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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 November 2006 (16.11.2006)

PCT

(10) International Publication Number
WO 2006/120365 A1

(51) International Patent Classification:
G06F 1/00 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/GB2005/001770

(22) International Filing Date: 10 May 2005 (10.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/IB2004/050628 10 May 2005 (10.05.2005) EP

(71) Applicants and
(72) Inventors: **GIRGIS, Hani** [EG/EG]; 52 Youssef Street, Helwan, Cairo, 11421 (EG). **ISKANDER, Nader** [GB/EG]; EME International, 11 El Masgued El Aqsa street, Mohandeseen, Cairo, 12411 (EG).

(74) Agents: **BARTON, Russel** et al.; Withers & Rogers LLP, Goldings House, 2 Hays Lane, London SE1 2HW (GB).

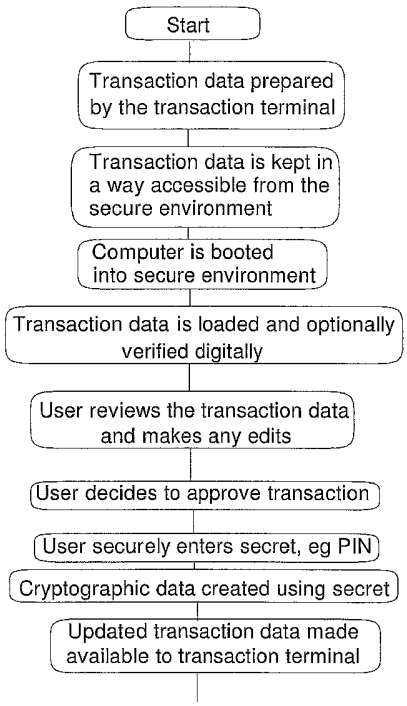
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK,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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

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

(54) Title: SECURE TRANSACTIONS USING A PERSONAL COMPUTER



(57) Abstract: A transaction terminal and a process which allows a user to make secure transactions, such as PIN-based transactions, using his personal computer. The steps are: (a) Preparing transaction data (b) Storing it in non-volatile memory, (c) Restarting or hibernating the computer, (d) booting into a secure, un-networked, environment from a bootable media or device; this bootable media or device must be sufficiently difficult to counterfeit and sufficiently difficult to tamper with the data stored in it and optionally difficult to copy, (e) Securely launching the secure PIN entry software, (f) Loading transaction data from the non-volatile memory, (g) Presenting the transaction data to the user and optionally allowing the user to modify and/or complete it, (h) Secure PIN entry resulting in an encrypted PIN block and/or enabling the user and/or the generation of the appropriate keys for creating message authentication codes(s) and/or cryptogram(s) and/or digital signature(s) according to the transaction security standards; The user can also enter a password to enable secure access to password encrypted secret keys, private keys and confidential data; The user can also enter secure information to update his records in the server side system, like a user choosable CVV2/CVC2 or new 3D Secure password, (i) Storing the secured transaction request in non-volatile memory, (j) Restarting the computer back to normal operation, (k) Loading the secured transaction request from non-volatile memory, (i) Sending the transaction for authorisation, (m) Receiving the response, (n) Presenting the response to the user and optionally storing it. The process radically protects the user from any malicious software that might affect the security of PIN entry; it dramatically reduces the user responsibilities to physical security considerations only, like those in ATM transactions. The user should use a personal computer that he knows that it does not contain malicious hardware; this could easily be his own notebook or PC at home.

WO 2006/120365 A1

SECURE TRANSACTIONS USING A PERSONAL COMPUTER

Technical Field

The invention relates to secure transactions such as PIN based transactions, where the Personal Identification Number known as PIN, is used to authenticate the user. An ATM transaction is an example of PIN based transactions. In particular, but not exclusively, the invention relates to electronic transactions that are made on a computer trusted by the user to be physically secure yet not trusted to be logically secure due to the possibility of malicious software that may exist on that computer. The electronic transaction itself can be of any kind, including but not limited to making orders, payments, voting, stocks trading ...etc.

Any personal a PC or notebook at home or in office is usually trusted by the user to be physically secure yet not trusted to be logically secure due to the possibility of malicious software that may exist on that computer.

The user can normally validly assume that no malicious hardware, designed for example to steal his confidential data, is implanted on his computer, but he can never be sure that there is no hidden malicious software, sent by a hacker for example, is actually running on his computer. Statistics show that most PCs connected to the Internet are eventually infected with critical malicious software, like spyware and Trojan horses.

Background of the invention

In this field, a PIN Entry Device, PED, is a device used in securing the entry and processing of the PIN of the user. It needs to have a keypad, a display, storage and processing capabilities. The keypad-like device found beside merchant POS machines for the use by the cardholder to enter his PIN is an example of a PED. A transaction terminal is the system used to initiate the transaction. It should have a PED if PIN-based transactions are to be supported on this transaction terminal.

In the art of Personal Computer security various problems exist in preventing secure transactions including spying software, Trojan horses and viruses. While, ecommerce

purchases are done electronically over the Internet there is no known secure way of enabling a user to enter a PIN, which may be valid also for other types of transaction such as in controlled networks of ATMs and hence other than the internet, to authenticate an ecommerce transaction. It is also known to use smartcards in making secure transactions, to make card payments, where a payment card is used to make financial transactions, and to enable secure transactions using digital signature by generating and verifying digital signatures on transactions, but all of these known techniques have limitations when applied to potential use by a home user using a personal computer.

PIN based transactions enjoy very high security due to the very well established technical standards that support it. Also the market has responded by implementing the standards and making available Hardware Security Modules, HSMs, for physically securing the server side part of PIN-based transactions and PIN Entry Devices, PEDs, for securing the client side part. In most critical environments, like banks and governments, only certified HSMs and PEDs are used. One result of this well established field is the possibility of assuming non-repudiation and putting the liability of the PIN on the user. This is because the whole system for PIN based transactions is based on well established standards of well proven security that need not to be verified every time there is a dispute, the only thing needed is to verify that the standards are taking place in the system. The PIN entered in PIN based transactions is either a PIN to be verified against an offset in the server side, or it is a PIN that unlocks the access to a smartcard, which would create a cryptogram or a digital signature for the transaction.

The PIN Entry Device, PED, is a secure device that captures and processes the PIN. It should have a display to present transaction details on its screen in order to allow the user to review what he is about to authorise by entering his PIN. After the PIN is entered, the PED immediately processes it, either by encrypting it to from an Encrypted PIN Block, or by using it to enable access to a smartcard that would create a transaction cryptogram or a digital signature for the transaction. Transaction cryptogram is a sort of digital signature, but based on symmetric keys and utilising the Derived Unique Keys (DUK) method for key agreement; it is used mainly in the EMV standard to authorise the transaction. The PIN Entry Device is also able to create and verify Message Authentication Codes, MAC. The details of these operations are well defined

by national and international standardisation bodies like ANSI and ISO. Also, the national and the international institutions that rely on PIN based transactions further enforce the standards by putting their detailed specifications and minimum requirements on the security of PIN Entry Devices. Like the VISA PED Security Requirements.

A transaction terminal is a device or system that initiates a transaction. In known forms it must include or interact with a PED if it should support PIN based transactions. It provides the interface to the acquirer's host. The clearest example is the POS terminal with an external PIN pad; in this case the PIN pad is the secure PED. An ATM machine is also a transaction terminal, but it has the PED integrated in the transaction terminal itself, because there is no separate screen for the PED of the ATM machine. The whole ATM machine is considered as both a transactional terminal and a PED.

Security from malicious software, like Trojan horses, software key loggers and viruses is either by software or by avoiding the PC altogether and using an external device for entering the highly critical information, like the PIN.

However, a software solution is never guaranteed to be perfect since there is always a possibility that a malicious software product would tamper the protecting software itself and another possibility is that a new type of malicious software would arise that the current protecting software would not be able to detect or prevent it for at least a period of time until an update is made available. It is unfair to put the responsibility of the software security of the PC on the user.

Also, the hardware solution could be very secure, but what prohibits it from spreading to most PC users is its high cost. A good example of a hardware solution is the smartcard reader with a PIN pad integrated in the smartcard reader itself; the PIN is never sent to the host computer, it is sent directly to the smartcard. This integrated PIN pad adds a very high value to security because without it, i.e. using a normal smartcard reader that does not have an integrated PIN pad, a malicious software can capture the PIN from the keyboard of the PC and every time it detects that the smartcard is inserted, it would maliciously use it to make fraudulent transactions, as many times as it

wishes but when the smartcard reader has an integrated PIN pad on it, this can never happen. A different attack however can still happen even if the smartcard reader has a PIN pad: the malicious software can always tamper with the data coming out of the PC to the smartcard reader in order to be signed. This would make the actual data that is sent to the smartcard reader to be different from what the user sees on the PC screen and wants to sign. The hardware solution to this attack has been to integrate a small screen on the smartcard reader in addition to the PIN pad. The smartcard reader displays what is actually going to be signed. Of course this makes the smartcard reader more expensive; but without this secure screen on the smartcard reader, the malicious software can make a fraudulent transaction each time the user tries to make a transaction.

E-commerce transactions using credit cards was initially treated in the same way as Mail Order and Telephone Order (MO/TO) transactions where the card holder writes or tells his card number in clear to the merchant. Four security problems arose here, one after the other:

1. The Internet is a public network and people other than the card holder and the merchant can capture the packets passing between them and get the credit card number and use if for fraudulent use. The best solution that solved this network sniffing problem was the Secure Socket Layer or SSL, which is now integrated in most web browsers.

2. The second problem is that, the credit card number and the expiration date are not really confidential. Any physical merchant will have the credit card number and the expiration date of the credit card written on the receipt and if he is malicious, he could use it to make fraudulent Internet transactions. This fraud happens even with card holders who never use the Internet altogether. At solution is to incorporate the Cardholder Verification Value 2 CVV2 or CVC2 or the like, which is a three or four digits number on the back of credit cards, to be entered in internet transactions and should not be saved by merchants in the database. To further enhance security they added the Address Verification System, where the card holder must enter his billing address to be verified by the bank in order to

make sure that the legitimate card holder is the one who is actually making the transaction.

3. The third problem was in the internet merchant himself being fraudulent and stores a copy of all the card holder's confidential data, including the CVV2 and billing address in order to intentionally make fraudulent transactions. The most famous solution that solved this problem was the 3D Secure, which is basically a variant from the well known SET protocol. It requires that the issuer bank, the bank of cardholder, gets involved in the transaction to authenticate the user. The two most common methods for authenticating the user are: the user name and password method and the smartcard and PIN method. Of course the smartcard readers commonly used, do not integrate a PIN pad and search because of cost. So, the password or the smartcard PIN are entered on the keyboard on the PC.

4. The fourth problem, is the most critical of because it places unfair liability on the user. This problem is malicious and spying software that are able to read every key stroke typed on the keyboard and see everything displayed on the screen and even take full control over the victim's computer. This is not a rare problem, statistics how that computers today are infected with Trojan horses and spying software. Unfortunately, except for this current invention, this problem has no low cost solution. The secure solutions are very expensive, like the use of smartcard with a reader that has an integrated PIN pad and an integrated LCD screen or a unconnected token with display and keypad for generating transaction certificates.

Of the various problems in the field, there is a particular need for a low cost, customer owned transaction security terminal, like a secure PIN pad or a secure digital signature that allows secure transactions terminals to be built around it to provide non-repudiation while relying on well established security standards, like the ANSI and ISO PIN security standards and KC2SB and E951 for electronic signature standards..

The personal computer is made of hardware, kernel, operating system, services and applications; it is practically unfair to hold the user liable or even responsible for the security of all these layers especially because malicious software can in many cases be undetectable or at least difficult to detect for some period of time until it is discovered and a detection and/or removable tool becomes available. Also, statistics reveal that

most PCs today are infected with Trojan horses and/or spy software. Malicious hardware, like hardware key loggers can also be attached to personal computers. Accordingly these and other problems exist in prohibiting customer owned relatively low cost transaction security terminals.

Summary of the invention

In order to solve or at least mitigate one or more problems in the prior art, one aspect of the invention provides a transaction terminal adapted to enable a secure transaction over a communications network, comprising a controller having a first source for booting a normal operating environment which operably enables communication between the terminal and the network, the controller being adapted to create transaction data in accordance with user operation of the transaction terminal, to save the transaction data to a specified location, and a second source for booting the terminal into a secure environment having limited operability compared to the normal operating environment but enabling security critical operation to be done securely, like for example enabling entry by the user of authentication information, such as a PIN, for authenticating the transaction data which is retrievable from the specific location by the transaction terminal in the secure operating environment, the transaction terminal being adapted to enable shut down of the normal operating environment and booting of the secure operating environment to enable authentication and encryption of the transaction data.

Beneficially the transaction terminal can be a personal computer and the source of software for the secure operating environment can be a thin operating system application stored on a removable device such as a readable medium like a CD or USB memory stick.

Another aspect of the invention provides a process that allows a transaction terminal such as a personal computer to be used as a secure PIN entry device. The process includes one or more of the steps of: A preparing transaction data, B storing the transaction data, C shutting down, D rebooting from a secure media or device, E loading the transaction data, F allow modification of the transaction data, G entry of secret information such as a PIN, H storage of the secured transaction data in memory, shutting down and rebooting to a normal operating environment and subsequent

processing of the transaction in the normal operating environment. These steps are explained in more detail below.

A Preparing transaction data, as with the normal transaction terminal, like POS terminals, the terminal prepares the transaction data, like the amount, the card number and currency, before interacting with the PED, and a separate aspect of the invention is the secure PIN entry and its interface with the transaction terminal,

B Storing the transaction data in a specified location such as non-volatile memory, which is the suggested interface between the transaction terminal and the secure PIN entry using personal computer, since the non-volatile memory could be for example, the hard disk of the computer or a USB token or any memory that is persistent between system restarts,

C Rebooting, in other words restarting, or hibernating and starting, or shutting down and restarting the computer, beneficially to ensure that any malicious software is no longer in control since restarting the computer is usually very characteristic and very easily distinguishable by the normal user and hence it is too difficult for malicious software to imitate the behaviour of restarting the computer while still being in control on the computer, but the user might need to adjust the boot sequence of his personal computer, and even this adjustment, if ever needed, is not a very advanced task and only need be done once, to configure the computer to seek booting from the bootable media or device mentioned in the next step before attempting to boot normally from the hard disk for example, but fortunately, most computers are already adjusted to seek booting from CD/DVD and USB devices before normally booting from the hard disk, hence need no any adjustment,

D Booting from a bootable media or device; beneficially the bootable media or device being sufficiently difficult to counterfeit and sufficiently difficult to tamper with the data stored in it and optionally difficult to copy. The reason why it is necessary for this bootable media or device to be difficult to counterfeit, is to help the user to distinguish between the original and nay fake bootable media or device that would steal his PIN. The bootable media or device would be mailed to the user. this bootable media or device should also be sufficiently difficult to tamper with the data on it, in order to prevent malicious software from being injected in it. Two examples are:

1. Bootable media: A CD-ROM or a DVD-ROM or a closed CD-R, are by default immune to any tampering on the data on it. To make it anti-counterfeit, special artwork, holograms, marks, even shaping of the CD itself can be applied on the physical disc or make it difficult to imitate.

2. Bootable device: personal computers today allow for booting from USB devices. People today are able to boot from USB drives and USB sticks but there is no technical problem at all to boot from a USB smartcard reader, for example. This may sound very strange, but most, if not all, USB smartcard readers today have firmware which can very easily be enlarged in size to even a few megabytes; this is quite more than sufficient to boot into the secure environment, described in the next step, with a very nice GUI. The firmware of most USB smartcard readers, especially the EMV certified ones, is sufficiently difficult to tamper with the data in it; even firmware upgrades is done securely using cryptographic methods, like digital signatures. Smartcard readers, especially the EMV certified ones, are sufficiently difficult to counterfeit, they even carry certification logos and holographic.

Preferably, the booted environment (secure operating environment) should be minimalist, ideally with no networking capabilities in order to dramatically reduce the effort needed to verify its security. It should load only the PED software among with any helper programs or trusted software. Software is assumed to be trusted if it is loaded from the bootable media or device mentioned in the above step or digitally signed using a valid digital software from an untrusted source, is to have its checksum stored on a trusted media or device, but this method does not allow newer versions to be loaded without modification on the trusted media or device that contains the checksum. Another requirement on the booted environment is its ability to boot from a read-only source, like a CD, because most boot media or devices that satisfy the requirements in the previous step are not writeable, at least during normal operation.

There are many was to allow booting from a read-only media, here are two examples:

1. Writing it from scratch: this allows far more optimisation on the size and takes into consideration that the source media or device from which it is booting is not writeable.

2. Adapting an already available OS: usually normal operating systems require booting from a writeable media, one way to solve this problem is by using the RAM disk technique, in which the boot program would initially reserve part of the personal computer's RAM and use it as a disk emulation, which is of course writeable.

E Loading transaction data from the non-volatile memory. This is where the secure PIN entry system gets the request from the transaction terminal.

F Presenting the transaction data to the user and optionally allowing the user to modify and/or complete it. If the transaction data was digitally signed by its creator, then it should be verified and the user should be notified of the validity of the digital signature. Because this secure booted environment will prevent the user from being faked by a malicious software by telling him that the signature on the transaction data is valid while it is not or the inverse.

G If the user agrees to the information that was presented to him, he would enter his PIN. The PIN would either be used to create an encrypted PIN block or enable the use of a smartcard to create a transaction cryptogram or a digital signature. Password encrypted private or secret keys can be used as a low cost alternative to smartcards for creating digital signatures. Other confidential data, like for example the ATM card track2 data which is the data on the magnetic stripe of a debit or credit card, can also be password protected using password based encryption. This way, any one who could maliciously copy information from the users computer or even from a bootable media or device, can not access the confidential information stored on it. Another level of securing the very confidential data like the TM card track2 data is to have it stored encrypted under the public key of the server. This way, no one can use the information on the personal computer to try to make an attack on another channel, like real ATM machines for example, simply because he does not have the ATM card information in clear.

Storing the secured transaction request in non-volatile memory. The is where the secure PIN entry system brings its reply to the transaction terminal. It is worthy to note that is need not be the same non-volatile memory used above for obtaining the transaction data.

Restarting the computer back to normal operation. The bootable media or device should have been ejected, or the boot loader on the bootable media or device is able to detect that there is no currently pending transaction so resumes to normal operation or the boot loader on the bootable media or device allows the user to choose manually whether he wants to boot to the secure environment or to normal operation.

H Loading the secured transaction request from the later non-volatile memory. This step is done by the transaction terminal system which would normally:

1. Send the transaction for authorisation,

2. Receiving the response,

3. Present the response to the user and optionally store it.

Existing precautions for protecting your PIN when using Secure PIN Entry using Personal Computer are to a great extent similar to the precautions for protecting your PIN on an ATM or a POS.

1. You should hide your hand that enters the PIN, so that no one can see your PIN while you type it, this also protects your PIN from being stolen using a remote camera.

2. You should check that there is no hardware key logger connected through keyboard cable

3. You should be able to trust that the personal computer does not contain malicious hardware or firmware for capturing the clear text PIN. This can be an easy prerequisite given that you are using your own notebook or PC at home. You should never for example use a computer in an Internet cafe to make a PIN based transaction.

As the process allows for secure PIN entry, it allows also for entering additional critical pieces of information like a user choosable one time password or secret number to be encrypted and sent to the server in order to be used some time later on another less-secure channel like the Internet or WAP or telephone or IVR. The user can for example enter a user choosable CVV2 or CVC2 or 3D secure password that the he will use in

the next transaction that he will make on the Internet. The user may also specify the same constraints on the next transaction that will happen using the specified CVV2/CVC2, like the maximum amount of the transaction and/or the time within which the transaction should be made.

A mobile phone that supports Java is a sort of personal computer that always boots into a secure environment from its firmware by default. Hence it does not need the parts of the process that involve restarting and booting into a secure environment; all that is needed is to launch the secure PIN entry application. The PIN entry application can execute directly and securely on the mobile phone. Of course the physical phone must be trusted by the user; the mobile phone owned by the user himself could satisfy this requirement very easily, because the user knows that his own phone was never physically accessible to anyone who has the technical expertise and malicious motivation in addition to the ability to pay the cost of obtaining and inserting a malicious hardware or firmware in his mobile phone. The PIN entry application that will run on the mobile phone can also interact with the SIM in the mobile, which is basically a smartcard; this allows for the second use of the secure PIN entry, which is to enable the access to a component in the smartcard that would create a transaction cryptogram or a digital signature for the transaction.

Beneficially, the terminal and process can be used to protect the PIN entry of any PIN that needs to be secured, not just the ATM PIN, but using the same security standards that were originally made to secure the ATM PIN.

The process radically protects the user from any malicious software that might affect the security of PIN entry; it dramatically reduces the user responsibilities to physical security considerations only, like those in ATM transactions.

If the transaction data was digitally signed by some source, Digital signatures on transaction data will be verified securely in this secure environment, because there is no malicious software that can report falsely about the validity of the digital signature.

Allowing the user to choose the CVV2 or the CVC2 or the 3D Secure password to be used in the next transaction to be done on a less secure channel, has a huge benefit because it allows for extremely secure e-commerce without requiring any change in the

traditional merchant system, acquirer bank system and the card transactions network; the only system that will need little adaptation is the user bank system in order to allow the CVV2 or the CVC2 or the 3D Secure password to the user choosable and changeable, even for every transaction. An example of the traditional merchant system is a payment web-page with SSL that allows the user to enter his credit card number, expiration date and CVV2 or CVC2 and sends this information to the acquirer bank, which is usually the bank of the merchant. Neither the merchant system nor the acquirer bank needs to change anything. Not even the card transactions network, like MasterCard and VISA will need to modify anything. The same thing can be done for the 3D Secure password, the process allows the 3D secure password to be different, user choosable, for every transaction; because before every transaction, the user would follow the said process and enter in the secure state, away from any malicious software, the 3D secure password. After that he will use it in the next transaction to be done on unsecured channels like the Internet where malicious software can capture it, or on IVR where it could also be captured or on any less secure channel, but it will be of no use for the attacker any more.

Other aspects and features of the invention are described in the claims at the end and in relation to the following embodiments.

A device for enabling a computer to be booted into a secure operating environment, comprising a body for carrying a storage medium for storing software which software is readable by computer in use and enables the computer to run in a secure operating environment, wherein the body comprising anti counterfeit feature and/or a tamper-evident feature, and the storage medium comprises a special area which cannot be overwritten by an unauthorised user, wherein the software is stored at least in part in the special area.

Brief Description of Drawings

Embodiments of the terminal and process according to the invention will now be described by way of example only, with reference to the following drawings:

Figure 1 is a block diagram of a personal computer for carrying out the invention,

Figure 2 shows the security dependency in the case when the PIN is encrypted by the PIN entry system itself as described by the first embodiment below,

Figure 3 is a flow diagram of some of the process steps according to the invention,

Figure 4 comprises three views of keypads presented to the user to help prevent detection of entry secret information through a keypad having malicious key-logger devices,

Figure 5 is a schematic representation of events versus time according to the process, and

Figure 6 is a flow diagram of steps in the process according to the invention.

Figure 7A and B shows a USB memory according to the invention.

Figure 8 is a USB token according to the invention.

Figure 9A and B shows a smartcard reader according to the invention.

Figure 10A and B is a CD adapted according to the invention.

Figure 11 is a block diagram of independent inventive concepts according to the invention.

Detailed description of preferred embodiments

A first embodiment is now described in which a transaction terminal application puts the transaction data in a file on the hard disk. With reference to Figure 1 there is shown a transaction or client terminal 10 comprising a personal computer 12, a monitor 14, a keyboard 16 having a letters and number keypad 18. The PC 12 comprises usual components such as a mother board 30 and a hard drive 32 as well as a CD RW drive 20, a floppy 31/2" drive 22, and input ports 24 such as USB sockets for example enable connection of a smart card reader. The PC comprises other components enabling operation as a network terminal, including a network connection device such as a modem 26 for connection to a network 28 such as the internet.

The user inserts the secure PIN entry CD in his computer's CD drive. The CD is a business card shaped CD-R with a hologram sticker on it. The CD-R is closed, i.e. no other sessions can be added to the CD--R in order to change the data on it. The user hibernates the computer. The user starts the computer. The computer boots from the CD. The secure PIN entry application reads the transaction data from the hard disk. The secure PIN entry application displays the transaction data to the user. The user agrees on the transaction and enters his PIN. The secure PIN entry application generates a random DES or 3DES key, PIN Key, and uses it to encrypt the PIN. The secure PIN entry application generates another DES or 3DES key, Auth Key, and uses it for generating a Message Authentication Code, MAC, on the transaction. The secure PIN entry application encrypts the Auth Key under the PIN Key.

The secure PIN entry application encrypts the PIN Key under the public key of the HSM used in the server side. The public key is stored on the CD-R itself. The user may also enter, once or twice, the CVV2 to be sued in the next Internet transaction. The secure PIN entry application would also encrypt this user choosable CVV2. The secure PIN entry application forms its reply which is the transaction data, the MAC and the encrypted PIN. The secure PIN entry application stores the reply in a file on the computer's hard disk. The user ejects the CD and restart the computer. The computer returns back from hibernation. The transaction terminal application can now load the reply file from the hard disk. This reply file is almost the transaction request which the transaction terminal needs to send to the acquirer to be authorised. If there was a user choosable encrypted CVV2, the server side would use it to update the cards database in order to secure the next e-commerce transactions coming from less secure channels like the Internet. The user may also specify the same constraints on the next transaction that will happen using the specified CVV2/CVC2 or 3D secure password, like the maximum amount of the transaction and/or the time within which the transaction should be made.

These steps are shown in Figure 3 as steps 100 to 118. After step 118 the computer completes the transaction as described above.

In another embodiment the personal computer has a bootable smartcard reader, as described in the description above, but the transaction terminal forms the transaction

data and puts it in a file on the smartcard which is inserted in the smartcard reader. The user hibernates the computer. The user starts the computer. The computer boots from the smartcard reader, the boot loader checks the smartcard and finds that there is a pending transaction data, so it launches the secure PIN entry application. The secure PIN entry application displays the transaction data to the user. The user agrees on the transaction and enters his PIN. The PIN is sent to the smartcard as a verify PIN command. The transaction data is sent to the smartcard with a command to create an Authorisation Request Cryptogram, ARQC. The secure PIN entry application forms its reply which is the transaction data and the ARQC. The secure PIN entry application stores the reply in a file on the smartcard. The user restarts the computer. The boot loader on the smartcard reader will sense that there is no new application data so it skips its booting and the computer boots normally. The computer returns back from hibernation. The transaction terminal application can now load the reply file from the smartcard. This reply file is almost the transaction request which the transaction terminal needs to send to the acquirer to be authorised.

The physical bootable media or devices can be manufactured and loaded with software that allows the user to apply the process in the transactions he makes.

The process according to the invention can secure the entry of the PIN of an ATM transaction made from the user's own computer. This allows the user to access and make transactions from his debit and credit account while he is in his home using his personal computer or mobile phone. The issuer bank of the cardholder will feel exactly that the transaction was coming from a real ATM, because the process not only secures the PIN entry, but also allows for transactions to be made in the same way specified in the ISO 8583 standard for card originated transactions. So, the issuer bank will require no change in his system allowing his customers to use this type of transaction. All the changes could be handled by the d riving system, which is usually owned by the Acquirer bank who is the first to receive the transaction from the transaction terminal, in our case the personal computer.

The invention can be used to allow secure PIN entry for using a smartcard to sign a contract or transaction.

In one embodiment, the invention provides an Electronic Check with Digital Signature

using a USB Token

A USB token can be viewed as a smartcard reader and a smartcard combined in a single device, which is used in this embodiment but the user may also be given other options, such as use of a special bootable media to be used in case his computer does not support booting from USB; but he will be using the USB token in the process of digitally signing the check.

1- The user decides to make an electronic check for a supplier who sent him an email requesting payment

2- The user logs onto his bank's website, which utilizes two-way SSL. The browser asks the user for his certificate, the user inserts his USB token and selects his certificate and enters his sign-on PIN.

3- After the successful sign-in, the user selects create check

4- He uses the details in the email that was sent to him to copy and paste the supplier's name and the requested amount to the web form of the electronic check.

5- Now the check is almost done, except for the signature. The user presses, the "sign" button

6- The website, through a special browser plug-in, stores the unsigned check-data on the USB token and sends a command to the operating system to hibernate the computer

7- The computer hibernates and turns off

8- The user turns on the computer manually and deliberately presses, for example the "F1" button, to enter the firmware setup of his computer in order to configure the boot priority of his computer if it was not already configured to Boot from USB first. If it was already configured, then the user just needs to note the message usually displayed by firmware that tells which boot device it is actually going to boot from.

9- The firmware boots the computer from the USB token

10- The booted environment checks if there is a pending transaction in the USB token; it finds that there is a check that needs to be signed.

11- The check details gets displayed on the user's screen

12- The user reviews it

13- The user enters his secure PIN, which is used solely in secure environments (the user should never enter it on the normal operating system).

14- The entered PIN enables the digital signing application on the USB Token and signs the check and stores it back on the USB Token again.

15- The secure environment notifies the user that it will return back to the normal environment, so that the user would be able to resume the transaction

16- The user presses ok

17- The secure environment chain-boots directly to the second boot device, which is the hard disk (it does not need to restart the computer)

18- The computer returns back from the hibernate state with every thing exactly as left. But now the USB token contains the signed check

19- If the user is using a dial-up modem, he will have to reconnect to the Internet in order to resume the transaction

20- The user selects send check on his bank's website

21- He is asked about the email address of the receiver, the user copies it from the email and pastes it on the web-form and presses ok

22- The bank's website, through the plug-in, copies the signed check from the USB Token and sends it to the requested receiver

In yet another embodiment, the invention provides an internet banking process using digitally signed payment transactions using a laser bank card. The laser bank card is the bootable media described in the invention that will be used to enable the secure process to be done.

Initially, the bank sends a shaped CD to his customer by mail. The CD is already personalized for the user, who is preferably instructed to call the bank to activate it, like most credit card or secondary credit card.

Physically, the laser bank card is a shaped CD with special printing and optionally holograms, similar to those found on credit cards. The CD is for example a 50MB CD-R, personalized with the user ID of the customer to whom the bank is sending it to. It contains a windows application, that auto-runs when the CD is inserted in a computer running windows. The application also looks like a credit-card on the screen; it does

not require any setup; it just auto-runs when the CD is inserted. The application first asks customer for his Internet Banking password; entering the password on a windows application is much more secure than entering it on a website, because faking an application is much more difficult than DNS spoofing or phishing. But still such passwords are not sufficiently secure at all for the non-repudiation requirements of electronic signatures, simply became a software key-logger can very easily capture this password.

After the user enters his password on the credit-card looking application, he sees his credit card number, expiration date, his name and his credit card balance in addition to the bank's logo. When he clicks on the bank's logo, the application launches the default web browser passing to it a URL that enables the customer to enter automatically to the Internet Banking page of his bank. He does not need to enter any more usernames or passwords. This is because the credit-card looking application already sent the username and a special one time password in the URL.

Until this phase, the customer is happy, because he replaced the process of entering a username and entering a password, by inserting a CD and entering a password, which looks more secure and appealing to the user.

The user selects transfers on the Internet banking website, and selects the payees to whom he wants to make transfers, e.g. the telephone company, the cable TV company, the card loan...etc. He fills in the amounts and the accounts numbers in the table presented to him on the Internet banking web page. When he presses submit on the web page, the credit-card shaped windows application that was used in the login to the Internet banking website, will popup again and tell the user that the computer should go into the secure environment in order to be able to enter the ATM PIN to approve the transaction.

The user clicks ok; the application will store the transaction data in a special location on the hard disk, for example the last 30 sectors on track 0 of any hard disk is intentionally left free by operating systems to allow for special boot loaders to work. Track 0 is usually 64 sectors; each can sector holds 512 bytes. So, using sectors 40 to 50 on Track 0 should be very sufficient and very safe.

After successfully storing the transaction data, the credit-card application will send a special command to the operating system telling it to hibernate and then restart. This is a custom modified version of hibernate that would reset the computer after the hibernate operation, instead of turning the computer off.

Most computers are already configured to seek boot from CD before any other boot source. Assuming that this was the case; the user will simply see his computer is saying, "booting now from CD".

The user did not need to intervene to make these things happen.

Now, when the CD starts booting, the booted environment will initially check if there is a pending transaction or transactions left for it on Track 0. If yes it will continue launching the secure environment, otherwise it will immediately chain boot to normal environment; chain-booting does not require restarting, hence does not require ejecting the CD.

In our example, there is actually a pending transaction left on Track 0. The secure environment will load it and display it on a virtual graphical PIN Entry Device on the screen. The user reviews the transaction details and amounts. He then uses the keyboard to enter his ATM PIN, which he already knows by heart.

After pressing enter, the encryption and security algorithms are executed according to the banking standards, which can make use of the public key of the bank that is burnt on the CD to send information that can only be read by the bank's Hardware Security Module; finally the result of these tasks that require secure environment is stored back on Track 0. The environment notifies the user that it will return back to the normal environment to enable the transaction to resume.

The secure environment chain boots to the original environment, without requiring the CD to be ejected; because chain booting means booting an operating system after another one was already booted without the need for restarting the computer. This makes the process very smooth, because the user would not have to eject the CD at all.

The normal operating system returns back very quickly, because it is returning back from hibernation, which is usually extremely fast.

The credit-card application will sense that it returned from hibernation, because the windows operating system notifies all the applications that register for this event (returning from hibernation). Now the credit-card application will silently check if any reply was sent back to it in Track 0; if no reply is found, the application will simply ignore the event; in our example the application will actually find the reply that was left for it by the secure environment. So, it automatically loads it and asks the user to enter again his Internet banking password, in order to connect to the bank's server and resume the transaction.

In the case when the user is using a dialup connection to connect to the Internet, then the application will automatically reconnect to the Internet or at least launch the Internet connection for the user to confirm reconnecting to the Internet, by pressing dial.

After successful entry of the password the application will send the transaction to the bank's server, which in turns verifies the secure transaction and replies back telling the user about its success or failure.

Accordingly the invention provides a process that allows a user to make secure PIN-based transactions using his personal computer through one or more of the steps of: (a) Preparing transaction data (b) Storing it in non-volatile memory, (c) Restarting or hibernating the computer, (d) booting into a secure, un-networked, environment from a bootable media or device; this bootable media or device must be sufficiently difficult to counterfeit and sufficiently difficult to tamper with the data stored in it and optionally difficult to copy, (e) Securely launching the secure PIN entry software, (f) Loading transaction data from the non-volatile memory, (g) Presenting the transaction data to the user and optionally allowing the user to modify and/or complete it, (h) Secure PIN entry resulting in an encrypted PIN block and/or enabling the user and/or the generation of the appropriate keys for creating message authentication codes(s) and/or cryptogram(s) and/or digital signature(s) according to the transaction security standards; The user can also enter a password to enable secure access to password encrypted secret keys, private keys and confidential data; The user can also enter secure information to update his records in the server side system, like a user choosable CVV2/CVC2 or new 3D Secure password, (i) Storing the secured transaction request

in non-volatile memory, (j) Restarting the computer back to normal operation, (k) Loading the secured transaction request from non-volatile memory, (i) Sending the transaction for authorisation, (m) Receiving the response, (n) Presenting the response to the user and optionally storing it. The process radically protects the user from any malicious software that might affect the security of PIN entry; it dramatically reduces the user responsibilities to physical security considerations only, like those in ATM transactions. The user should use a personal computer that he knows that it does not contain malicious hardware; this could easily be his own notebook or PC at home. He should still quickly check that there is no external hardware key logger attached to the computer. The process enables non-repudiation in e-commerce transactions. The process is also an ideal solution for securing the use of smartcards on personal computers.

With reference to the above mode of operation identified by the letters a to n, some alternatives are set out below.

Hibernating restart: regarding to step c), there is another flavor of restarting the computer proved to make the process easier to the user. It is basically a modified version of hibernate where the computer would restart immediately automatically after hibernation instead of the normal action of hibernate where the computer would turn off and the user would have to power on the computer manually by himself. Many users prefer hibernate rather than restarting the computer because hibernate usually takes much less time than normal restart and more importantly it keeps all the user's applications opened in the same state that the user left it. This new flavor is not normally bundled with operating systems, like Microsoft Windows; it is a custom implementation designed specifically to support this invention.

Chain booting: regarding to step j) where the computer is restarted back to normal operation. We were able to make an optimized implementation where the step of returning back to the normal operating system does not require a restart of the computer; instead the secure environment would itself boot the computer back to normal operation after the user finishes working in the secure environment, without restarting the computer; this process is called chain booting because a second operating environment is booted after an initial one boots and finishes its work, without restarting

the computer. This is realized technically, by the secure environment after finishing all its work with the user, performs the operation that a BIOS would normally do to boot the computer, i.e. loading the boot sector of the hard disk into memory and giving control to that code by jumping into it. Our code is able to detect the correct device to which it should chain boot; this is done by asking the BIOS about the boot priority and taking the second on the list, because the first is the boot device that booted the computer into the secure mode. This dramatic enhancement also bypassed the need for the user to eject the boot media or change the boot sequence of the computer manually in order to get back to the normal operation and made the process much easier to the user. The chain booting idea can not be used in step c) because in that case we will not be able to guarantee that the activity of any malicious software is killed; also it is technically much difficult to do chain booting from a full-fledged environment like for example Windows XP, but it is possible to chain boot from light environments like our optimized secure mode environment.

Ensuring the secure boot: Between steps c) and d) the user should take care that his PC's firmware is actually going to boot from the secure bootable media or device. This check is essential, because if a malicious person configured the PC to attempt booting from network for example as a priority before the intended secure bootable media or device, then he can boot a fake environment to the user that may look like the secure environment to trick the user and capture his proofs of identity and secrets. Many PC's firmware display in a very characteristic way which device it is actually going to boot from, this should be sufficient for the user to note. But if the PC's firmware does not display this information, the user would have to enter the setup of his computer and check the boot priority is correctly configured, in order to avoid the possibility of such fraudulent attacks. Though this is an extra step on the user, but it ensures the security of the process. To clarify how important this step is, malicious software that are active in the normal mode can change the boot priority of the computer by writing to specific memory locations in the CMOS; though this is technically difficult and requires difficult calculation of difficult check sums and sometimes digital signatures and is very different from one computer to another but it is theoretically possible. As a conclusion, in some old computers, the boot priority can be tampered with directly from Windows, even if the BIOS is protected by a password. So, the user should check that

his computer is actually going to boot from the intended source in each time he boots into the secure mode.

Adding confidentiality to the communication between the normal mode and the secure mode: Step b) and Step f) involve data exchanged between normal mode and secure mode. As an optional additional security, this data can be encrypted to ensure that if the process is interrupted in any way due to a computer hardware failure or any other reason, the non-critical transaction data would still be kept undisclosed. Basic encryption mechanisms can be used, because the data is already not critical, the critical parts of the transaction data are already encrypted using the appropriate encryption mechanism according to the transaction protocol. This additional layer of security is something like communication line encryption.

Beneficially a keyboard layout change to circumvent cheap hardware attacks on keyboard entered secrets can also be provided. This is a way to even circumvent the hardware keyboard logger attacks without having to physically detect them and remove them! I.e. using this extra step, even if the there is a keyboard logger attached to the computer, it would contain information that would not benefit the attacker in any way. The idea is that during the entry of a secure PIN for example, the secure environment would display the numeric key pad on the screen and label each key with a random letter that can be typed on the keyboard instead of the actual numeric key. Because key loggers can not detect what is displayed on the screen, they would not know what the actual PIN was. Another straight forward way is to use a mouse on a graphical key pad or a graphical on screen keyboard, without any keyboard layout change. Referring to Figure 4, 4A shows a typical keypad layout on keyboard such as keyboard 16 shown in Figure one, and this layout can be presented to the user via an output such as a display monitor 14. In order to vary the actual key used by the user to input a PIN number, the user is informed of other keys which represent the number for the purposes of the secure entry. Examples of random number allocation to keys are shown in Figures 4B and 4C. In figure 4B for example, to enter the PIN 3378, the user types LLEK on the keyboard.

In another form, the whole keyboard including letter key can be scrambled e.g. randomly allow entry of other type of secret information other than PINs.

The possible bootable media or devices that can be used are based on the necessary and sufficient criteria for a bootable media or device that can be used in this process are: the physical characteristics of this bootable media or device must be sufficiently difficult to counterfeit and the data stored on it must be sufficiently difficult to tamper with. Two examples are, one based on the CD-R and another based on a smartcard reader, but there are many other examples that can be made to satisfy the process requirements, and here are some of the examples:

a) High end smartcards today have memory of up to 4 mega bytes. They also have circuits that talk the USB protocol directly without the need for a reader, just a connector (http://www.egateopen.axalto.com/) while the current implementation that we have implemented in our company for the secure bootable environment including all the required software is only 400KB and can be further optimized depending on the specific transaction protocol needs. So, the smartcard itself, without any reader, can be used as a bootable device.

b) USB tokens, they are basically a tamper resistant device like smartcards, with a USB interface.

c) Specially modified Secure Digital Card (SD Card) that has write-protected areas to protect the secure boot environment from tampering.

d) DVD and its derivatives, except the re-writable ones.

e) The bootable device could even be the BIOS of the computer itself. It may contain a light secure environment

f) Many other examples can still be stated, the important thing is that special features must be present on the bootable media or device that makes it distinguishable from a fake one that does not have the legitimate secure boot environment. And it must be sufficiently difficult to tamper or change the data on the bootable media or

device that contain the secure environment, at least the area that is responsible to load the secure boot environment.

The proofs of identity of the user need not to be just a PIN, since the proofs of identity of the user are categorized into three main categories:

a) Things that the user knows: like passwords, PINs or any secret that the user knows; not necessary keyboard entered secret. It can be a shape or a choice that he knows or looks-up from an external secrets sheet. It can be a word or words dictated through a microphone or a written on a write-pad using a digital pen. The essence here is on transferring the secret to the computer in any way; the biometrics part of it is covered below.

b) Things that the user has: like any token, whether it was a hardware token, a software token or a media-based authentication method

c) Things that the user is: this is biometrics, like voice print, finger print...etc.

Sometimes these proofs of identity can be coupled together, for example, a smartcard requires a PIN to be activated, which is an example of a something that you have and something that you know that are coupled together.

All these proofs of identity face risks if the environment is not secure:

a) Things that the user knows: would be immediately tapped by the malicious software

b) Things that the user has: Though the malicious software may not be able to make a fraudulent copy of the secure token of the user, but as stated the malicious software would maliciously make the token, or smartcard, sign fraudulent transactions instead of, or in addition to, the ones the user actually wants to sign

c) Things that the user does: the malicious software can obviously tap what the user presents to identify him and simply replays it. The

replay attack is a famous attic in biometrics. The secure environment prevents this attack from ever happening, without the cost of sophisticated protection hardware. For example the normal sound card of the computer can be used to take the voice print, without the fear of malicious software that might tap it. Same thing in finger print and paper signature. The fingerprint scanner does not need to be sophisticated and integrated in a smartcard reader making it more expensive; it can just be simply connected directly to the PC without fear of tapping by malicious software that would then attempt to make replay attacks.

Types of transaction covered are diverse and any type of electronic transaction can be secured using the current invention. We have specifically mentioned two types that are extremely important: a) PIN based financial transactions (with and without a smartcard) b) Digital signature transactions (where contracts or forms need to be digitally signed) they may already have initial signatures on them that should be verified, before the user signs them. Or the user may just be satisfied by verifying the signature that may already be on the contract or form. Focus on type a) (PIN based financial transactions) has been given here as mentioned in the application were covering a PIN based transaction. In the first best mode the PIN was an online PIN, i.e. the PIN gets encrypted and attached to the transaction data to be verified by the host after returning to the normal mode. While the second Best Mode is a PIN of a smartcard or token where the PIN is used to enable access to the smartcard or token which in turns generate cryptograms that make the transaction data authorized by the user.

- The second type of transactions proved to be of even more importance, our secure process enables digital signatures to be made securely using the personal computer, without having to buy a PIN pad with an LCD screen which would be very expensive to be owned by the user

- These two types of transactions are the two types that we attempted to protect in the first application, but we would like to at least enforce this clearly in the new application.

4- Non-volatile memory used in transferring the transaction data from the transaction terminal (in the normal mode) to the secure environment can be, any or a combination of the following:

a. Hard disk location:

i. A special free sector(s) on the hard disk. For example:

1. It is known that the first track of any hard disk is unused, except for the first three sectors used by the OS boot loaders, so the remaining sectors of the track can be used; typically there are 64 sectors in a track, each of 512bytes of size

2. Also at the end of each hard disk there is also a free area

ii. A special location on the hard disk. For example:

1. The cookies folder of the web browser of the normal environment

2. Temp areas of the operating system

iii. If the secure environment is able to browse the file system of the hard disks of the user, then the user may be allowed to choose the file that contains the transaction data, manually, by him

iv. In RAM and relying on hibernating to bring all memory dumped to the disk, and then the secure mode will search for the data in the hibernate dump file

b. Non-volatile memory area on a device, like for example

i. The smartcard reader itself can have a special temp memory area to be used for this purpose

ii. An area on a smartcard or token

iii. The NVRAM of the motherboard. New motherboards has a big NVRAM to be used by operating systems to pass data between restarts

iv. Devices like display cards, modems or sound cards may have free locations

c. Media or user storage devices:

i. Floppy disks

ii. USB sticks

iii. Memory cards, like MMC, SD Memory and the like

d. Manually, by writing it down or remembering it and typing it back again when in the secure environment

e. Through a server side: In case where the secure environment is able to connect to some server, i.e. has some sort of networking capability or can use any simple communication channel like sending and receiving touch tones using a modem. Then the transaction data can be transferred through that server side which is accessible from both normal environment and secure environment, even if the channels used to access the server from the normal environment and the secure environment is different.

5- The way for transferring the updated transaction data from the secure mode back to the transaction terminal system

a. Through *any* of the ways described in the above section

b. Since, chain booting can be used to boot back to the normal environment without resetting or turning off or restarting the computer, the normal RAM can be used as a channel for passing the transaction data back to the normal environment, there are many ways to do this, like for example:

i. Installing a virtual device driver, that will become accessible in the secure mode

ii. Using memory locations normally not over writable by the operating system (like BIOS interrupts area)

c. If the changed info in the transaction is a small thing, like a small transaction certificate of a small number of digits, or if it is a simple one time password, then the user can study it or even write it down and type it again after returning to the normal environment. Here we used a manual channel to pass back the information to the transaction terminal in the normal environment.

d. In case where the transaction terminal is also accessible through another channel, like Interactive Voice Response (IVR), then the user can even resume the transaction without even restarting back to normal operation. He simply dials the telephone number of a server, and keys, through the touch tone keys of the telephone, the transactions details that were changed, which can be as simple as a small transaction certificate or a one time password in addition to a transaction ID for example; things that he can see on the screen before leaving the secure mode. In this case the transaction completes completely without the need to return back to normal operation to complete the transaction.

e. Another similar way to the above case is when the secure environment can connect to the Internet or to some server using some communication channel, while inside the secure mode. In this case the transaction data can be sent back to the server directly while inside the secure mode or sent to an interim server and when restarted back to normal operation the data is fetched from the interim server and the transaction be resumed by the transaction terminal

The solution not only prevent malicious software from tapping and/or fraudulently using the proofs of identity of the user, the solution also prevents malicious software

from fraudulently affecting the correctness of the digital verification of the transaction data, if it were already digitally signed by one or more other parties.

We also suggest that the coupling of the original patent with CVV and 3D Secure would be removed as long as these types of transactions would still be protected. I can provide you with different Best Modes for different transaction types, if this is essential to help making people understand the invention and clearly emphasize that we want to protect these types of transactions.

The last dilemma is in steps e) f) g) h) i) namely the steps that happen after entering the secure mode. These steps have lots of variants and can also be rearranged in a number of ways. Does this mean that each variant and each rearrangement would not be protected unless clearly stated?

It is easy to specify the logical requirements and the rules that govern the variants and the possible rearrangements. Is this sufficient to protect us?

Putting the first four steps e), f), g), h) in a single step to simplify stating these requirements, the process would roughly look like this:

1- Preparing transaction data

2- Storing transaction data in non-volatile memory; it may optionally be stored in encrypted form to ensure that only the legitimate user is able to view it.

3- Restarting the computer, this could be simple restart or a shutdown and then power-on of the computer or a modified version of hibernate that restarts the computer immediately after hibernate or a normal hibernate and then power-on of the computer.

4- The user should verify that his PC firmware is configured to boot from the intended source (bootable media or device)

5- Booting into a secure, preferably un-networked, environment from a bootable media or device; the physical characteristics of this bootable media or device should be sufficiently difficult to counterfeit and sufficiently difficult to tamper with the code and/or data stored in it. (at least the part which is responsible for booting the secure environment).

6- Making the critical parts of the transaction, this involves: loading the transaction data from the non-volatile memory, decrypting it if it were already encrypted and optionally verifying it whether manually or digitally or both; the user may also be allowed to modify and/or complete the transaction data; the user should also present one or more of his proofs of identity which could be things that he knows, things that he has or things that he is, i.e. biometrics. The presenting of the proofs of identity can be done immediately after booting into the secure environment or at a later stage after the user decides that he really wants to authorize a transaction or both in which case the proofs of identity presented in each stage may of different types. Finally, cryptographic operations are made to process the transaction data to make it eligible for authorization.

7- The processed transaction data is stored on a non-volatile memory; it can optionally be encrypted to ensure that only the legitimate user would be able to view or complete the processed transaction data by sending it for authorization.

Also, in the case of digitally signing the document, we would emphasize that the user may be allowed to browse his hard disk or different storage devices to select the document that he wants to review in the secure mode and to securely verify any digital signatures it may already have and finally sign it securely. Note the use of the words computer firmware and BIOS interchangeably.

In one form the invention also provides a personal computer physically trusted by the user who wants to enter his PIN to authorise a transaction. The said PIN is either a PIN to be encrypted or a PIN to be verified by a smartcard to enable operations on it. The user prepares the transaction using a transaction terminal system or software that is outside the scope of the invention. The said transaction terminal stores the transaction data in a non-volatile memory. The user would restart, or hibernate and start or shutdown and start the said personal computer. The computer would boot from a bootable media or device that is sufficiently difficult to counterfeit and sufficiently difficult to tamper with the data stored in it. The said bootable media or device launches the secure PIN entry application. The said secure PIN entry application reads the transaction data from the said non-volatile memory. The said secure PIN entry application displays the said transaction data to the said user. The said user verifies the said displayed transaction data and decides whether to enter his PIN or not. If the said

user decided not to enter his PIN, he can just cancel the transaction and restart the computer back to normal operation and the process stops here. Else, the said user would enter his PIN. Depending on whether a smartcard is to be used or not, the said PIN entry application would process the PIN either by encrypting the PIN and optionally generating a MAC or by sending the PIN to a smartcard to enable the operation that authorises the transaction on it, which is either an EMV cryptogram or a digital signature or an encrypted PIN and optionally a MAC. The said PIN entry application would form the reply and store it in a non-volatile memory possibly different from the one said above in the transaction data. The said user may also enter a password to enable secure access to password encrypted secret keys, private keys and confidential data. This step can be done before or after the said PIN entry step. The said user may also enter secure information that updates his records in the server side system, like a user choosable CVV2/CVC2 or new 3D Secure password. The said PIN entry application also encrypts the said other secrets. The said user restarts the computer. The said computer boots normally or resumes from said hibernation. The said transaction terminal loads the reply from the later said non-volatile memory and resumes the transaction, which is outside the scope of the process. The server side of the system would receive the said transaction terminal request, including the said encrypted PIN or the said cryptogram or the said digital signature, depending on whether the said PIN was encrypted or was used to create a cryptogram or a digital signature. The said request also optionally includes the said MAC. The said server side also receives the said encrypted other secretes and decrypts it. The said server side processes the request as normal PIN based transactions relying on the whole sale PIN security standards. If the said user entered the said other secret, the said server side would decrypt it and update the system with the new secret. The transaction processing is done using the established transaction processing processes for processing PIN based transactions.

Process according to the invention can be summarized as:

1. Transaction Data is prepared by software on the un-trusted operating environment.

2. Transaction Data is kept in a non-volatile memory accessible to both the un-trusted environment and trusted secure environment.

3. Computer is restarted or turned off or reset or hibernated or hibernated with auto-restart; this guarantees that any software, including malicious software, in the computer's memory is Inactive.

4. The computer is booted using the invention apparatus into a special secure, trusted environment.

5. Being in the secure trusted environment, the critical security tasks related to the transaction, which were previously considered not secure to be done due to the possibility of presence of malicious software, can now be done without any fear or worry from malicious software.

6. The updated transaction data, or at least the changes occurred to it stored back in a non-volatile memory; the normal RAM can also be used in this step if chain-booting is going to be used in the next step, because chain booting does not reset or clear the RAM of the computer.

7. The computer is either chain-booted or rebooted back to the normal un-trusted operating environment.

8. The updated transaction data is loaded and used to resume the transaction.

Also, with reference to figures 7, 8, 9 and 10 the provision of a device to enable secure booting is provided comprising a contribution of physical media and software. The device provides a bootable source which is a combination of a physical part and a software part; it is either a combination of a physical media and software written on that media, e.g. a bootable CD or a bootable floppy, OR a combination of a physical device and software residing inside that physical device, e.g. a bootable USB memory stick or a bootable memory card

The invention apparatus is a bootable source whose physical part has two distinguishable characteristics and whose software part has another two distinguishable characteristics. If any of these four characteristics, whether physical or logical, is

missing in the bootable source, the whole value of the apparatus, which is Securing Electronic Transactions that are made on Un-trusted Operating Environments, becomes lost. Additionally, the package which includes the physical part combined with the software part must be provided by a trusted party to the user, through a trusted physical delivery chain.

The two distinguishable characteristics of the physical part:

1- Show evidence of sufficient anti-counterfeit features, e.g. printed artwork, holograms, logos, special shape of the media or device itself, artwork or holographic images that become visible by warmth through for example finger pressing for a short while, printing of special trade marks…etc.

2- Contain an unchangeable area, which is an area that the data written on it cannot be changed by an attacker or by mistake after the bootable source is packaged, i.e. the physical and the software part are combined. For media, this can be easily realized by using a media technology that is already tamper immune, like for example a CD-ROM, or a write once media that can be fully written or at least prohibit new sessions to replace the sessions in the unchangeable area. For devices, the device itself must be tamper proof, i.e. the device becomes unusable if tampered, or at least tamper evident, i.e. when tampered the user can easily see or distinguish that it is tampered, hence he will not use it because he knows that the unchangeable area might have been tampered. Of course the device should not allow commands sent to the device through its external interfaces to make any changes in the unchangeable area. Temporary or transient data, whether volatile or non-volatile, can still be stored inside the device in a memory area other than the unchangeable area. The unchangeable area does not necessarily need to be implemented as a ROM chip that is never changeable; the manufacturer may for example allow this area to be securely overwritten with a newer version of the critical software that should be kept in the unchangeable area. There are a number of technologies that allow for this secure upgrade, like for example checking the digital signature that is on the software, inside the device itself, before making the overwriting.

The two distinguishable characteristics of the software part:

1- Contain minimalist secure operating environment that resides in the unchangeable area and allows only trusted software to execute; i.e. software that is either on the unchangeable area also, or is validly digitally signed by a signor accepted by the said minimalist secure operating environment.

2. Run dedicated trusted software that perform security critical operations related to an electronic transaction that is being made on an un-trusted operating environment, rather than performing the transaction completely by itself. This requires the said dedicated trusted software to load transaction data that were initially prepared by software that was running on the said un-trusted operating environment, do some processing on it that involve taking authorization from the user and finally store back the updated transaction data containing the cryptographic non-repudiate-able consent of the user. These updated transaction data are to be used by software on the said un-trusted operating environment to resume making the transaction, by for example sending these secured data to a server and waiting back for a reply.

At figure 7 is shown a USB memory device 200 having a casing/body from which the USB contacts 206 protrude for connection to a USB drive is a PC in a conventional manner. The exterior of the casing has an anti-counterfeit feature 204 which can be for example artwork, logos, holograms etc. The casing/body is held together by tamper proof rivets 202. As an alternative to rivets 202, other forms of tamper proof or tamper evidence technology could be used.

Inside of the casing/body there is USB memory with a non-volatile memory 210 and a controller 208 able to run software within the memory 210.

The non-volatile memory 210 comprises a dedicated reading only area 212. Preferably the dedicated read only area 212 imposes a read only constraint on the controller 208 rather than use a separate read only memory chip.

The non-volatile memory 210 contains a software which is readable by the connected computer in use and enables the computer to run in a secure operating environment.

The software is stored at least in part in the dedicated read only area 212 so that it cannot be interfered with or overwritten by an unauthorized user.

In figure 8 is shown an adapted USB token 250 or USB smartcard. The token 230 comprises a body/casing and USB terminals as with device 200. On the body/casing of token 230 also comprises an LED 230 and tamper proof or a tamper evidence fastening such as rivet 236 (alternatively known USB token's tamper proof technology can be used) and an anti-counterfeit feature 234 such as artwork, logos, holograms etc.

The token 230 is adapted to make it function as two devices at the same time, a connected USB token and a read only memory store. The read only memory can actually be allowed to be updated if the appropriate logic for verifying the signature on the loaded software is implemented or at least require a high security Admin code to change, which is preserved securely with the provider. When Admin codes are used, the updated memory should be used in a secure environment, because there is no digital signature to verify internally inside the USB token 230. A smartcard can be used instead of a token such as by using a modern smartcard which is able to natively talk with the USB protocol, and can connect to a computer through a USB connector with no bits of logic necessary.

An example of a smartcard reader adapted to be performed as the invention apparatus is shown in figure 9. The smartcard reader 240 comprises a main body/casing 241a USB cable 240 and USB interface 250. On the external casing 244 there is again anti-counterfeit features such artwork, logos, holograms etc 242. Tamper evident screw cap 246 which could also be any other form of tamper proof or tamper evidence fastener. The inside of the casing 244 is shown in figure 9b. Inside is a controller 252, EERPOM 258, memory 256, and contacts 254. This is a schematic representation of the conceptual components inside the smartcard reader 240 rather than showing real integrated circuits. As can be seen from the schematic diagram, controller 252 is connected to contact 254 memory 256 EEPROM 258 and the USB interface 250 via a cable 248. The smartcard reader 240 is adapted to the invention by adding more memory to the EEPROM and is supplemented in the USB (read only) disk interface in the controller, in addition to the existing USB smartcard interface as if the device is acting as two devices at the same time.

In figure 10 is shown a CD-R 270 which is shaped in the form of round-edged rectangle. On the exterior of CD-R 270 is anti-counterfeit artwork, logos, holograms etc 272. Additional extra artwork and anti-counterfeit features can be added on the back of the CD-R 270 and even on the unused part of the storage area. The back of the CD-R is shown in figure 10b with the used readable media itself depicted as 274, the surrounding area being more artwork by 272. In the case of the CD-R all the software stored on it can of course be read only.

Claims

1. A client terminal adapted to enable a secure transaction over a communications network, comprising a controller having a first source for booting a normal operating environment which operably enables communication between the client terminal and the network, the controller being adapted to create transaction data in accordance with user operation of the client terminal, to save the transaction data to a specified location, and a second source for booting the client terminal into a secure operating environment having limited operability compared to the normal operating environment but enabling entry by the user of authentication information, such as a PIN, for authenticating the transaction data which is retrievable from the specific location by the client terminal in the secure operating environment, the client terminal being adapted to enable shut down of the normal operating environment and booting of the secure operating environment in which authentication of the transaction data is enabled.

2. A client terminal according to claim 1 comprising a non-volatile memory such as a hard drive, which provides the specified location for storage of the transaction data between shutting down the normal and booting the secure operating environments.

3. A client terminal according to claim 2 comprising a smart card reader/writer to enable storage of the transaction data on a smart card.

4. A client terminal according to any preceding claim wherein the second source is in the form of a software medium, such as a CD or DVD-ROM.

5. A client terminal according to any preceding claim wherein the second source is in the form of a device such as a USB smartcard reader.

6. A client terminal according to any preceding claim wherein the client terminal is a personal computer.

7. A client terminal according to any of claims 1 to 5 wherein the client terminal is a mobile phone.

8. A client terminal according to any preceding claim wherein the secure operating environment is configured not to enable any networking capability of the client terminal when in the secure operating environment.

9. A client terminal according to any preceding claim wherein the secure operating environment only allows use of software from the second source and/or trusted source established from a directory reference, digital signature, or other mechanism mentioned herein.

10. A method of enabling a secure transaction over a network using a transaction terminal having a normal operating environment(s) potentially infected with malicious code or devices, the method comprising the steps of:

a) commencing the transaction in a normal operating mode thereby to generate transaction data representative of the user activity and to store the transaction data to a specified location,

b) terminating the normal operating environment,

c) booting the transaction terminal into a secure operating environment having predetermined operability less than a normal operating environment,

d) enabling authentication of the transaction by the user by entry of secret information, such as a signature or PIN, and

e) termination of the secure operating environment and rebooting of a normal operating environment thereby enabling completion of the transaction over the network.

11. A method according to claim 10 using a non-volatile memory such as a hard drive, which provides the specified location for storage of the transaction data between shutting down the normal and booting the secure operating environments.

12. A method according to claim 10 or 11 using a smart card reader/writer to enable storage of the transaction data on a smart card.

13. A method according to claim 10, 11, or 12 comprising the step of using a second source storage of the software for the secure operating mode, which second source is in the form of a software medium, such as a CD or DVD-ROM.

14. A method according to any of claims 10 to 13 wherein the second source is in the form of a device such as a USB smartcard reader and the method includes booting the secure operating environment from the second source.

15. A method according to any of claims 10 to 14 using a personal computer.

16. A method according to any of claims 10 to 15 using a mobile phone.

17. A method according to any preceding claim wherein the secure operating environment is configured not to enable any networking capability of the client terminal when in the secure operating environment.

18. A method according to any preceding claim wherein the secure operating environment only allows use of software from the second source and/or trusted source established from a directory reference, digital signature, or other mechanism mentioned herein.

19. A method according to claim any preceding claim wherein the transaction terminal comprises a hard drive and the transaction data is saved to a specified location on the hard drive, preferably in the last 30 sectors on track 0.

20. A computer readable medium comprising data which operably enables a computer to operate in a secure operating environment.

21. A computer readable medium according to claim 20 comprising data which enables a computer according to any of claims 1 to 9 and/or a method of operating a computer according to any of claims 10 to 19any of the

22. A transaction terminal adapted to enable secure entry of secret information through an input with reduced risk of logging of the information, comprising a controller, an output and a keyed input, the terminal being adapted to present the user via the output with an altered configuration of the keyed input, thereby enabling the user to enter the secret information using the keyed input using the change configuration and the controller being adapted to interpret the keyed data representative of the secret information from the altered configuration.

23. A transaction terminal as described in claim 22 wherein the controller is adapted randomly to change the configuration of the keyed input.

24. A device for enabling a computer to be booted into a secure operating environment, comprising a body for carrying a storage medium for storing software which software is readable by computer in use and enables the computer to run in a secure operating environment, wherein the body comprising anti counterfeit feature and/or a tamper-evident feature, and the storage medium comprises a special area which cannot be overwritten by an unauthorised user, wherein the software is stored at least in part in the special area.

25. A process of enabling secure transactions comprising the following steps:

(a) preparing (generation/gathering) transaction data and storing it on one or more of the following, hard disc, special memory on the smartcard reader, smartcode or token, NVRAM of the motherboard, USB memory, a writable non-volatile memory area on a special bootable source, manual storing by the user, even outside the computer through remembering or printing or writing down on a piece of paper or a combination thereof;

(b) terminating the normal operating system using any of the following, power-off, reset, shutdown, restart, hibernate, hibernate followed by reset rather than power-off;

(c) booting from an adapted media or device provided by a trusted party. The adapted media device for example being adapted read-only or write once, optical media such as CD, CD-R, DVD-R or DVD+R, an adapted smartcard reader, an adapted USB memory or an adapted USB smartcard or token;

(d) performing security critical operations related to the electronic transaction prepared in step (a) through performing any one or more of the following, secure entry of the user secrets, secure capture of user biometrics, secure verification of electronic signatures, secure presentment of the transaction data to the user, secure sending of operations to smartcards and tokens, secure performing of cryptographic operations on the computer's processor and memory, generating message authentication codes, generating one time passwords and transaction certificates, loading transaction data from specific non-volatile memories, storing processed/amended transaction data;

(e) booting back the normal operating environment thereby enabling completion of the said electronic transaction such as by restart or chain booting.

26. A process according to claim 25 wherein the step of booting from an adapted media or device, uses a device adapted in one or more of the following ways: imposing sufficient, yet deliberate, physical ant-counterfeit features; ensuring the imposing presence of a physically difficult to tamper area of storage or memory (for

example read only storage, write controlled memory in a tamper proof or tamper evident device), preferably realized either through deliberate manufacturing or through choosing the right technology or a combination thereof; the software in the said difficult to tamper area is a minimalist secure bootable environment that execute trusted code only; code can be considered secure if it is loaded from the same source as the secure environment, or the physically difficult to tamper area, or has valid signature created by a signatory trusted by the secure environment; the said secure environment run trusted dedicated software for performing security critical operations related to an electronic transactions being made outside the said secure environment.

27. A process according to claim 26 wherein the adaptation ensures that the media or device:

contains a difficult to copy area that can serve as "something that you have" authentication factor; such as an optical media with special copy protection technology, or a PIN protected memory area on a device like a USB memory, or a USB token or smartcard that by virtue have this capability, and/or

contains software that runs, or auto-runs, on the normal operating environment of the user to facilitate the realization of steps performed on the normal operating environment, like for example steps (a) and (b), preferably this software does not reside on the said difficult to tamper area because it already runs on an operating environment suspicious of malicious code.

Figure 1

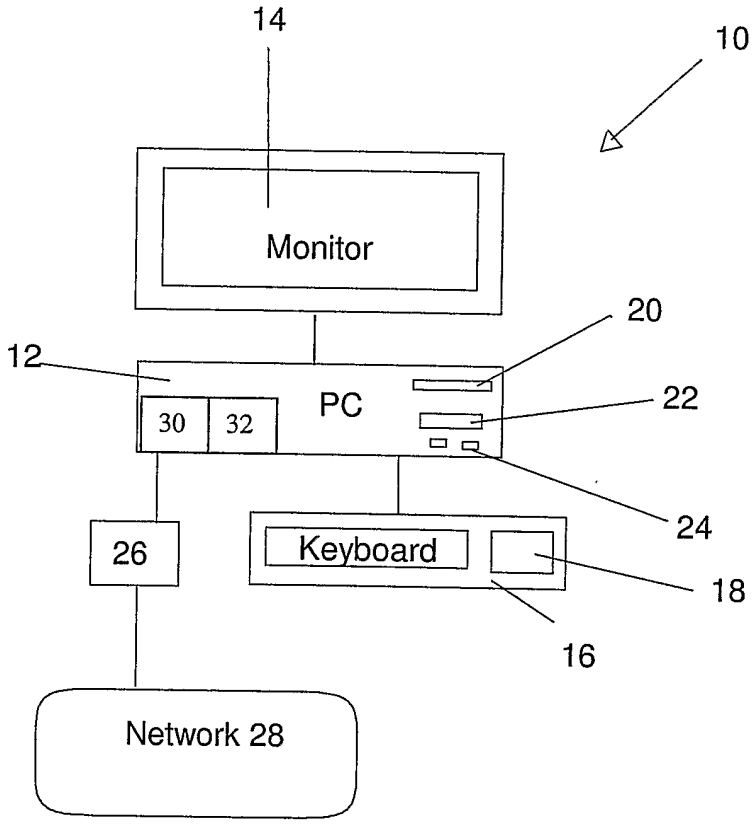


Figure 2

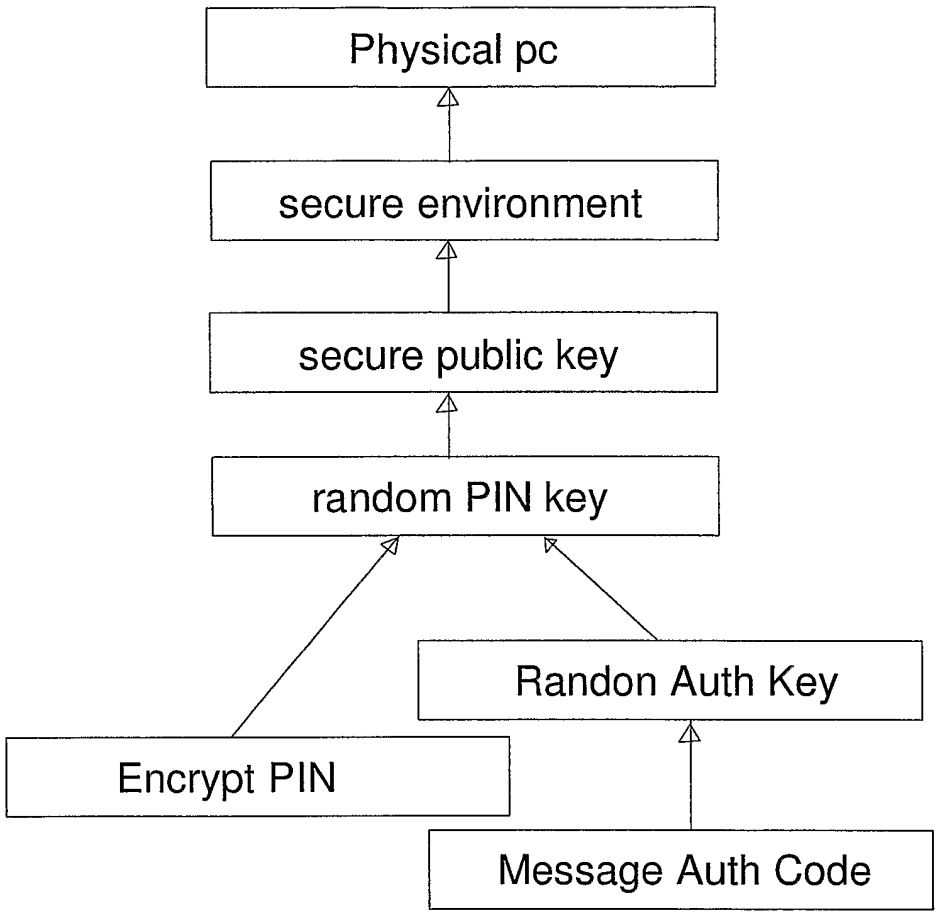

Physical pc
secure environment
secure public key
random PIN key
Randon Auth Key
Encrypt PIN
Message Auth Code

Figure 3

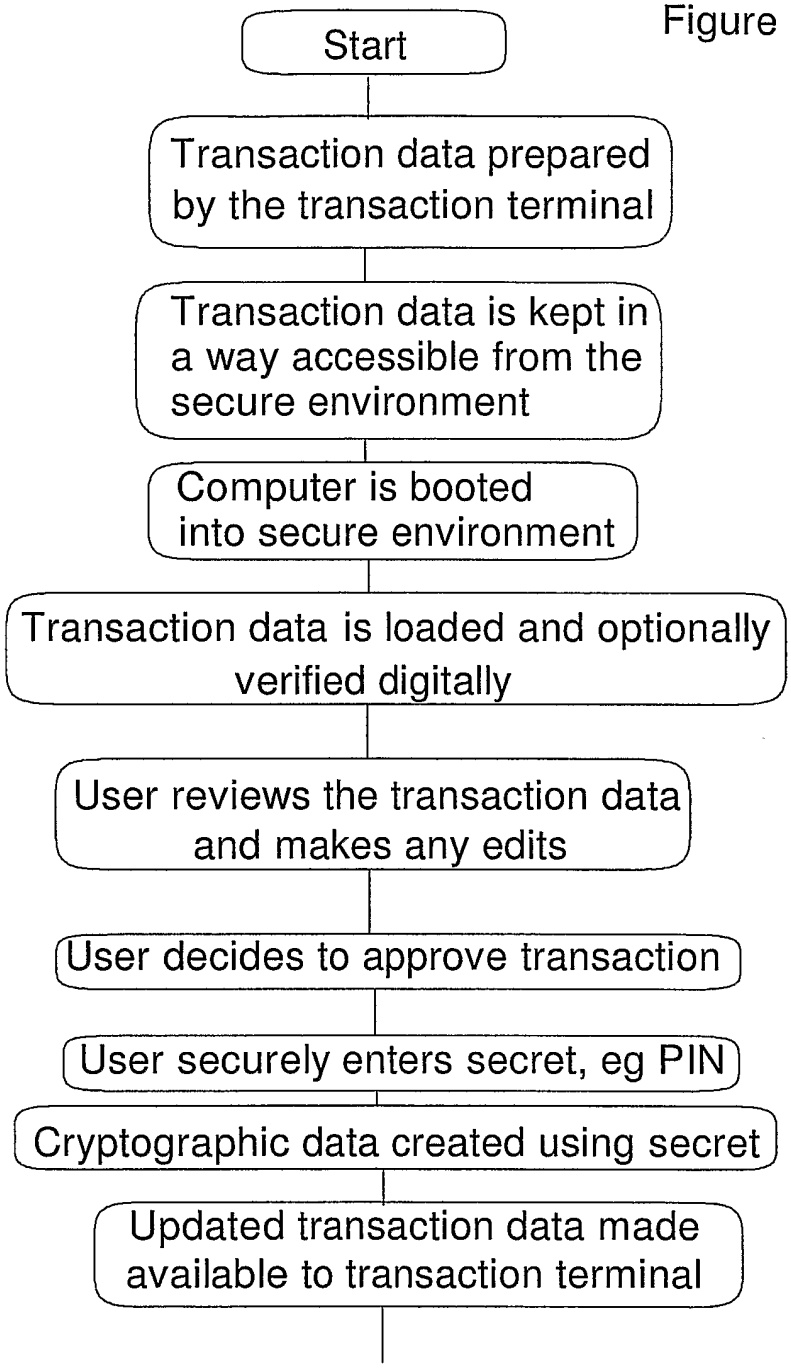

Start
Transaction data prepared by the transaction terminal
Transaction data is kept in a way accessible from the secure environment
Computer is booted into secure environment
Transaction data is loaded and optionally verified digitally
User reviews the transaction data and makes any edits
User decides to approve transaction
User securely enters secret, eg PIN
Cryptographic data created using secret
Updated transaction data made available to transaction terminal

Figure 4

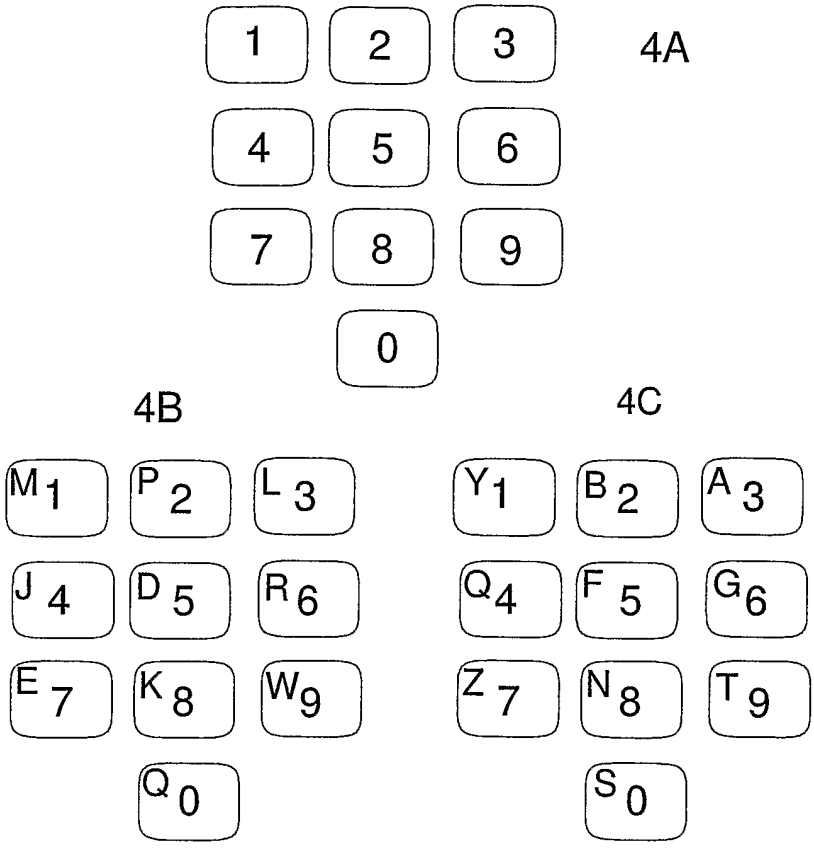

1
2
3
4A
4
5
6
7
8
9
0
4B
4C
M 1
P 2
L 3
Y 1
B 2
A 3
J 4
D 5
R 6
Q 4
F 5
G 6
E 7
K 8
W 9
Z 7
N 8
T 9
Q 0
S 0

FIGURE 5

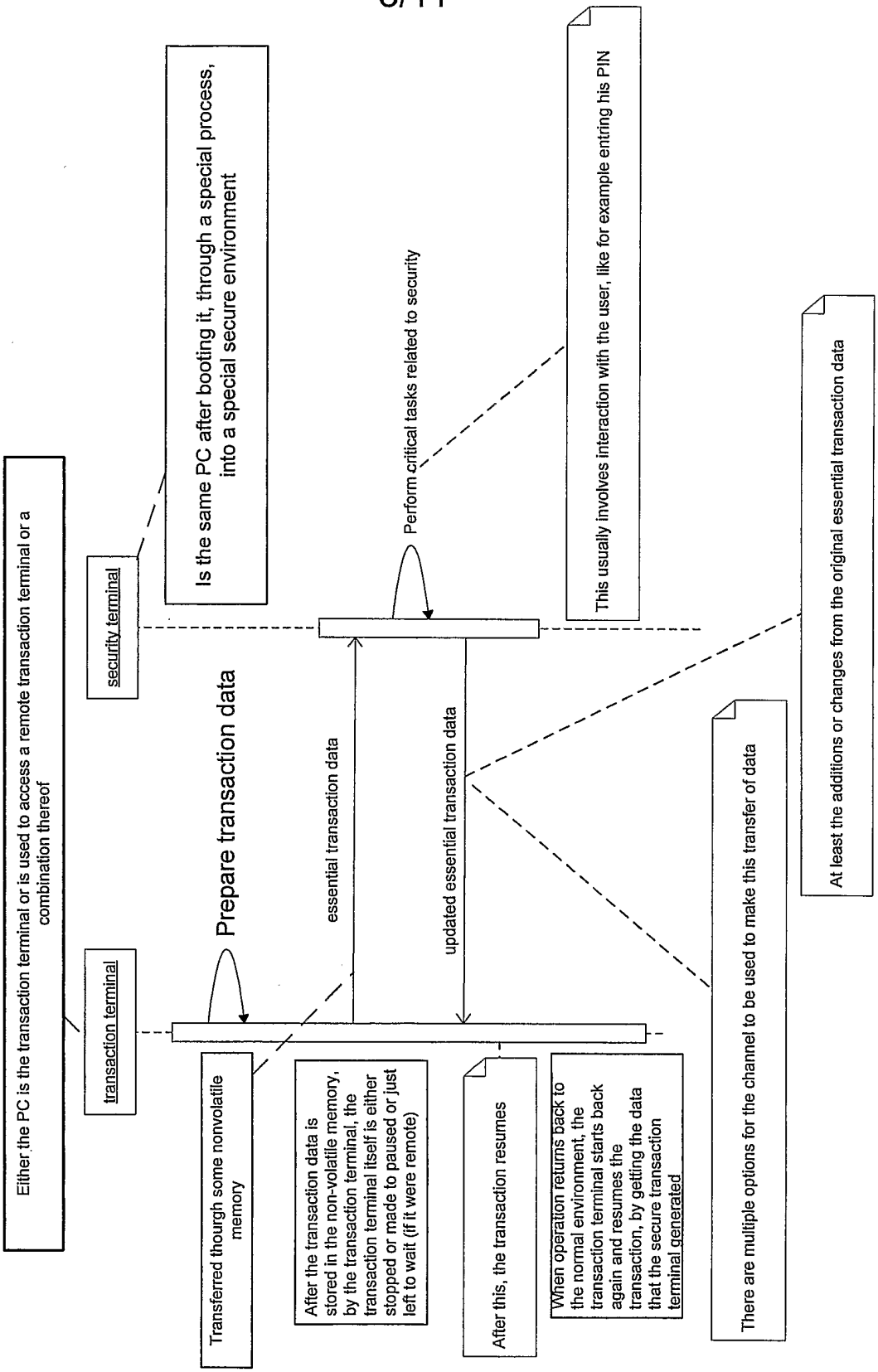

Either the PC is the transaction terminal or is used to access a remote transaction terminal or a combination thereof
transaction terminal
security terminal
Is the same PC after booting it, through a special process, into a special secure environment
Prepare transaction data
Perform critical tasks related to security
This usually involves interaction with the user, like for example entring his PIN
essential transaction data
updated essential transaction data
At least the additions or changes from the original essential transaction data
There are multiple options for the channel to be used to make this transfer of data
Transferred thourgh some nonvolatile memory
After the transaction data is stored in the non-volatile memory, by the transaction terminal, the transaction terminal itself is either stopped or made to paused or just left to wait (if it were remote)
After this, the transaction resumes
When operation returns back to the normal environment, the transaction terminal starts back again and resumes the transaction, by getting the data that the secure transaction terminal generated

FIGURE 6

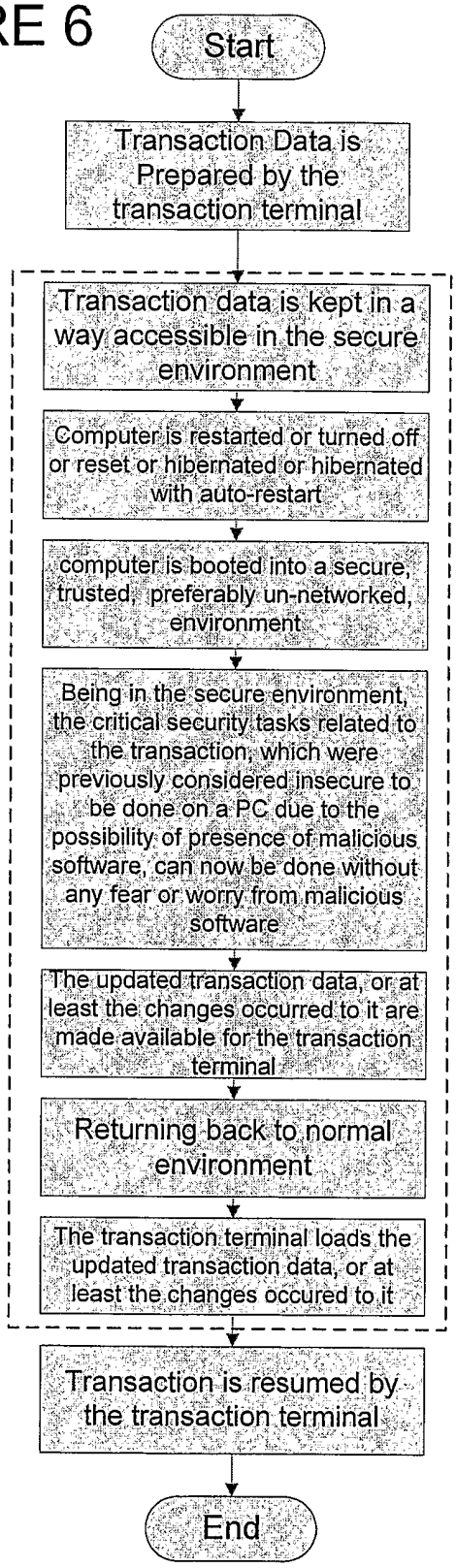
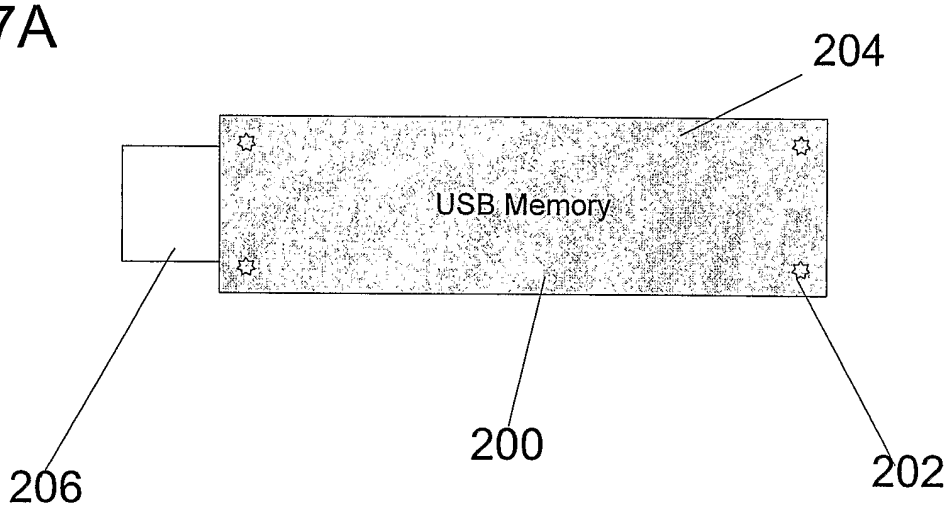

Start
Transaction Data is Prepared by the transaction terminal
Transaction data is kept in a way accessible in the secure environment
Computer is restarted or turned off or reset or hibernated or hibernated with auto-restart
computer is booted into a secure, trusted, preferably un-networked, environment
Being in the secure environment, the critical security tasks related to the transaction, which were previously considered insecure to be done on a PC due to the possibility of presence of malicious software, can now be done without any fear or worry from malicious software
The updated transaction data, or at least the changes occurred to it are made available for the transaction terminal
Returning back to normal environment
The transaction terminal loads the updated transaction data, or at least the changes occured to it
Transaction is resumed by the transaction terminal
End

FIGURE 7

7A



7B

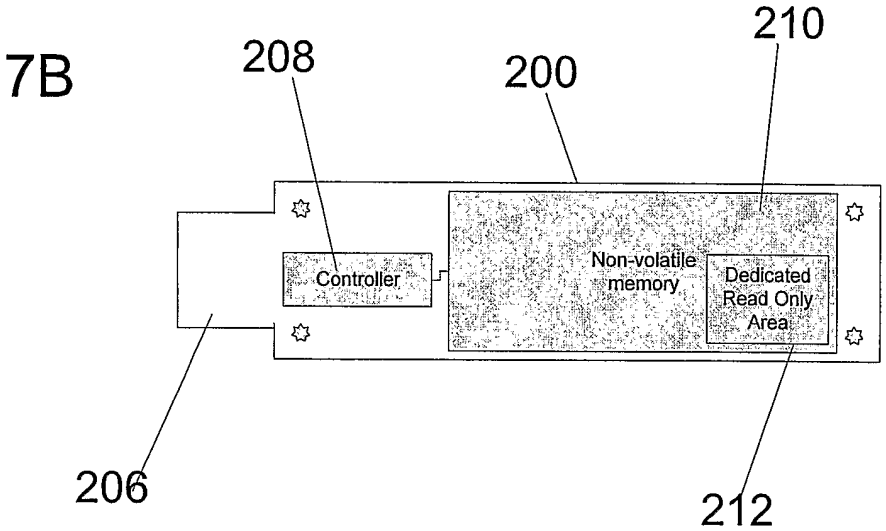


FIGURE 8

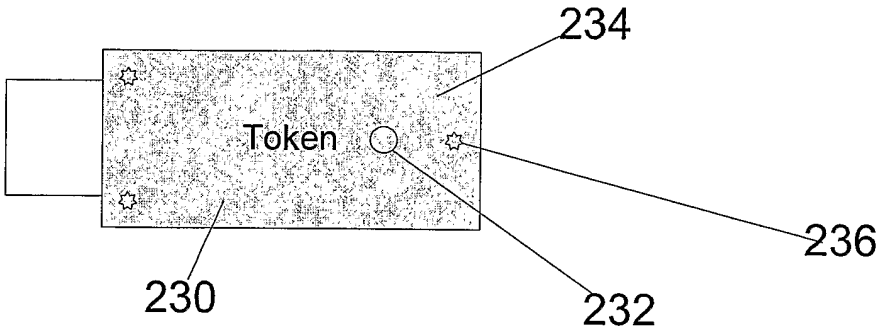
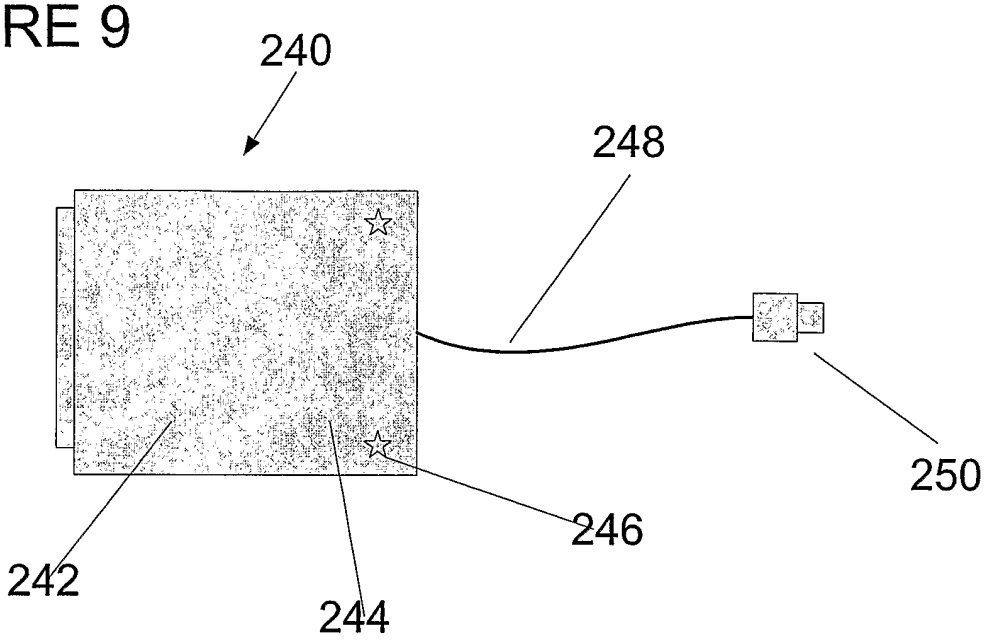


FIGURE 9

9A



9B

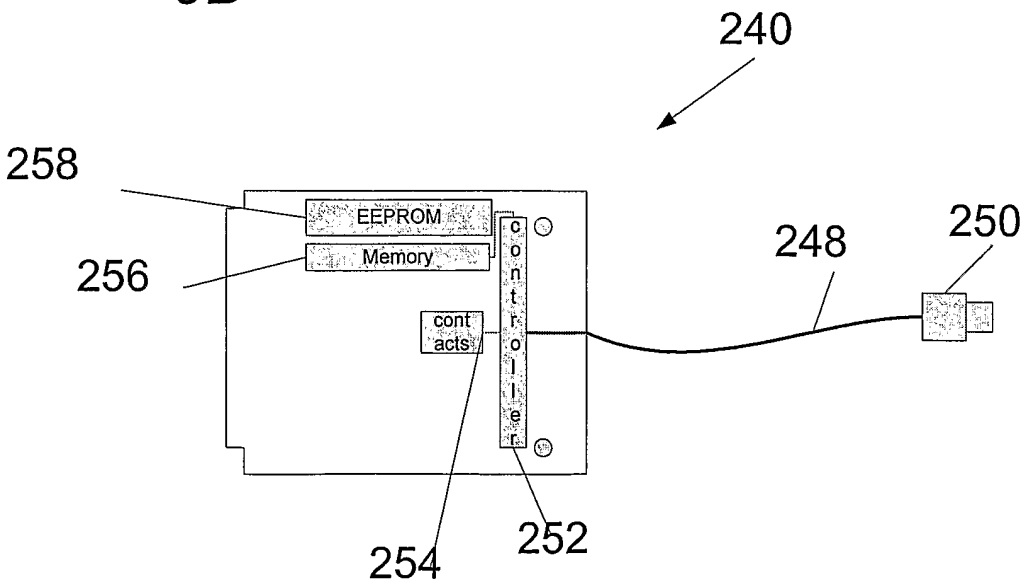
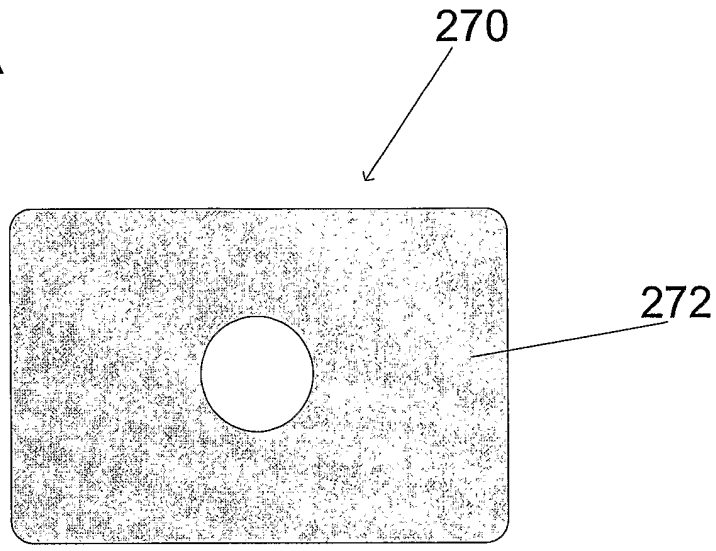


FIGURE 10

10A



10B

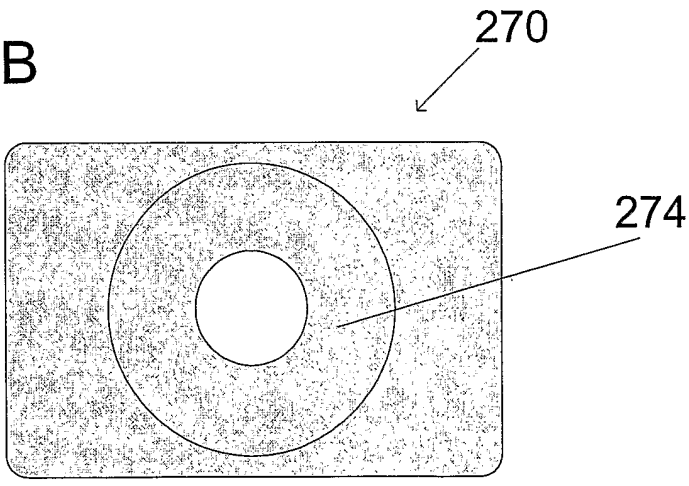
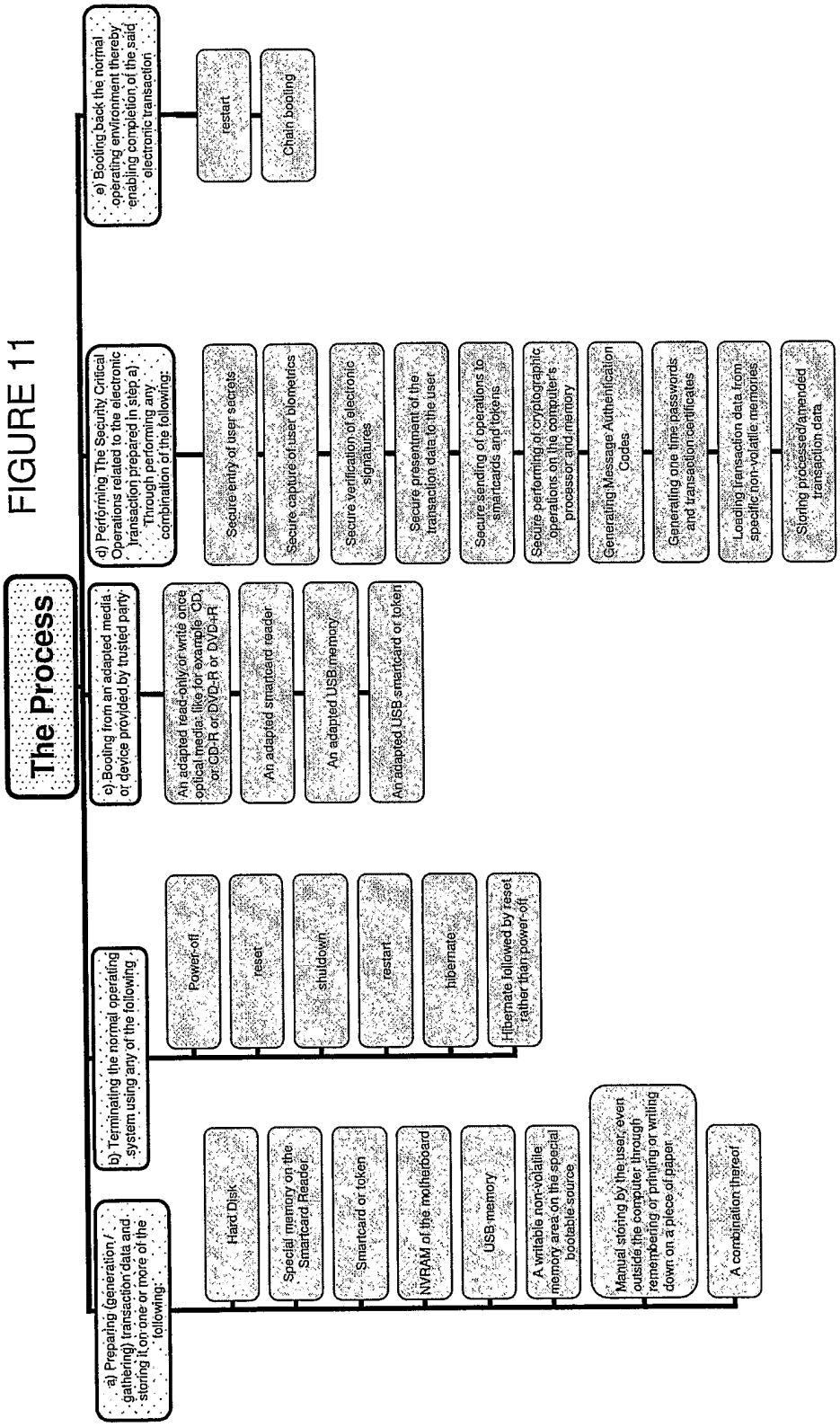


FIGURE 11


The Process
a) Preparing (generation / gathering) transaction data and storing it on one or more of the following:
Hard Disk
Special memory on the Smartcard Reader
Smartcard or token
NVRAM of the motherboard
USB memory
A writable non-volatile memory area on the special bootable source
Manual storing by the user, even outside the computer, through remembering or printing or writing down on a piece of paper
A combination thereof
b) Terminating the normal operating system using any of the following
Power-off
reset
shutdown
restart
hibernate
Hibernate followed by reset rather than power-off
c) Booting from an adapted media or device provided by trusted party
An adapted read-only or write once optical media: like for example CD-R or CD-R or DVD-R or DVD+R
An adapted smartcard reader
An adapted USB memory
An adapted USB smartcard or token
d) Performing The Security Critical Operations related to the electronic transaction prepared in step a) Through performing any combination of the following:
Secure entry of user secrets
Secure capture of user biometrics
Secure verification of electronic signatures
Secure presentment of the transaction data to the user
Secure sending of operations to smartcards and tokens
Secure performing of cryptographic operations on the computer's processor and memory
Generating Message Authentication Codes
Generating one time passwords and transaction certificates
Loading transaction data from specific non-volatile memories
Storing processed/amended transaction data
e) Booting back the normal operating environment thereby enabling completion of the said electronic transaction
restart
Chain booting

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB2005/001770

A. CLASSIFICATION OF SUBJECT MATTER
G06F1/00 H04L29/06

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 085 396 A (HEWLETT-PACKARD COMPANY) 21 March 2001 (2001-03-21) paragraphs [0004] - [0006], [0062], [0069], [0072], [0081], [0095]	1-21, 25-27
A	WO 02/01520 A (COVADIS S.A; HILLION, HERVE) 3 January 2002 (2002-01-03) page 4, line 39 - page 6, line 24 page 23, line 25 - page 24, line 31	1-21, 25-27
A	US 2001/056411 A1 (LINDSKOG HELENA ET AL) 27 December 2001 (2001-12-27) the whole document	1-21, 25-27

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

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

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 September 2005	24. 11. 2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Veillas, E

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2005/001770

Box II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

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

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

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

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

Box III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

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

see additional sheet

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

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

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-21, 25-27

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-21, 25-27

   A method and a system where two operating environments are used during a transaction, wherein one is a secure operating environment being booted for retrieval of a PIN

   ---

2. claims: 22-23

   A transaction terminal comprising a keyed input and an output on which an altered configuration of the keyed input is presented.

   ---

3. claim: 24

   A device for enabling a computer to be booted in a secure operating environment comprising anti-counterfeit feature and comprising a storage area which can not be overwritten by an unauthorized user.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB2005/001770

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1085396 A	21-03-2001	WO 0127722 A1	19-04-2001
		JP 2003511783 T	25-03-2003
WO 0201520 A	03-01-2002	AU 5659101 A	08-01-2002
		WO 0201522 A1	03-01-2002
US 2001056411 A1	27-12-2001	AU 7245501 A	17-12-2001
		CN 1446329 A	01-10-2003
		WO 0195070 A2	13-12-2001
		EP 1314077 A2	28-05-2003