherein the second transaction party verifies the digital signature provided by the first transaction party using the authentication data stored at the second transaction party.

5. The method according to claim 1, wherein the authentication software has its own unique serial number, software ID and/or private encryption key.

6. The method according to claim 1, wherein the authentication data are encrypted by the second transaction party using an encryption key before the authentication data are provided to the first transaction party.

7. The method according to claim 6, wherein the authentication software retrieves a decryption key associated with the encryption key and decrypts the authentication data at its first use.

8. The method according to claim 1, wherein the authentication software is installed in an application environment in the secure area such that it may obtain the authentication data without passing through an unsecured part of said electronic device.

9. The method according to claim 1, wherein the authentication data are encrypted using at least two encryption layers.

10. The method according to claim 1, wherein the

authentication data are encrypted using an encryption key which is associated with a hardware device serial number.

11. The method according to claim 1, wherein the authentication data are encrypted using an encryption key which is associated with a user identifying number, in particular a personal identifying number, PIN, or a number or a template associated with a fingerprint of the user.

12. The method according to claim 1, wherein the decryption key is associated with one or more serial numbers of hardware components of said electronic device.

13. The method according to claim 1, wherein the authentication data is decrypted by the authentication software.

14. The method according to claim 1, wherein the authentication data is decrypted by an application stored in the electronic device using a device specific encryption key.

15. The method according to claim 1, wherein the authentication data is decrypted using a decryption key associated with any identifying characteristic, in particular a serial number, a personal identification number, PIN, or a fingerprint of a user.

16. The method according to claim 11, wherein the authentication data are decrypted in a secure processing environment inaccessible to said user and to any user-operated software.

17. The method according to claim 1, wherein the authentication data comprise an authentication table.

18. The method according to claim 17, wherein the authentication table is generated from a bit string which is generated from fixed data and variable data.

19. The method according to claim 18, wherein the variable data comprise a random table.

20. The method according to claim 19, wherein the random table is calculated from a random two dimensional or three-dimensional pattern.

21. The method according to claim 1, further comprising identifying the user of the electronic device before activating the authentication software.

22. The method according to claim 1, wherein the authentication software has a private encryption key for encrypting digital data.

23. A system for performing an electronic transaction between a first transaction party and a second transaction using an electronic device operated by the first transaction party, the electronic device having an operating system creating a run-time environment for user applications, the system comprising:

means for providing authentication data in a memory of said electronic device, , said memory being a secure part of a Basic In Out System or any other secure location in said electronic device, which authentication data are inaccessible to a user of the electronic device, , wherein the authentication data are encrypted when the authentication data are stored in said memory, a decryption key for decrypting the authentication data being incorporated in the electronic device, said decryption key being

inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device, thereby rendering the authentication data inaccessible to said user;

means for providing authentication software in said electronic device, the authentication data being accessible to said authentication software, wherein the authentication software is stored in a secure memory location inaccessible to said operating system;

means for activating the authentication software to generate a digital signature from the authentication data, wherein the authentication software is run in a secure processing environment inaccessible to said operating system;

means for providing the digital signature to the second transaction party; and

means for providing digital data from the second transaction party to the first transaction party.

24. An electronic device for performing an electronic transaction between a first transaction party, and a second transaction party, wherein the electronic device is operated by the first transaction party, the electronic device having an operating system creating a run-time environment for user applications, the electronic device comprising:

a memory storing authentication data , said memory being a secure part of a Basic In Out System or any other secure location in said electronic device, which authentication data are inaccessible to a user of the electronic device, , wherein the authentication data are encrypted when the authentication data are stored in said memory, a decryption key for decrypting the authentication data being incorporated in the electronic device, said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said

memory is inaccessible to said operating system of said electronic device, thereby rendering the authentication data inaccessible to said user;

a memory storing authentication software, the authentication data being accessible to said authentication software, wherein the authentication software is stored in a secure memory location inaccessible to said operating system;

means for activating the authentication software to generate a digital signature from the authentication data, wherein the authentication software is run in a secure processing environment inaccessible to said operating system; and

means for providing the digital signature to the second transaction party.

25. The electronic device according to claim 24, wherein the electronic device is a personal computer, a cellular phone, a hand-held personal digital assistant with wireless communication capabilities, or a device containing a BIOS.



## ABSTRACT

A method for performing an electronic transaction is disclosed. The method provides authentication data and authentication software to an electronic device and preferably stored in a secure storage location or other location inaccessible to the user or the operating system of the device. The authentication software is activated to generate a digital signature from the authentication data. Next, the digital signature is provided to the other transaction party.

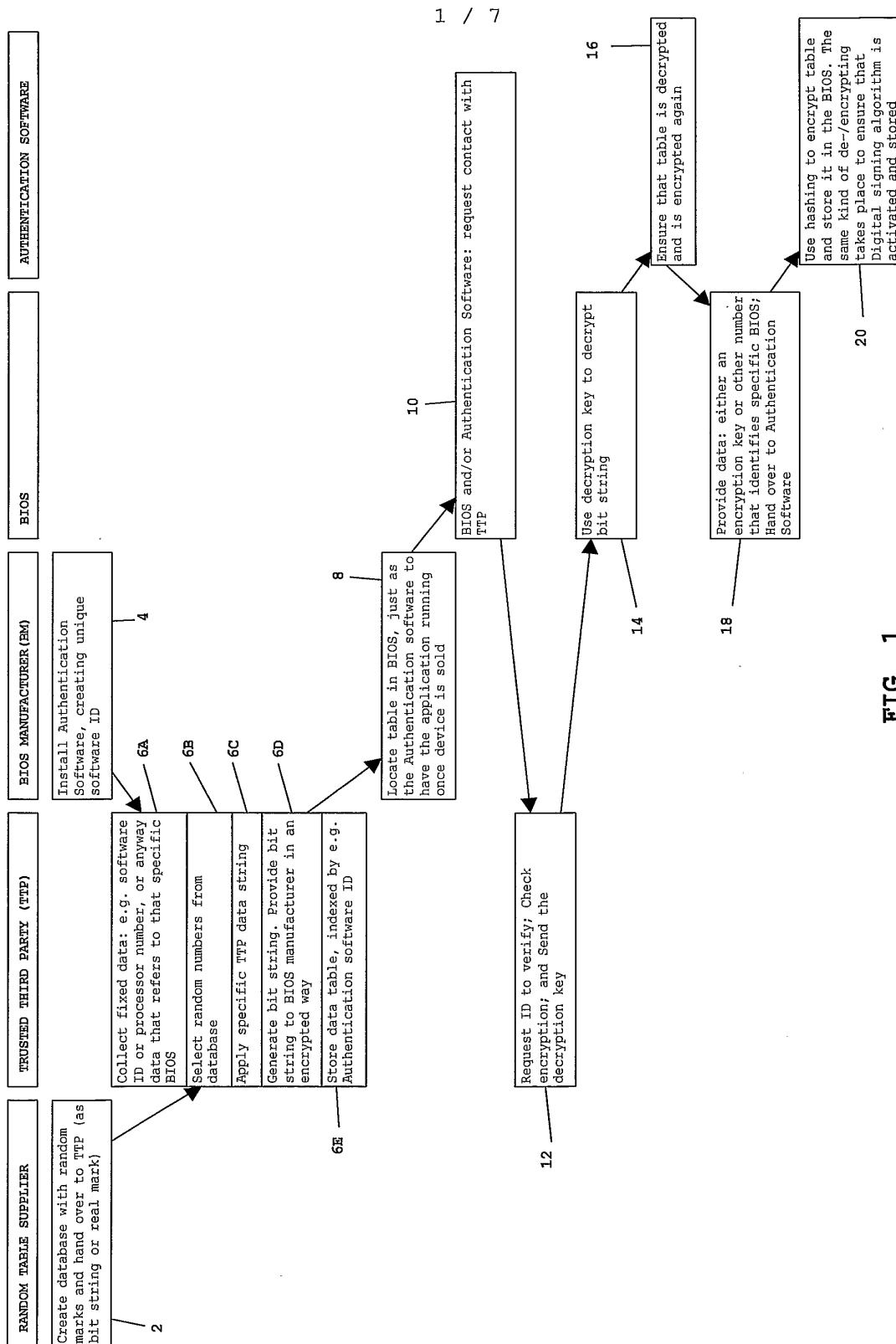


FIG. 1

2 / 7

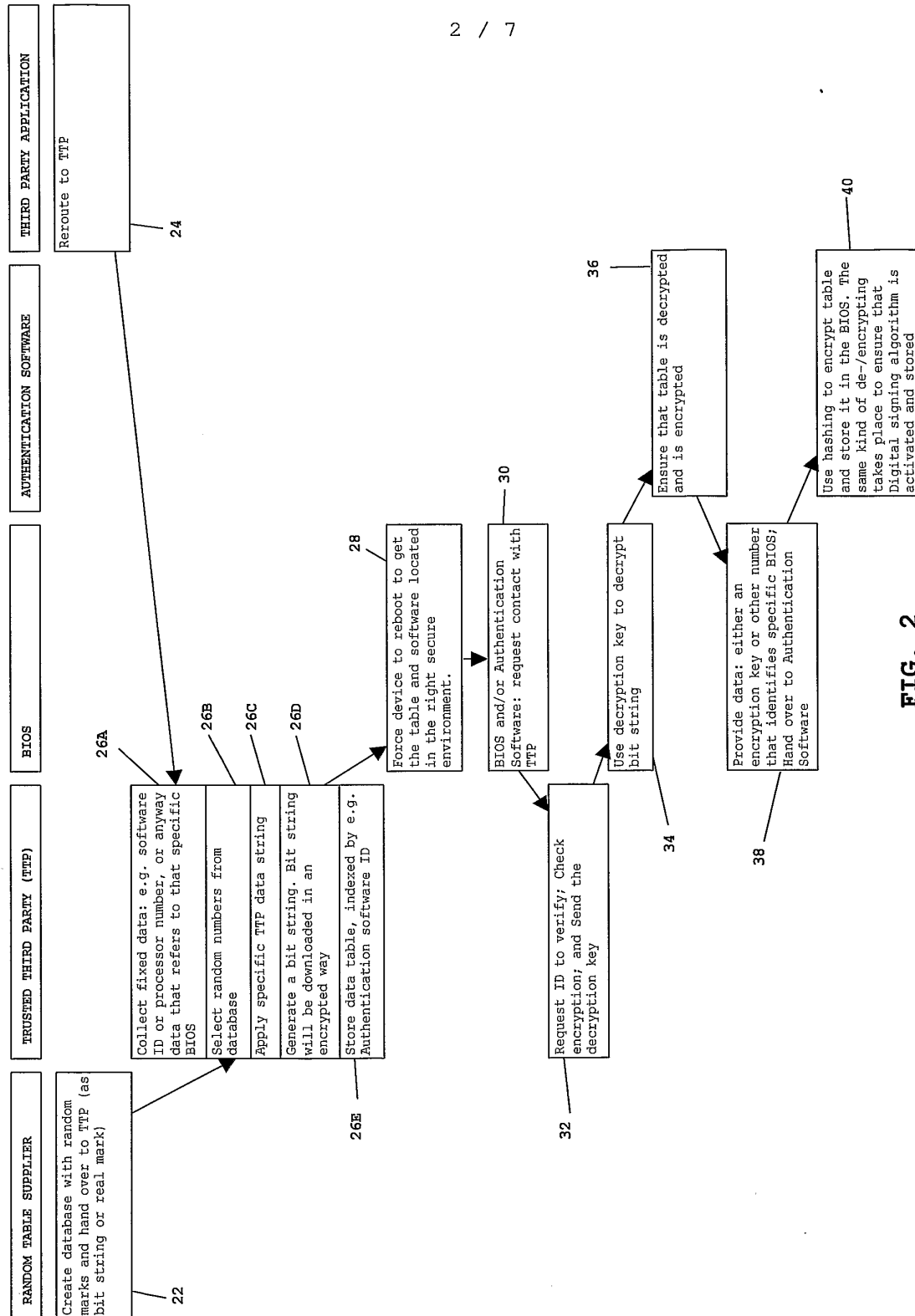


FIG. 2

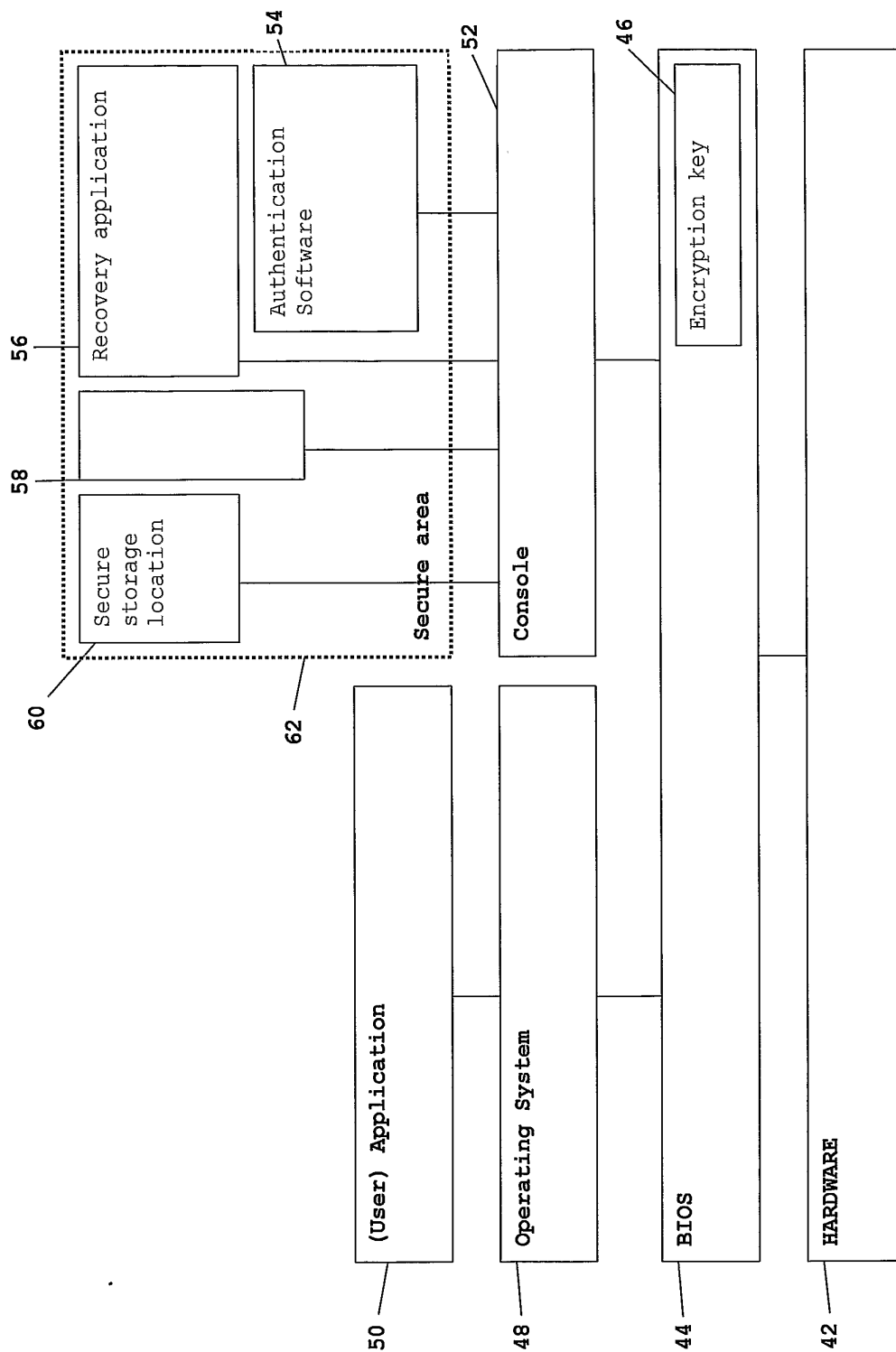


FIG. 3

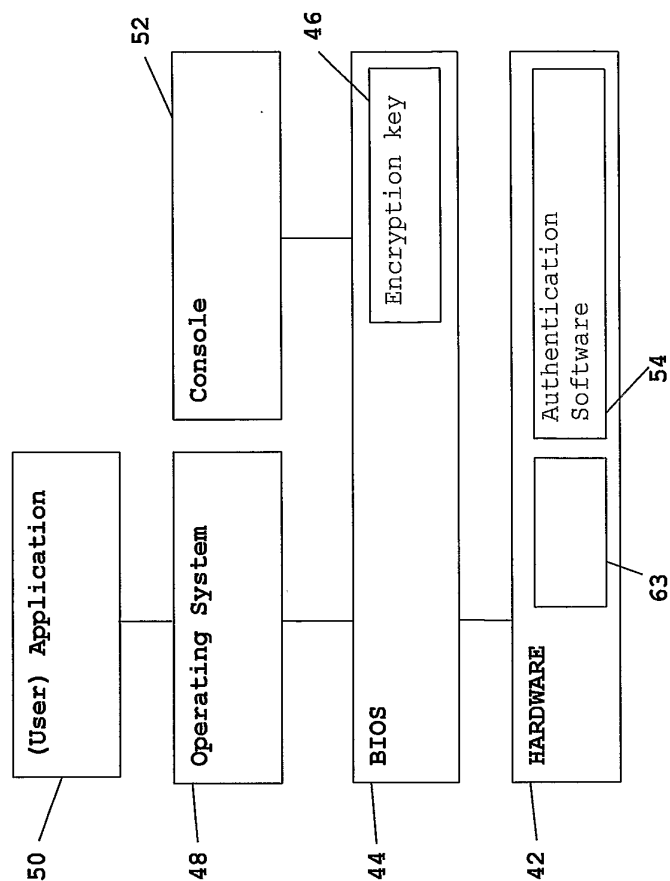


FIG. 5A

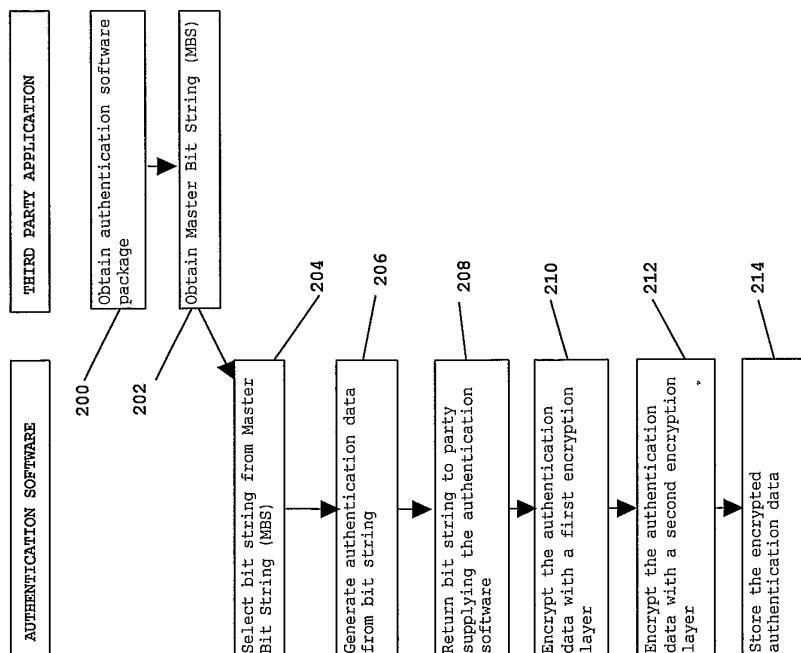


FIG. 4

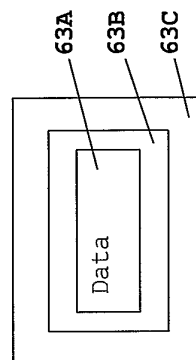


FIG. 5B

5 / 7

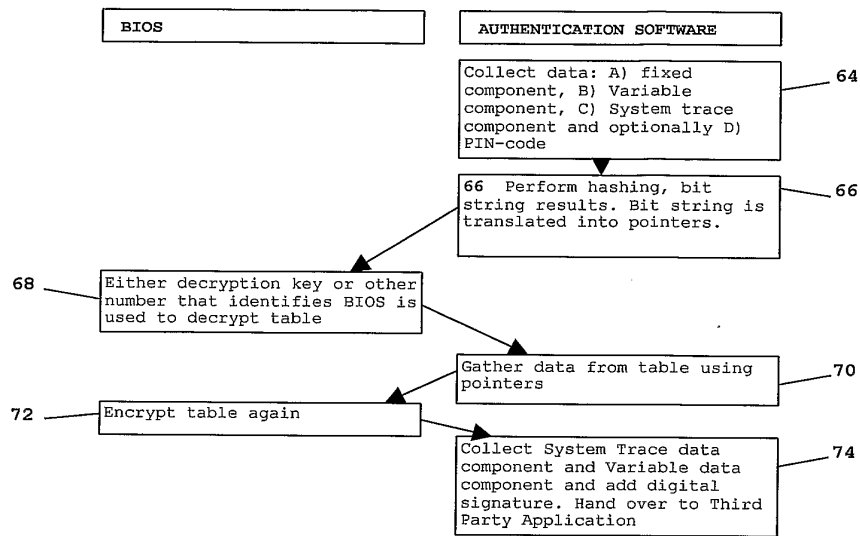


FIG. 6

6 / 7

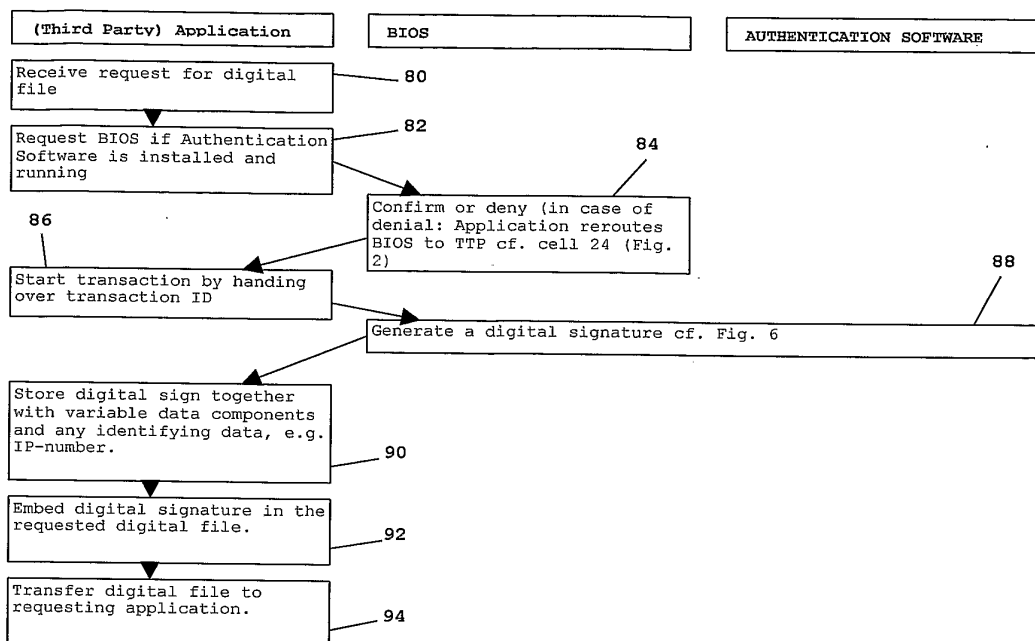


FIG. 7

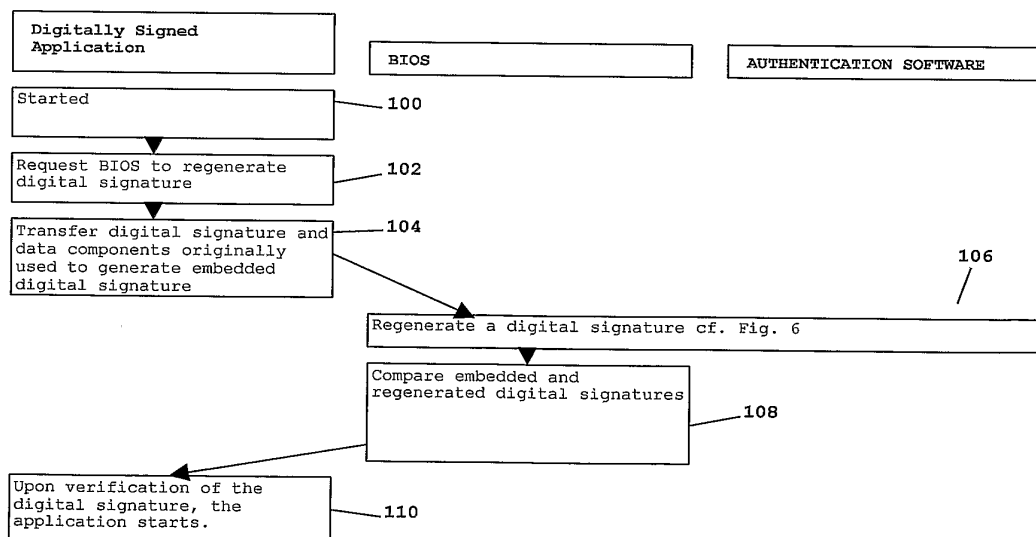


FIG. 8

7 / 7

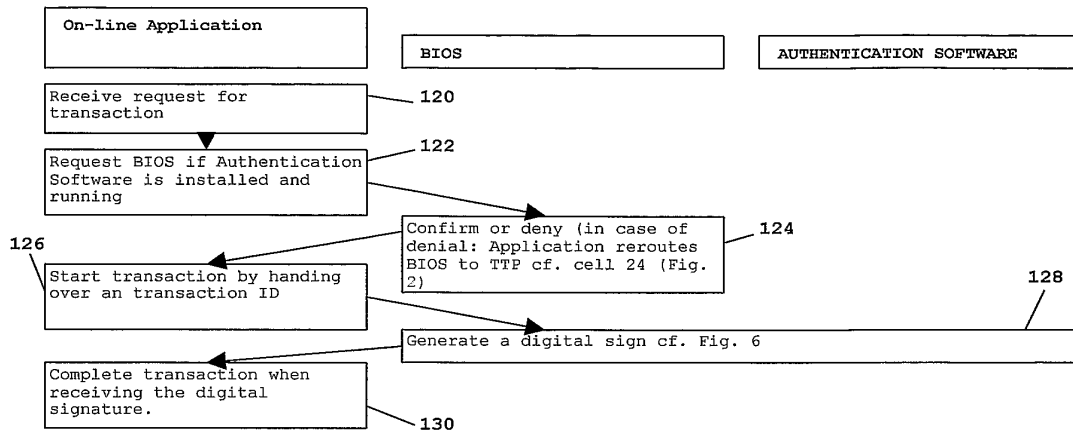


FIG. 9

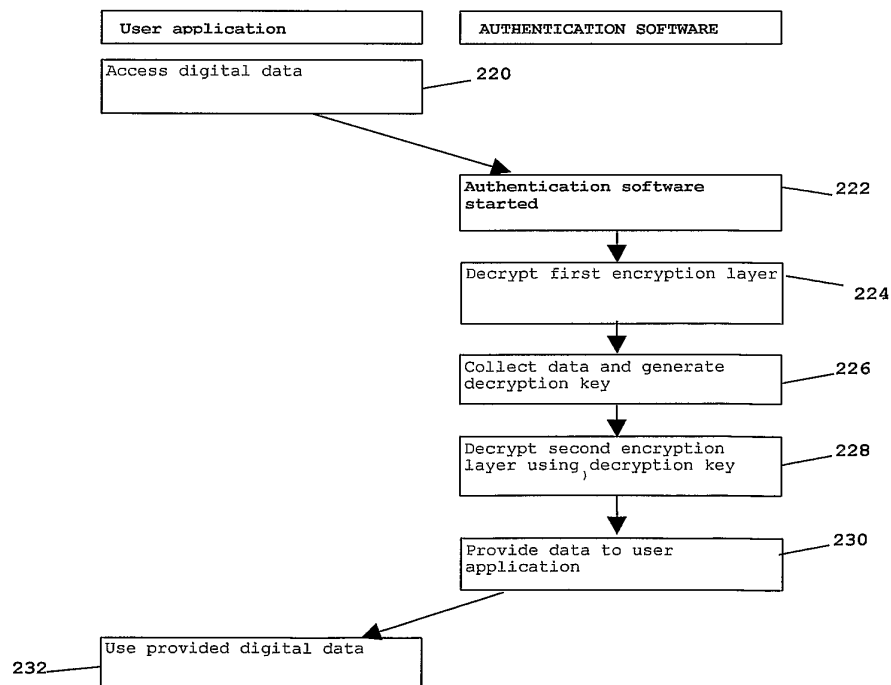
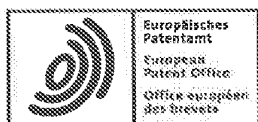


FIG. 10





European Patent Office  
80298 MUNICH  
GERMANY

Questions about this communication ?  
Contact Customer Services at [www.epo.org/contact](http://www.epo.org/contact)



DeltaPatents B.V.  
Fellenoord 370  
5611 ZL Eindhoven  
PAYS-BAS

ONTVANGEN 08 AUG 2018

Date
02.08.18

Reference 330.002 EPw2	Application No./Patent No. 17000713.2 - 1218 / 3229099
Applicant/Proprietor Ward Participations B.V.	

**Decision to grant a European patent pursuant to Article 97(1) EPC**

Following examination of European patent application No. 17000713.2 a European patent with the title and the supporting documents indicated in the communication pursuant to Rule 71(3) EPC (EPO Form 2004C) or in the information (EPO Form 2004W, cf. Notice from the EPO dated 8 June 2015, OJ EPO 2015, A52) dated 20.06.18 is hereby granted in respect of the designated Contracting States.

Patent No. : 3229099  
Date of filing : 14.06.04  
Priority claimed : 13.06.03/NLW 0300436

Designated Contracting States  
and Proprietor(s) : AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC  
NL PL PT RO SE SI SK TR  
Ward Participations B.V.  
1, De Schaepstal  
1251 MZ Laren/NL

This decision will take effect on the date on which the European Patent Bulletin mentions the grant (Art. 97(3) EPC).

The mention of the grant will be published in European Patent Bulletin 18/35 of 29.08.18.

Examining Division

Kopp K

Juroca A

Prins L



Registered letter  
EPO Form 2008A 07.15 (26/07/18)

to EPO postal service: 27.07.18  
page 1 of 1

ANMERKUNG ZUR ENTSCHEIDUNG ÜBER DIE ERTEILUNG  
EINES EUROPÄISCHEN PATENTS (EPA Form 2006)

1. **EPA Informationsbroschüre "Nationales Recht zum EPÜ"**  
Diese Broschüre enthält nützliche Informationen zu den formalen Erfordernissen und den Handlungen, die vor den Patentbehörden der Vertragsstaaten vorzunehmen sind, um Rechte in diesen Staaten zu erlangen. Da diese Handlungen einem ständigen Wandel unterworfen sind, sollte immer nur die neueste Ausgabe der Broschüre benutzt werden. Nachträgliche Informationen werden im Amtsblatt veröffentlicht.
2. **Übersetzung der europäischen Patentschrift nach Artikel 65 (1) des Europäischen Patentübereinkommens**  
Sie werden erneut darauf hingewiesen, dass bestimmte Vertragsstaaten nach Artikel 65 (1) EPÜ eine Übersetzung der europäischen Patentschrift verlangen; hierauf wird in der Mitteilung gemäß Regel 71 (5) EPÜ verwiesen. Die Nichteinreichung dieser Übersetzung kann zur Folge haben, dass das Patent in dem betreffenden Staat/in den betreffenden Staaten als von Anfang an nicht eingetreten gilt. Weitere Einzelheiten entnehmen Sie bitte der oben genannten Broschüre.
3. **Zahlung von Jahresgebühren für europäische Patente**  
Nach Artikel 141 EPÜ können "nationale" Jahresgebühren für das europäische Patent für die Jahre erhoben werden, die an das Jahr anschließen, in dem der Hinweis auf die Erteilung des europäischen Patents im "Europäischen Patentblatt" bekanntgemacht wird. Weitere Einzelheiten entnehmen Sie bitte der oben genannten Broschüre.

NOTE RELATING TO THE DECISION TO GRANT A  
EUROPEAN PATENT (EPO Form 2006)

1. **EPO information Brochure "National law relating to the EPC"**  
This brochure provides useful information regarding formal requirements and the steps to be taken before the patent authorities of the Contracting States in order to acquire rights in those states. Since the necessary steps are subject to change the latest edition of the brochure should always be used. Subsequent information is published in the Official Journal.
2. **Translation of the European patent application under Article 65(1) of the European Patent Convention**  
Your attention is again drawn to the requirements regarding translation of the European patent specification laid down by a number of Contracting States under Article 65(1) EPC, to which reference is made in the communication under Rule 71(5) EPC. Failure to supply such translation(s) may result in the patent being deemed to be void "ab initio" in the State(s) in question. For further details you are recommended to consult the above-mentioned brochure.
3. **Payment of renewal fees for European patents**  
Under Article 141 EPC "national" renewal fees in respect of a European patent may be imposed for the years which follow that in which the mention of the grant of the European patent is published in the "European Patent Bulletin". For further details you are recommended to consult the above-mentioned brochure.

REMARQUE RELATIVE A LA DECISION DE DELIVRANCE  
D'UN BREVET EUROPEEN (OEB Form 2006)

1. **Brochure d'information de l'OEB "Droit national relatif à la CBE"**  
Cette brochure fournit d'utiles renseignements sur les conditions de forme requises et sur les actes à accomplir auprès des offices de brevet des Etats contractants aux fins d'obtenir des droits dans les Etats contractants. Etant donné que les actes indispensables sont susceptibles de modifications, il serait bon de toujours consulter la dernière édition de la brochure. Toute information ultérieure est publiée au Journal Officiel.
2. **Traduction du fascicule du brevet européen en vertu de l'article 65(1) de la Convention sur le brevet européen**  
Votre attention est de nouveau attirée sur l'obligation faite par certains Etats contractants, en vertu de l'article 65(1) CBE, de fournir une traduction du fascicule du brevet européen, à laquelle il est fait référence dans la notification établie conformément à la règle 71(5) CBE. Si la(les) traduction(s) n'est(ne sont) pas fournie(s), le brevet européen peut, dès l'origine, être réputé sans effet dans cet(ces) Etat(s). Pour plus de détails, nous vous renvoyons à la brochure susmentionnée.
3. **Paiement des taxes annuelles pour le brevet européen**  
Conformément à l'article 141 CBE des taxes annuelles "nationales" dues au titre du brevet européen peuvent être perçues pour les années suivant celle au cours de laquelle la mention de la délivrance du brevet européen est publiée au "Bulletin européen des brevets". Pour plus de détails, nous vous renvoyons à la brochure susmentionnée.

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA		
First Named Inventor/Applicant Name:		Scott MacDonald WARD		
Filer:		David Joseph Kenealy/Yoshi Hassan		
Attorney Docket Number:		5061-0041CON		
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
<b>Basic Filing:</b>				
UTILITY APPLICATION FILING	1011	1	300	300
UTILITY SEARCH FEE	1111	1	660	660
UTILITY EXAMINATION FEE	1311	1	760	760
<b>Pages:</b>				
<b>Claims:</b>				
<b>Miscellaneous-Filing:</b>				
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
<b>Post-Allowance-and-Post-Issuance:</b>				
<b>Extension-of-Time:</b>				
<b>Miscellaneous:</b>				
<b>Total in USD (\$)</b>				<b>1720</b>

Electronic Acknowledgement Receipt	
<b>EFS ID:</b>	37415191
<b>Application Number:</b>	16597773
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	8373
<b>Title of Invention:</b>	METHOD AND SYSTEM FOR PERFORMING A TRANSACTION AND FOR PERFORMING A VERIFICATION OF LEGITIMATE ACCESS TO, OR USE OF DIGITAL DATA
<b>First Named Inventor/Applicant Name:</b>	Scott MacDonald WARD
<b>Customer Number:</b>	39083
<b>Filer:</b>	David Joseph Kenealy/Yoshi Hassan
<b>Filer Authorized By:</b>	David Joseph Kenealy
<b>Attorney Docket Number:</b>	5061-0041CON
<b>Receipt Date:</b>	09-OCT-2019
<b>Filing Date:</b>	
<b>Time Stamp:</b>	20:10:02
<b>Application Type:</b>	Utility under 35 USC 111(a)

### Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$1720
RAM confirmation Number	E201909K11283072
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	POA_WARD_PARTICIPATIONS_BV.pdf	157272	no	1
			8a09208c6526e0a4830fa40f069581f4def3e91e		
Warnings:					
Information:					
2	Application Data Sheet	10-09-19_ADS.pdf	1256250	no	8
			0973d946f1942662a94f53c5ad5e401b6aa b5e6c		
Warnings:					
Information:					
3	Oath or Declaration filed	Executed_Declaration.pdf	301135	no	2
			2a4a64f2875a0aa6d5b128820ce3719f86361e3e		
Warnings:					
Information:					
4		10-09-19_Specification.pdf	1599298	yes	36
			0271c106ed46992f373b21165dad904c5d3 9d747		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Specification		1	29	
	Claims		30	35	
	Abstract		36	36	
Warnings:					
Information:					
5	Drawings-only black and white line drawings	10-09-19_Drawings.pdf	154023	no	7
			ac48802fa96d1a150ea38878116bd277a70 0471b		

<b>Warnings:</b>					
<b>Information:</b>					
6	Petition to make special under Patent Prosecution Hwy	10-09-19_PPH_Request.pdf	1246625	no	3
			0145b3172748918495daa0e81fc90a2793f26632		
<b>Warnings:</b>					
<b>Information:</b>					
7	Petition to make special under Patent Prosecution Hwy	180802_330_002EPw2_fPTO_Decision_to_Grant.pdf	149430	no	2
			d660c804b0eb1e91f6e108de1beb9f2c60eb620e		
<b>Warnings:</b>					
<b>Information:</b>					
8	Fee Worksheet (SB06)	fee-info.pdf	34999	no	2
			5a8593e31c9866238433e45375ad0f02c00bd3f8		
<b>Warnings:</b>					
<b>Information:</b>					
<b>Total Files Size (in bytes):</b>			4899032		
<p>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.</p> <p><b><u>New Applications Under 35 U.S.C. 111</u></b>  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.</p> <p><b><u>National Stage of an International Application under 35 U.S.C. 371</u></b>  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.</p> <p><b><u>New International Application Filed with the USPTO as a Receiving Office</u></b>  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.</p>					