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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 January 2007 (04.01.2007)

PCT

(10) International Publication Number
WO 2007/001237 A2

(51) International Patent Classification:
G06F 1/00 (2006.01) *H04L 9/12* (2006.01)
G06F 21/04 (2006.01)

(21) International Application Number:
PCT/SG2006/000110

(22) International Filing Date: 28 April 2006 (28.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
200504060-5 25 June 2005 (25.06.2005) SG

(71) Applicant *(for all designated States except US)*: **KRYPT TECHNOLOGIES** [SG/SG]; Block 12, Toh Yi Drive, #05-383, Singapore 590012 (SG).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **ENG, Kern Shen** [SG/SG]; Block 12, Toh Yi Drive, #05-383, Singapore 590012 (SG).

(74) Agent: **FOO, Moo Kwang**; Axis Intellectual Captital Pte Ltd, 21A Duxton Road, Singapore 089487 (SG).

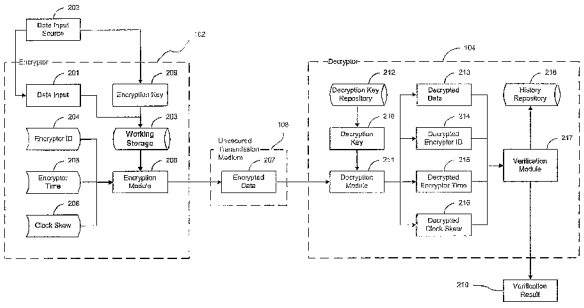
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, 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, LV, 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:
— *without international search report and to be republished upon receipt of that report*

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

(54) **Title:** ENCRYPTION SYSTEM FOR CONFIDENTIAL DATA TRANSMISSION



(57) **Abstract:** The ubiquitous and borderless Internet has greatly increased connectivity between people. Together with its popularity as a commercial activities platform, the Internet has also become a popular target for cyber criminals who intentionally exploit on the Internet's numerous vulnerabilities. Perpetrators exploit the vulnerability of computer systems with malicious software programs designed specifically to steal confidential information such as user names, passwords and credit card numbers by recording keystrokes from the computer keyboard. The two most common methods used are keystroke recording and phishing scams. Phishing is a means of fraudulently acquiring confidential information through deception, for example, by masquerading as an official email or website for requesting such information. Online services relying on password challenges alone for authentication are the most vulnerable to keystroke recording as well as to other means of obtaining passwords. Multi-conditional authentication can be deployed to replace Password Challenge. However, these solutions incur high implementation cost and bring forth privacy issues. Embodiments of the invention describe an encryptor and a method for implementation thereof for encrypting data input and a dynamic variable obtainable by the encryptor into encrypted data. The encryptor data communicatively interfaces an input device and a computing system wherein a user is able to interact with the input device for generating the data input. The encrypted data is transmitted to and relayed by the computer system for subsequent reception and decryption by a decryptor for comparing with a reference dynamic variable obtainable by the decryptor for verifying validity of data decrypted therefrom.

WO 2007/001237 A2

Encryption System for Confidential Data Transmission

Field of Invention

This invention generally relates to data security within a system. In particular, it relates to a system and a process for ensuring confidentiality, integrity and validity of data during transmission thereof.

Background

The ubiquitous and borderless Internet has greatly increased connectivity between people, allowing its users to work, play and communicate with one another across the globe 24 hours a day and 7 days a week. Together with its popularity as a commercial activities platform, the Internet has also become a popular target for cyber criminals who intentionally exploit on the Internet's numerous vulnerabilities. The general purpose computer system demonstrates one of these vulnerabilities as it readily carries out both instructions intended by the user, as well as malicious instructions unintended by and unknown to the user.

Perpetrators exploit the vulnerability of computer systems with malicious software programs such as Trojan horses, viruses, worms and spywares. These malicious software programs, also known as malwares, have been typically designed to operationally disrupt computer systems and networks. However in recent years, there has been a rising trend of malwares being designed specifically to steal confidential information such as user names, passwords and credit card numbers by recording keystrokes from the computer keyboard.

Through the survey of 5,000 U.S. adults, Gartner, Inc estimated that in the 12 months ending April 2004, nearly 2 million Americans have fallen victim to bank account fraud. The cost to banks and consumers is a staggering US $2.4 billion in direct losses, an average of US $1,200 per victim. The two most common methods used to lift bank account numbers, user names and passwords are keystroke recording and phishing scams. Phishing is a means of fraudulently acquiring confidential information through deception,

for example, by masquerading as an official email or website for requesting such information.

Malware scanners are the most commonly used countermeasures to detect and remove malwares. However, such measures have their limitations. Malwares that are new and unregistered in the scanner's database or keystrokes recording functions embedded in legitimate software applications may not be detectable by scanners. Furthermore, since malware scanners are also software applications, it is possible for malwares to disable or even infect malware scanners. Also, malware scanners are helpless against both phishing scams and hardware based keystroke recording device.

Online services relying on password challenges alone for authentication are the most vulnerable to keystroke recording as well as to other means of obtaining passwords. Multi-conditional authentication, based on the "What You Know", "What You Have" and "What You Are" principles, for example through the use of a combination of devices such as Smart Card, USB token, One-Time-Password generator and Biometric sensor, can be deployed to replace Password Challenge. However, these too have their shortcomings. Firstly, these solutions incur high implementation cost which is difficult for many online services to justify. Online services with a large user base, such as online banking, can easily incur millions of dollars just to issue the security tokens.

Secondly, user-specific identification brings forth privacy issues. Smart cards, USB tokens and especially biometrics system introduce means to uniquely identify individuals. Even if these systems are not explicitly designed to support individual surveillance or law enforcement, users will always fear that someday, their information may be exploited for such purposes. Furthermore, once a biometric attribute has been compromised, it is lost forever. Hence, subscribing to online service with fingerprinting is not recommended as either the service providers can be compromised by cyber criminals or the providers themselves can be fraudulent.

United States Patent No. 5,596,718 (Boebert) describes a method and apparatus for ensuring confidentiality and integrity of data input from the computer keyboard by inserting a trusted path subsystem between the keyboard and the computer system. The trusted path subsystem employs end-to-end encryption whereby data is encrypted as close to the source as possible and decrypted only at its final destination. Such end-to-end encryption prevents confidential data input from being captured by malwares. However, the importance of the encryption process that determines the eventual usability of the invention is being overlooked here.

The invention specifies the use of either an asymmetric or symmetric cryptographic technique for encrypting confidential data input from the keyboard. However, Boebert overlooks the fact that cryptographic techniques only convert plaintext data into a fixed encrypted data. Although the confidentiality and integrity of the data is ensured, the fixed encrypted data can be captured and exploited by perpetrators. In authentication application, perpetrators can easily capture this encrypted data and replay the encrypted data at a later time to gain access to a secured system.

United States Patent No. 5,406,624 (Tulpan) describes a security unit which isolate the unsecured computer system from its keyboard. Similarly, the security unit is inserted between the keyboard and the computer system. This allows security functions of the security unit to be operated in isolation from the unsecured computer system. However, this does not prevent the encrypted data from interception and replay by perpetrators.

United States Patent No. 6,134,661 (Topp) describes an apparatus for encrypting password for network authentication. Each functionally distinct apparatus, either a keyboard or portable keycard, possesses a unique encryption function. To gain access to a network, a user has to use the unique apparatus with its corresponding password to log onto the network. However, similar to both Boebert and Tulpan, this invention does not prevent the encrypted data from interception and replay by perpetrators.

Hence this clearly affirms the need for a system and a process for ensuring confidentiality and integrity of data transmitted from the input source to the final destination, and ensuring that confidential data cannot be replayed to carry out fraudulent process.

Summary

In accordance with a first aspect of the invention, there is disclosed an encryptor for encrypting data input comprising an encryption module for data communicatively interfacing an input device and a computing system and for obtaining a dynamic variable. A user interacts with the input device for generating the data input with the input device being in direct data communication with the encryption module. The encryption module encrypts data input and the dynamic variable into encrypted data using an encryption algorithm. The encrypted data is transmitted to and relayed by the computer system for subsequent reception and decryption by a decryptor. The decryptor decrypts the encrypted data into decrypted data and a decrypted dynamic variable wherefrom the decrypted dynamic variable is comparable with a reference dynamic variable obtainable by the decryptor for verifying validity of the decrypted data. The decrypted data is representative of the data input when the decrypted dynamic variable corresponds with the dynamic variable and the reference dynamic variable.

In accordance with a second aspect of the invention, there is disclosed a decryptor for decrypting encrypted data of a data input comprising a decryption module for decrypting encrypted data into decrypted data and a decrypted dynamic variable using a decrypting algorithm. The encrypted data is subsequently received and decrypted into the decrypted data and the decrypted dynamic variable. The encrypted data is encrypted from a data input. The decrypted dynamic variable is compared with a reference dynamic variable obtained by the decryptor for verifying validity of the decrypted data whereby the decrypted data is representative of the data input when the decrypted dynamic variable corresponds with a dynamic variable and the reference dynamic variable.

In accordance with a third aspect of the invention, there is disclosed a process comprising the steps of receiving encrypted data being encrypted from a data input and a dynamic variable. In addition, steps of decrypting the encrypted data with a decryptor into decrypted data and a decrypted dynamic variable and verifying validity of the decrypted data by comparing the decrypted dynamic variable with a reference dynamic variable obtained by the decryptor are also disclosed. The decrypted data is representative of the data input when the decrypted dynamic variable corresponds with the dynamic variable and the reference dynamic variable.

In accordance with a fourth aspect of the invention, there is disclosed an encryptor for encrypting data input comprising: an encryption module for encrypting data input, a clock skew and a timestamp into encrypted data using an encryption algorithm, the data input being introducible by a user; and a clock for generating the clock skew and the timestamp, wherein the encrypted data is subsequently receivable and decryptable by a decryptor into decrypted data, a decrypted clock skew and a decrypted timestamp, at least one of the decrypted clock skew and the decrypted timestamp being comparable with a reference timestamp obtainable by the decryptor for verifying validity of the decrypted data, and whereby the decrypted data is representative of the data input when the decrypted clock skew and the decrypted timestamp corresponds with the clock skew and timestamp and the reference clock skew and the reference timestamp.

In accordance with a fifth aspect of the invention, there is disclosed an encryptor for encrypting data input comprising: an encryption module for encrypting data input introducible by an user and a time dependent value into encrypted data using an encryption algorithm and a user encryption key, the data input and the user encryption key being provided by a user; and a clock for generating the time dependent value, wherein the encrypted data is subsequently receivable and decryptable using a decryption algorithm and a decryption key by a decryptor into decrypted data and a decrypted time value, the decrypted time value being comparable with a reference time value obtainable by the decryptor for verifying validity of the decrypted data, and the decryption key being

associated with the user decryption key, and whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the time dependent value and the reference time value.

In accordance with a sixth aspect of the invention, there is disclosed an encryptor for encrypting data input comprising: an encryption module for encrypting data input introducible by a user into encrypted data using an encryption algorithm and a user input encryption key; and a user input device, the user interacting with the user input device to thereby enter the data input into the encryptor, wherein the encrypted data is subsequently receivable and decryptable using a decryption algorithm and a decryption key by a decryptor into decrypted data and whereby the decrypted data is representative of the data input with the decryption key being associated with the user encryption key, and whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the time dependent value and the reference time value.

In accordance with a seventh aspect of the invention, there is disclosed an encryptor for encrypting data input comprising: an encryption module for encrypting data input and a clock skew into encrypted data using an encryption algorithm, the data input being introducible by a user; and a clock for generating the clock skew, wherein the encrypted data is subsequently receivable and decryptable by a decryptor into decrypted data and a decrypted time value, the decrypted time value being comparable with a reference time value obtainable by the decryptor for verifying validity of the decrypted data, and whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the clock skew and the reference time value.

Brief Description of The Drawings

Embodiments of the invention are described hereinafter with reference to the drawings, in which:

Fig. 1 illustrates a secure system with the use of an encryptor and a decryptor for confidential data transmission through an unsecured transmission medium according to first and second embodiments of the invention;

Fig. 2 is a flow diagram illustrating a partial data flow of components within the encryptor and the decryptor of Fig. 1;

Fig. 3a is a flow diagram illustrating an encryption process of the encryptor of Fig. 1;

Fig. 3b is a flow diagram further illustrating retrieval of encryption key from a repository module for the encryption process according to Fig. 3a;

Fig. 4a is a flow diagram illustrating a verification process of the decryptor of Fig. 1;

Fig. 4b is a flow diagram illustrating a feedback challenge process between the encryptor and the decryptor of Fig. 1;

Fig. 4c is a flow diagram further illustrating a feedback verification process within the decryptor of Fig. 1; and

Fig. 5a illustrates components of the encryptor of Fig. 1;

Detailed Description

With reference to the drawings, a system and a process according to embodiments of the invention for implementing data security within a system are described hereinafter for encrypting data so as to ensure confidentiality, integrity and validity of data on the system.

For purposes of brevity and clarity, the description of the embodiments of the invention is limited hereinafter to applications relating to data encryption. This however does not

preclude the invention from other areas of application that requires similar operating functions and performance for establishing data security within a system.

Embodiments of the invention are described in greater detail hereinafter for a system and a process for ensuring confidentiality, integrity and validity of data input via computer peripheral devices and its transmission over unsecured mediums such as the computer system and Internet. Furthermore, the embodiments of the invention are for ensuring that encrypted data cannot be replayed to carry out fraudulent operations. In the detailed description provided hereinafter and illustrations provided in Figs. 1 to 5 of the drawings, like elements are identified with like reference numerals.

With reference to Fig. 1, a diagram of a secure system 100 of a preferred aspect of the invention is shown. Confidentiality and integrity of data is ensured by employing end-to-end encryption whereby data is encrypted as close to the source as possible and decrypted only at its final destination. The secure system 100 employs end-to-end encryption whereby one end of the end-to-end encryption is an encryptor 102 with an encryption module and the other end is a decryptor 104. Confidential data input (not shown in Fig. 1) entered through an input device 106, such as a computer keyboard, is captured by the encryptor 102 and encrypted before it is transmitted over an unsecured transmission medium 108, such as a client application 109 running on an unsecured computer system 110 and an unsecured network 112, to a secured computer system 113. As confidential data is encrypted before it is being transmitted via the unsecured transmission medium 108, its confidentiality and integrity is ensured because malwares residing on the unsecured computer system 110 or the unsecured network 112 are unable to decipher the encrypted data. On the secured computer system 113, a server application 114 relays the encrypted data to the decryptor 104 for decryption and verification thereof. Upon verification of the encrypted data, results of the verification are provided to the server application 114.

Fig. 2a shows a partial data flow diagram of the encryptor 102 and decryptor 104. Referring to Fig. 2a, data input 201, captured from a data input source 202, is stored in a working storage 203. The data input 201 is then encrypted together with an encryptor id 204, an encryptor time 205 and a clock skew 206. The encryptor id 204, which is preferably prefixed with the text "KRYPTKEY", is an identification code unique to the encryptor 102. The encryptor id 204, the encryptor time 205 and the clock skew 206 have a fixed data length while the data input 201 has a variable data length. However, when no data input 201 is provided, the encryptor data 207 is also generable from only the encryptor id 204, the encryptor time 205 and the clock skew 206. Inclusion of the encryptor id 204, the encryptor time 205 and the clock skew 206 enables validation of the encrypted data 207 by the decryptor 104.

A conventional symmetric or asymmetric cryptographic algorithm is adopted to generate the encrypted data 207 in the encryption module 208 with the use of an encryption key 209. Symmetric cryptographic algorithm is preferable over asymmetric cryptographic algorithm as symmetric cryptographic algorithm requires lower processing power. The encryption key 209 is obtained directly from the input source 202 or indirectly generated from a user input memorable passphrase comprising a series of characters. With the encryption key 209 provided at the input source 202 being variable, the encryptor 102 is not limited to usage by a single user or specific application. Various users are able to use the encryptor 102 for transmitting confidential data to different decryptors 104 through the use of the corresponding encryption key 209.

In a symmetric cryptographic algorithm implementation, a decryption key identical to the encryption key 209 is required for decrypting the encrypted data 207 in the decryption module 211. Alternatively, in an asymmetric cryptographic algorithm implementation, a pair of non-identical encryption and decryption keys 209/210 is used in the encryption and decryption modules 208/211 respectively. The decryption key 210 is retrievable from a decryption key repository 212.

A typical cryptographic algorithm only transforms plaintext data into a fixed encrypted data which can be easily captured and replayed for conducting fraudulent operations. However, the encrypted data 207 generated by the encryptor 102 is not fixed for each identical data input 201. This is achieved by encrypting a dynamic variable, such as a time dependent value or a randomly generated code, together with the data input 201 for generating a dynamic and non-predictable encrypted data 207.

The time dependent value is preferably a transmission timestamp which indicates the time of encryption. The transmission timestamp allows the validity of the encrypted data 207 to be limited to a time period definable by the decryptor 104. Therefore, the validity of the encrypted data 207 expires after a time period defined by the decryptor 104. This greatly limits the possibility of replay of the encrypted data 207 for intentionally conducting fraudulent operations.

The inclusion of a transmission timestamp requires time synchronisation of the encryptor 102 with decryptor 104 as an electronic clock disposed in either one thereof becomes off-sync over time. For multiple encryptors 102 and decryptors 104 to be time synchronised, a standard time system such as the Coordinated Universal Time (UTC) is preferably used as a standard reference time. However, the clock of the encryptor 102 must not be modifiable directly because this will allow perpetrators to conduct fraudulent operations. By modifying the clock into a "future" time and tricking a user into revealing confidential data such as credit card information, a perpetrator can capture and replay the encrypted data 207 at the pre-determined "future" time. Therefore, the clock skew 206 is introduced to prevent such fraudulent operations. While the encryptor time 205, preferably the present time obtainable from the clock of the encryptor 102, is not modifiable by the user, the encryptor 102 is synchronised by modifying the encryptor clock skew 206. When the encryptor 102 synchronises with the standard reference time, the clock skew 206 is calculated as follows:

Clock Skew = Standard Time Reference – Encryptor Time

Hence, the transmission timestamp is obtainable from both the encryptor time 205 and the clock skew 206 by the following:

Transmission Timestamp = Encryptor Time + Clock Skew

As an encrypted data 207 generated in the "future" time has a relatively larger clock skew value, the decryptor 104 is able to determine its validity simply by checking this clock skew value. If the clock skew value is greater than a predefined maximum clock skew limit, it is considered invalid. For example, when the maximum clock skew limit allowable is +30 seconds, the validity of the encrypted data 207 is determined in the following validity table:

Standard Reference Time	Encryptor Time	Clock Skew	Validity
00 hr 30 min 00 sec	00 hr 30 min 45 sec	-45 sec	Valid
00 hr 30 min 00 sec	00 hr 29 min 45 sec	+15 sec	Valid
00 hr 30 min 00 sec	00 hr 00 min 00 sec	+30 min	Invalid

When the decryptor 104 receives the encrypted data 207 as shown in Fig. 2a, the encrypted data 207 is decrypted into a decrypted data 213, a decrypted encryptor id 214, a decrypted encryptor time 215 and a decrypted clock skew 216 for verification by a verification module 217. Once verified, the decrypted encryptor id 214, the decrypted encryptor time 215 and the decrypted clock skew 216 are recorded into a history repository 218 and a verification result 219 is generated therefrom.

The history repository 218 further allows the decryptor 104 to check for duplication of the encrypted data 207 which alerts of possible fraudulent operation occurring via replay of the encrypted data 207. The decrypted encryptor id 214, which is a unique identification code of the encryptor 102, is used together with the decrypted encryptor time 215 and the decrypted clock skew 216 for comparison with the records in the history repository 218 for duplication checking. The duplication occurs and is detected when a record with an

identical decrypted encryptor id 214, the decrypted encryptor time 215 and the decrypted clock skew is found.

The encryptor id 204, the encryptor time 205 and the clock skew 206 further preclude a need for the data input 201 in a user authentication implementation. The encryption key 209 is one of represented directly and generated indirectly by the user password. Using the encryption key 209, the encryptor 102 generates the time limited and non-predictable dynamic encrypted data 207 which can be used for authentication. In the user authentication implementation, a user is authenticated when the encrypted data 207 is decipherable using the user's decryption key 210 and validated by the verification module 217.

Figs. 3a-4b are flow diagrams of the encryptor 102 and the decryptor 104 for illustrating the encryption and verification processes 300/400 of the encryptor 102 and decryptor 104. Hereinafter, details of the encryption and verification processes 300/400 for respectively the encryptor 102 and the decryptor 104 are described.

In the encryption process 300 as shown in Fig. 3a, the encryptor 102 is first activated through a control means (not shown in Fig. 3a) before the user enters the data input 201 through the input device 106. The control means allow the user the flexibility to switch between normal usage of the input device 106 and use of the input device 106 in an encryption mode for confidential data input therevia.

Upon activation of the encryptor (Step 302), the user inputs the encryption key 209 or a memorable passphase to generate the encryption key 209 corresponding to the decryptor 104 (Step 304). The encryption key 209 is subsequently saved in the working storage 203 within the encryptor 102 (Step 306). Thereafter, the user enters the data input 201 into the encryptor 102 (Step 308) whereby the data input 201 is then saved within the working storage 203 in preparation for encryption (Step 306).

To transmit the encrypted data 207, the user initialises the encryption module 208 using the control means (Step 310). Upon initiation of the encryption module, the encryptor id 204, the encryptor time 205 and the clock skew 206 are encrypted with the data input 201 using the encryption key 209 stored in the working storage 203 and a cryptographic algorithm (Step 314). The cryptographic algorithm, such as the Advance Encryption Standard (AES), preferably uses the encryption key 209 to generate the encrypted data 207.

Subsequently, the encrypted data 207 also goes through a Base64 encoding process (Step 316) before being transmitted from the encryptor 102. The Base64 encoding process transforms the binary encrypted data 207 into printable ASCII characters. This ensures that the encrypted data 207 can be displayed using any conventional display media.

Once the encrypted data 207 is transmitted from the encryptor 102 (Step 318), the encryptor 102 erases the data input 201 in the working storage 203 and prepares the encryptor 102 for encryption again using the same encryption key 209. In addition, the encryptor 102 checks for deactivation thereof (Step 320). The encryptor 102 can be deactivated using the control means which erases the encryption key 209 and the data input 201 from the working storage 203 (Step 322).

To facilitate the usage of the encryptor 102, the encryption keys 209 can be stored in a repository module 324 and subsequently retrieved for usage as shown in Fig. 3b. The repository module 324 allows a plurality of encryption keys 209 and their associated descriptors to be stored therewithin by the user. Besides containing the encryption keys 209, the repository storage 324 also stores frequently used data input 201, such as credit card numbers, to be encrypted with the associated encryption key 209. This removes the need for the user to memorise the encryption key 209 for each different server application 114.

In order to prevent unauthorised access to the repository storage 324, the repository storage 324 is password protected. Upon activation of the encryptor 102, the user enters a

password to access the encryption keys stored within the repository module 324 (Step 326). The password is verified against a registered password stored in the repository module 324 (Step 328). Subsequently, the required encryption key 209 is selectable by selecting the associated descriptor (Step 304). The encryption key 209 is subsequently automatically fed (Step 304) and stored in the working storage 203 (Step 306) for use in the encryption module 208. Subsequently, when the data input 201 corresponds with the encryption key 209, the data input 201 is automatically provided to (Step 308) and stored in the working storage 203. The aforementioned encryption process 300 continues thereafter from Step 308.

The repository module 324 is preferably located within the encryptor 102 or alternatively within an external device removably attachable to the encryptor 102. When located in an external device, the repository module 324 has the added advantage of portability. The portable external device enables the encryption keys and confidential data to be easily carried around by the user.

With reference to Fig. 4a, a flow diagram of the decryption process 400 as performed by the decryptor 104 is shown. Upon receiving the encrypted data 208 from the encryptor 102, the verification process 400 is initiated (Step 402) by the server application 114.

During the verification process 400, the encrypted data 207 is decoded back into the original binary coding from the Base64 encoding (Step 404). Within the decryptor 104, a decryption key repository 212 stores the decryption key 210, together with a unique identification code, for decrypting the encrypted data 207. The identification code for retrieving the decryption key 210 from the decryption key repository 212 is provided by the server application 114. In a simple user authentication implementation, a username supplied by the user and relayed to the decryptor 104 by the server application 114 is preferably used as the identification code. The decryption key 209 is used with the cryptographic algorithm to decrypt the encrypted data 207 into the decrypted data 213, the

decrypted encryptor id 214, the decrypted time 215 and the decrypted clock skew 216 (Step 408).

The verification module 217 subsequently performs a series of validation steps to determine the validity of the encrypted data 207. Firstly, the decrypted encryptor id 214 is checked for the presence of the text "KRYPTKEY" prefixed to the decrypted encryptor id 214 (Step 410). The validity of the decrypted encryptor id 214 determines whether the encrypted data 207 is successfully decrypted with the decryption key 210. If the decrypted encryptor id 214 fails the validity check, the decrypted data 213 is invalidated (Step 412). When the decrypted data 213 is invalidated, a history record comprising the decrypted encryptor id 214, the decrypted encryptor time 215, the decrypted clock skew 216 and an invalid status is recorded in the history repository 218 (Step 414).

However, if the decryptor id 214 is found to be valid, the decryption process 400 continues with the validation checks to determine the validity of the transmission timestamp (Step 416). The transmission timestamp is computed as the sum of the decrypted encryptor time 215 and decrypted clock skew 216:

Transmission Timestamp = Decrypted Encryptor Time + Decrypted Clock Skew

This check initially determines the time difference between a standard reference time 418, obtainable from a standard reference time source and the transmission timestamp:

Time Difference = Standard Reference Time – Transmission Timestamp

The time difference is then compared with the predefined upper and lower time limits to determine the validity of the decrypted data 213:

Valid: Lower Time Limit <= Time Difference <= Upper Time Limit
Invalid: Lower Time Limit > Time Difference > Upper Time Limit

These upper and lower time limits determine the lifespan of the encrypted data 207. They are pre-definable by the decryptor 104 and variable.

The decrypted data 213 is invalid when the time difference is numerically lower than the lower time limit or numerically higher than the upper time limit. If the transmission timestamp fails the validity check, the encrypted data 207 is invalidated (Step 412) and the history record thereof is recorded with an invalid status in the history repository 218 (Step 414).

However, if the transmission timestamp is valid, the decrypted encryptor id 214, the decrypted encryptor time 215 and the decrypted clock skew 216 is checked with the records in the history repository 218 to determine if the encrypted data 207 has been duplicated and therefore reoccurs (Step 420). This catches replay of the encrypted data 207 previously verified by the decryptor 104. If the decrypted encryptor id 214, the decrypted encryptor time 215 and the decrypted clock skew 216 fail the duplication check, the encrypted data 207 is invalidated (Step 412) and the history record thereof is recorded with an invalid status into the history repository 218 (Step 414).

Subsequently, if the decrypted encryptor id 214, decrypted encryptor time 215 and the decrypted clock skew 216 passes the duplication check (Step 420), the decrypted clock skew 216 is then used for comparison with a predefined maximum clock skew limit to determine whether the encrypted data 207 is generated from a "future" time (Step 422).

However, once the encryptor time 205 of the encryptor 102 is substantially different from the standard reference time 418 to the extent where the encryptor clock skew 206 is greater than the maximum clock skew limit, the encrypted data 207 subsequently regenerated from the encryptor 102 will always fail the clock skew check. To remedy this problem, the difference between the decrypted clock skew 216 of the current encrypted data 207 and the decrypted clock skew 216 of a previously verified encrypted data 207 from the same

encryptor 102, namely a clock skew drift, is computed and used for comparison with the maximum clock skew limit. The clock skew drift is computed as follows:

Clock Skew Drift = Decrypted Clock Skew - Previous Clock Skew

The previous clock skew is retrieved from the latest history record, containing a matching decrypted encryptor id 214, an earlier decrypted encryptor time and a valid status, stored in the history repository 218.

If a matching history record is found, the decrypted clock skew 216 is validated as follows:

Valid: Clock Skew Drift ≤ Maximum Clock Skew Limit
Invalid: Clock Skew Drift > Maximum Clock Skew Limit

For example, if the maximum clock skew limit is +30 seconds, the validity of the encrypted data 207 is determined as follows:

Standard Reference Time	Encryptor Time	Decrypted Clock Skew	Previous Clock Skew	Validity
00 hr 30 min 00 sec	00 hr 29 min 10 sec	+50 sec	+20 sec	Valid
00 hr 30 min 00 sec	00 hr 29 min 55sec	+5 sec	-20 sec	Valid
00 hr 30 min 00 sec	00 hr 30 min 10 sec	-10 sec	-50 sec	Invalid
00 hr 30 min 00 sec	00 hr 00 min 00 sec	+30 min	+40 sec	Invalid

If no history record is found with the matching decrypted encryptor id 214 and a valid status, the decrypted clock skew 216 is validated as follows:

Valid: Decrypted Clock Skew ≤ Maximum Clock Skew Limit
Invalid: Decrypted Clock Skew > Maximum Clock Skew Limit

If the decrypted clock skew 216 is valid, the encrypted data 207 is validated (Step 424) and the history record with a valid status is recorded into the history repository 218 (Step 414). However, if the decrypted clock skew 216 is invalid, the encrypted data 207 is invalidated (Step 412) and the history record thereof will be recorded with an invalid status into the history repository 218 (Step 414).

Thereafter, the verification result 219 is generated and provided to the server application 114. The verification result preferably comprises one of the decrypted data 213, should the encrypted data 207 be validated by the verification module 217, and an invalid status, should the encrypted data 207 be invalidated.

The encryptor time 205 of the encryptor 102 may substantially differ from the standard reference time 418 such that the encryptor clock skew 206 is greater than the maximum clock skew limit. In such an event and where there is no existence of a history record for computing the clock skew drift, the encrypted data 207 generated from the encryptor 102 will always fail the clock skew check (Step 422). Hence, an alternative means for verifying the encrypted data 207 is required as shown in Figs. 4b in a feedback challenge 430 without depending on the validity of the encryptor clock skew 206 (Step 422). The feedback challenge 430 is initiated (Step 422) by the server application 114 when verification without the clock skew check is required (Step 432). Prior to the transmission of any confidential data, the server application 114 first establishes a session with the client application 109 (Step 434). Data related to the session is stored in the session repository 436 by the application server 114. Thereafter, the server application 114 transmits a randomly generated feedback code (not shown), for example a random code, to the client application 109 (Step 438). At the same time, the server application 114 opens a time limited window for the feedback code to be sent back. The feedback code together with its expiry time, definable by the server application, is stored in the session repository 436.

Upon receiving the feedback code, the client application 109 prompts the user to enter the feedback code, together with other confidential data if required, as data input 201 into the

encryptor 102. The feedback code encrypted in the encrypted data 207 is subsequently transmitted back to the server application 114. After receiving the encrypted data 207 (Step 440), the server application 114 checks if expiry time of the feedback code has been exceeded (Step 442). If the expiry time is exceeded, the session is invalidated (Step 444) and the server application 114 will determine whether the verification process should end (Step 446) or whether a new feedback code (Step 438) should be sent. If the expiry time is not exceeded, the encrypted data 207 and the feedback code to be inter-matched are transmitted to the decryptor 104 for feedback verification (Step 448).

With reference to Fig. 4c, the decryptor 104 initiates a feedback code verification process 448 upon receipt of the encrypted data 207 and the feedback code (Step 450). The decryptor 104 decrypts the encrypted data 207 to obtain the decrypted data 213 containing the feedback code via the decryption module 211. Thereafter, the encryptor id's validity check (Step 410) and the transmission timestamp's validity check (Step 416) are performed.

In replacing the clock skew's validity check (Step 422), the feedback code embedded in the decrypted data 213 is compared with the feedback code stored within the session repository 436, for example a reference random code (Step 452). If the feedback codes are not identical, the decrypted data 213 is invalidated (Step 412) and the history record is recorded into the history repository 218 with an invalid status (Step 414). If the feedback codes are identical, a duplication check (Step 420) is performed.

If duplication of the history record is found during the duplication check, the encrypted data input 207 is invalidated (412) and the history record is recorded into the history repository 218 with an invalid status (Step 414). Otherwise, the decrypted data 213 is validated and a history record with a valid status is recorded into the history repository 218 (Step 414). With this valid history record, subsequent confidential data transmission from the encryptor 102 is verifiable by checkingthe decrypted clock skew 216 (Step 422).

Thereafter, the verification result 219 is generated and provided to the server application 114 (Step 426).

With reference to Fig. 4b, after the feedback verification (Step 448), if the feedback code is valid (Step 454), the server application 114 validates the current session (Step 456) and ends the feedback challenge process 430 (Step 458). Otherwise, the session is invalidated (Step 444) and checks are performed to determine if a reverification is required (Step 446). If not, the feedback challenge process 430 is terminated.

By getting the user to encrypt the feedback code with his encryption key 209 using the encryptor 102 and have the feedback code sent back to the server application 114 for verification, the server application 114 can verify whether the user is genuine and whether the encrypted feedback code is being transmitted in real-time. Without the user's encryption key 209, perpetrators have no means of faking the encrypted feedback code. Since the feedback code is generated randomly and has an expiry time, the perpetrators are not able to capture the encrypted feedback code and use it after the expiry time.

The encryptor 102 is an important element of the secure system 100 which provides the tools to secure confidential data input. Hereinafter the encryptor 102 is described as an embodiment of the invention according to Fig. 5. Fig. 5 shows a system-based partial representation of the components within the encryptor 102 for facilitating encryption. The encryptor 102 comprises a storage mean 502, a clock 504, a processor 506, a control means 508 and a display 510.

The storage means 502 include one or more non-volatile memory, such as a ROM (Read-Only Memory) and an EEPROM (Electrically Erasable Read-Only Memory), and volatile memory, such as a RAM (Random Access Memory). The non-volatile memory stores the program instructions including the cryptographic algorithm, the encryptor id 204 and the encryptor clock skew 206. In addition, the non-volatile memory provides a repository module 324 for storing multiple encryption keys 209 and the confidential data input 201.

The volatile memory provides the working storage 203 that temporarily stores the data input 201 and encryption key 209, as well as data required for processing by the processor 506.

The clock 504 is installed within the encryptor 102 for generating the encryptor time 205, which is not modifiable by the user and independent of the timing on the unsecured computer system. This prevents manipulation of the encryptor time 205 for conducting fraudulent replay for the encrypted data 207. The clock 504 is synchronised with the standard reference time during fabrication thereof.

A user preferably controls and interacts with the encryptor 102 via the control mean 508 and the display 510. The control mean 508, which can be in the form of electronic buttons control, is used for activating the encryptor 102 and initiating the encryption module 208 while the display 510 provides visual feedback to the user. In addition, the encryptor 102 is preferably controlled and interacted with by the user using the input device 106, for example a computer keyboard, connected thereto. The display 510 allows data input 201 to be viewed as they are keyed into the input device 106 to thereby prevent erroneous data input. The display 510 further allows both the display of the transmission timestamp and indication of the encryptor's 102 synchronisation with the standard reference time by the user.

The processor 506 operates the encryptor 102 via program instructions stored in the non-volatile memory of the storage mean 502. The processor 506 performs the functions of both the encryptor 102, for example the retrieval of the data input 201 and encryption key 109, and the encryption module 208, as well as performs synchronisation with the standard reference time.

The encryptor 102 is either operatively connected between the input device 106 and the computer system 110 or built into the input device 106. Input and output (I/O) interfaces 512/514 are provided with the encryptor 102 for interconnecting the input device 106 and

computer system 110 with the encryptor 102. Once activated, the encryptor 102 can immediately process any data input 201 from the input device 106. This prevents any malwares (not shown) residing on the unsecured computer system 110 from accessing the data input 201 since the data input 201 is encrypted before being sent to the unsecured computer system 110.

In addition, a portable external storage mean 516, attachable to the encryptor 102 via an external storage interface 518, can be used for providing the repository module 324 for storing multiple encryption keys and confidential data inputs. The portable external storage mean 516 is advantageous for users who require use of the same set of encryption keys on multiple encryptors 102.

In the foregoing manner, a system and a process according to embodiments of the invention for security of data within a system are described hereinafter for encrypting data so as to preserve data confidentiality, integrity and validity on the system are disclosed. Although only a number of embodiments of the invention are disclosed, it becomes apparent to one skilled in the art in view of this disclosure that numerous changes and/or modification can be made without departing from the scope and spirit of the invention.

Claims

1) An encryptor for encrypting data input comprising:

an encryption module for data communicatively interfacing an input device and a computing system, a user interacting with the input device for generating data input, the encryption module for obtaining a dynamic variable and for encrypting the dynamic variable with the data input into encrypted data using an encryption algorithm,

wherein the input device is in direct data communication with the encryption module, the encrypted data being transmitted to and being relayed by the computer system for subsequent reception and decryption by a decryptor into decrypted data and a decrypted dynamic variable, the decrypted dynamic variable being comparable with a reference dynamic variable obtainable by the decryptor for verifying validity of the decrypted data,

whereby the decrypted data is representative of the data input when the decrypted dynamic variable corresponds with the dynamic variable and the reference dynamic variable.

2) The encryptor as claimed in claim 1, further comprising:

an encryption key for encrypting the data input, the encryption key being at least one of introducible by the user using the input device and generable from the data input provided by the user using the input device.

3) The encryptor as claimed in any of claims 1 and 2,

wherein each of the dynamic variable, the reference dynamic variable and the decrypted dynamic variable being at least one of a random code and a time dependent value being generable from a clock,

wherein the reference dynamic variable corresponds with a reference time dependent value, the decrypted dynamic variable corresponds with a decrypted time

dependent value and the random code corresponds with a decrypted random code, and the clock for further generating an encryptor time.

4) The encryptor as claimed in claim 3, the decrypted data being one of valid and invalid, wherein the decrypted data is invalid when the decrypted time dependent value is one of:

more than a upper time limit; and

less than a lower time limit, the upper time limit and the lower time limit being predefined.

5) The encryptor as claimed in any of claims 3 and 4, further comprising:

a clock skew for time synchronizing the encryptor with the decryptor wherein the time dependent value is the sum of the clock skew and the encryptor time,

wherein the clock skew corresponds with a decrypted clock skew, with the decrypted clock skew being decryptable from the encrypted data.

6) The encryptor as claimed in claim 5, wherein the decrypted clock skew is comparable with a pre-defined maximum clock skew limit,

wherein the decrypted clock skew is decryptable from the encrypted data.

7) The encryptor as claimed in claim 6, the decrypted data being invalid when the decrypted clock skew is greater than a predefined maximum clock skew limit.

8) The encryptor as claimed in any of claims 5 and 7, further comprising:

an encryptor id for identifying the encryptor, the encryptor id being unique,

wherein the encrypted data is one of further encrypted from and concatenated with the encryptor id,

whereby a decrypted encryptor id decryptable from the encrypted data with the decrypted encryptor id corresponding with the encryptor id.

9) The encryptor as claimed in claim 8, wherein the decrypted encryptor id is checked by the decryptor for presence of a predefined text prefixed with the decrypted encryptor id wherein the presence of the predefined text indicates valid decryption of the encrypted data by the decryptor.

10) The encryptor as claimed in any of claims 8 and 9, further comprising:

a history record of the decrypted encryptor id, the decrypted clock skew and the decrypted time dependent value wherein the history record is recorded in a history repository after verification thereof.

11) The encryptor as claimed in claim 10, wherein the decryptor is further for comparing the decrypted clock skew with at least one of a pre-defined maximum clock skew limit and a previous decrypted clock skew contained in the history record for determining validity of the decrypted data, wherein the previous decrypted clock skew being subsequently updatable with the decrypted clock skew.

12) The encryptor as claimed in claim 11, the decrypted data being invalid when at least one of the decrypted clock skew and a value obtained when the previous clock skew is numerically subtracted from the decrypted clock skew is greater than a predefined maximum clock skew limit.

13) The encryptor as claimed in any of claims 10 and 12, wherein the decryptor invalidates the decrypted data when the decrypted encryptor id, the decrypted clock skew and the decryptor time dependent value are present in the history record prior to the encrypted data being decrypted.

14) The encryptor as claimed in claim 3, the random code being initially sent by the decryptor to the encryptor for encryption as the dynamic variable into the encrypted data, and the decrypted random code decrypted from the encrypted data being substantially checked with a reference random code for verification thereof, wherein the decrypted data is valid when the decrypted random code is identical to the reference random code stored in the session repository, the decrypted random being sent from the decryptor .

15) The encryptor as claimed in claim 1, further comprising:

a storage module for storing at least one of the data input, the encrypted data and the encryption algorithm.

16) The encryptor as claimed in any of claims 1 and 15, further comprising:

a control mean for initiating encryption by the encryption algorithm.

17) The encryptor as claimed in any of claims 1, 15 and 16, further comprising:

a display for displaying the operative status of the encryptor.

18) The encryptor as claimed in any of the claims, 1, 3, 15, 16 and 17, further comprising:

a processor for operatively interacting the encryptor.

19) The encryptor as claimed in claim 1, wherein the encryptor being one of integral with a keyboard and structurally computing system independent.

20) The encryptor as claimed in claim 1, further comprising:

a repository module for storing a plurality of encryption keys, the repository module being removably integratable with the encryption module for establishing data communication therewith and for providing at least one of the plurality of encryption keys thereto,

wherein the encryption module receiving the at least one of the plurality of encryption keys provided by the repository module, and the at least one of the plurality of encryptor keys corresponding with the data input.

21) The encryptor as claimed in claim 20, the repository module comprising:

a user interface adapted for interaction with a person for selecting at least one of the plurality of keys for transmission to the encryptor.

22) The encryptor as claimed in claim 20, the repository module being structurally computing system independent.

23) The encryptor as claimed in claim 1, wherein the encrypted data is decryptable by the decryptor into at least one of the decrypted data, a decrypted clock skew and the decrypted dynamic variable.

24) A decryptor for decrypting and verifying the validity of encrypted data comprising:

a decryption module for decrypting encrypted data into decrypted data and a decrypted dynamic variable using a decrypting algorithm,

wherein the encrypted data is subsequently received and decrypted into the decrypted data and the decrypted dynamic variable, the decrypted dynamic variable being compared with a reference dynamic variable obtained by the decryptor for verifying validity of the decrypted data, and the encrypted data being encrypted from a data input,

whereby the decrypted data is representative of the data input introducible by a user when the decrypted dynamic variable corresponds with a dynamic variable and the reference dynamic variable.

25) The decryptor as claimed in claim 24, wherein each of the dynamic variable, the reference dynamic variable and the decrypted dynamic variable is at least one of a random code and a time dependent value generated from a clock,

wherein the reference dynamic variable corresponds with a reference time dependent value, the decrypted dynamic variable corresponds with a decrypted time dependent value and the random code corresponds with a decrypted random code, and the clock for further generating an encryptor time.

26) The decryptor as claimed in claim 25, the decrypted data being one of valid and invalid, wherein the decrypted data is invalid if the decrypted time dependent value is one of:

more than a upper time limit; and

less than a lower time limit, the upper time limit and the lower time limit is predefined.

27) The decryptor as claimed in any of claims 25 and 26, further comprising:
a clock skew for time synchronizing the encryptor with the decryptor,

wherein the time dependent value is the sum of the clock skew and the encryptor time;

whereby a decrypted clock skew corresponding with the clock skew is decryptable from the encrypted data.

28) The decryptor as claimed in claim 27, wherein a decrypted clock skew is comparable with at least one of a predefined maximum clock skew limit and a previous clock skew for determining validity of the decrypted data, and wherein the decrypted clock

skew is decryptable from the encrypted data and the previous clock skew is subsequently updatable with the decrypted clock skew.

29) The decryptor as claimed in claim 28, the decrypted data is invalid when at least one of the decrypted clock skew and a value obtained when the previous clock skew is subtracted from the decrypted clock skew is greater than a predefined maximum clock skew limit.

30) The decryptor as claimed in any of claims 27 and 29, further comprising:

a encryptor id for identifying the encryptor, the encryptor id being unique,

wherein the encrypted data is one of further encrypted from and concatenated with the encryptor id;

whereby a decrypted encryptor id decryptable from the encrypted data being identical to the encryptor id.

31) The decryptor as claimed in claim 30, wherein the decrypted encryptor id is checked for presence of a predefined text prefixed with the decrypted encryptor id wherein the presence of the predefined text indicates valid decryption of the encrypted data by the decryptor.

32) The decryptor as claimed in any of claims 30 and 31, further comprising:

a history record of the decrypted encryptor id, the decrypted clock skew and the decrypted time dependent value wherein the history record is recorded in a history repository after verification thereof.

33) The decryptor as claimed in claim 32, wherein the decryptor invalidates the decrypted data when the decrypted encryptor id, the decrypted clock skew and the

decryptor time dependent value are present in the history record prior to the encrypted data being decrypted.

34) The decryptor as claimed in claim 25, the random code being initially sent by the decryptor to the encryptor for encryption as the dynamic variable into the encrypted data, and the decrypted random code decrypted from the encrypted data being substantially checked with a reference random number for verification, wherein the decrypted data is valid when the decrypted random code is identical to the reference random code stored in the session repository.

35) The decryptor as claimed in claim 25, the clock is pre-synchronised with the reference time dependent value.

36) The decryptor as claimed in claim 30, further comprising a decryption key repository for storing at least one decryption key and a unique identification code corresponding with each of the at least one decryption key.

37) A process comprising the steps of:

receiving encrypted data being encrypted from a data input introducible by a user and a dynamic variable, the dynamic variable being obtainable from an encryption module;

decrypting the encrypted data with a decryptor into decrypted data and a decrypted dynamic variable; and

verifying validity of the decrypted data by comparing the decrypted dynamic variable with a reference dynamic variable obtained by the decryptor,

wherein the decrypted data is representative of the data input when the decrypted dynamic variable corresponds with the dynamic variable and the reference dynamic variable.

38) The process as claimed in claim 37, further comprising the step of:

using at least one of a time dependent value and a random code as the dynamic variable, and each of the reference dynamic variable and the decrypted dynamic variable being at least one of a random code and a time dependent value being generable by a clock,

wherein the reference dynamic variable corresponds with a reference time dependent value, the decrypted dynamic variable corresponds with a decrypted time dependent value and the random code corresponds with a decrypted random code, and the clock for further generating an encryptor time.

39) The process as claimed in claim 38, further comprising the step of:

encrypting the data input and the time dependent value into the encrypted data with an encryptor.

40) The process as claimed in claim 38, further comprising the step of:
determining validity of the encrypted data by comparing the decrypted time dependent value against predetermined upper time limit and lower time limit values.

41) The process as claimed in claim 40, the step of determining the validity of the encrypted data further comprising the step of:

comparing whether the decrypted time dependent value is at least greater or equal to the lower time limit and is at least smaller or equal to the upper time limit,

whereby the encrypted data is valid when the decrypted time dependent value is within the range of the lower time limit and the upper time limit.

42) The process as claimed in any of claims 38 and 40, further comprising the step of:
providing a clock skew for time synchronizing the encryptor with the decryptor wherein the time dependent value is the sum of the encryptor time and the clock skew,

whereby a decrypted clock skew, that is identical to the clock skew, is decryptable from the encrypted data.

43) The process as claimed in claimed 42, further comprising the step of:

checking validity of the decrypted clock skew by comparing the decrypted clock skew with at least one of a predefined maximum clock skew limit and a previous clock skew wherein the decrypted clock skew is decryptable from the encrypted data and the previous clock skew is subsequently updatable with the decrypted clock skew.

44) The process as claimed in any of the claims 42 and 43, further comprising the step of:

using a encryptor id for encryption and identification of the encryptor;

wherein the decrypted encryptor id is checked for presence of a predefined text prefixed with the decrypted encryptor id wherein the presence of the predefined text indicates valid decryption of the encrypted data by the decryptor.

45) The process as claimed in any of the claims 43 and 44, further comprising the step of:

checking presence of the decrypted encryptor id, the corresponding decrypted clock skew and the corresponding decrypted time dependent value in a history repository for determining previous use thereof; and

recording the decrypted encryptor id, the corresponding decrypted clock skew and the corresponding decrypted time dependent value into the history repository.

46) The process as claimed in claim 45, the step of checking presence of the decrypted encryptor id, the corresponding decrypted clock skew and the corresponding decrypted time dependent value in a history repository, further comprising the step of:

invalidating the decrypted data when the decrypted encryptor id, the decrypted clock skew and the decryptor time dependent value are present in the history record prior to the encrypted data being decrypted.

47) The process as claimed in claim 38, the step of using at least one of a time dependent value and a random code as the dynamic variable, further comprising the step of:

checking the decrypted random code with a reference random code for verification thereof, wherein the decrypted data is valid if the decrypted random code is identical to the reference random code stored in the session repository.

48) An encryptor for encrypting data input comprising:

an encryption module for encrypting data input, a clock skew and a timestamp into encrypted data using an encryption algorithm, the data input being introducible by a user; and

a clock for generating the clock skew and the timestamp,

wherein the encrypted data is subsequently receivable and decryptable by a decryptor into decrypted data, a decrypted clock skew and a decrypted timestamp, at least one of the decrypted clock skew and the decrypted timestamp being comparable with a reference timestamp obtainable by the decryptor for verifying validity of the decrypted data, and

whereby the decrypted data is representative of the data input when the decrypted clock skew and the decrypted timestamp corresponds with the clock skew and timestamp and the reference clock skew and the reference timestamp.

49) An encryptor for encrypting data input comprising:

an encryption module for encrypting data input introducible by an user and a time dependent value into encrypted data using an encryption algorithm and a user encryption key, the data input and the user encryption key being provided by a user; and

a clock for generating the time dependent value,

wherein the encrypted data is subsequently receivable and decryptable using a decryption algorithm and a decryption key by a decryptor into decrypted data and a decrypted time value, the decrypted time value being comparable with a reference time value obtainable by the decryptor for verifying validity of the decrypted data, and the decryption key being associated with the user decryption key, and

whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the time dependent value and the reference time value.

50) An encryptor for encrypting data input comprising:

an encryption module for encrypting data input introducible by a user into encrypted data using an encryption algorithm and a user input encryption key; and

a user input device, the user interacting with the user input device to thereby enter the data input into the encryptor,

wherein the encrypted data is subsequently receivable and decryptable using a decryption algorithm and a decryption key by a decryptor into decrypted data and whereby the decrypted data is representative of the data input with the decryption key being associated with the user encryption key, and

whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the time dependent value and the reference time value.

51) An encryptor for encrypting data input comprising:

an encryption module for encrypting data input and a clock skew into encrypted data using an encryption algorithm, the data input being introducible by a user; and

a clock for generating the clock skew,

wherein the encrypted data is subsequently receivable and decryptable by a decryptor into decrypted data and a decrypted time value, the decrypted time value being comparable with a reference time value obtainable by the decryptor for verifying validity of the decrypted data, and

whereby the decrypted data is representative of the data input when the decrypted time value corresponds with the clock skew and the reference time value.

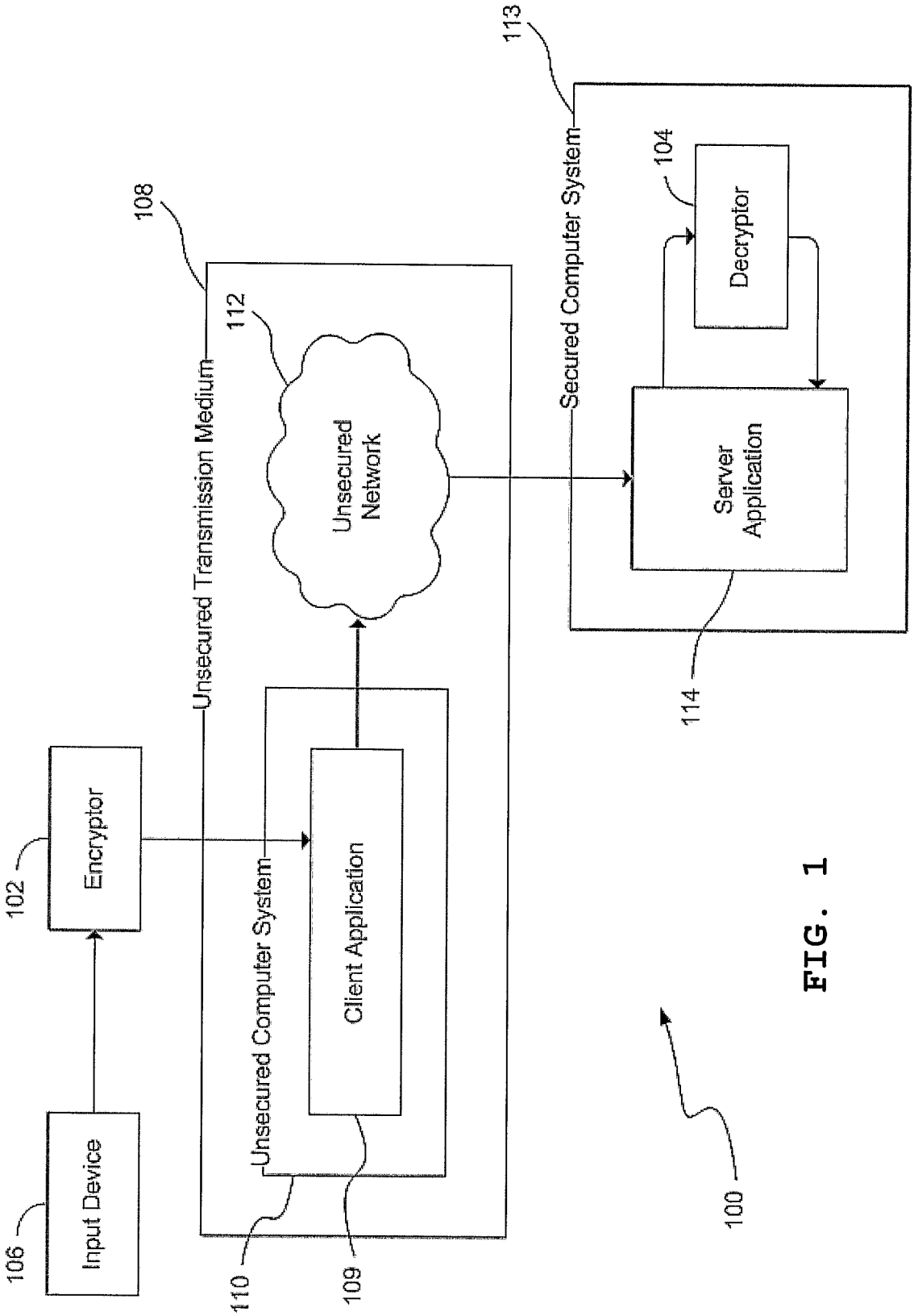

Input Device
106
Encryptor
102
Unsecured Transmission Medium
108
Unsecured Computer System
110
Client Application
109
Unsecured Network
112
Secured Computer System
113
Server Application
114
Decryptor
104
100

FIG. 1

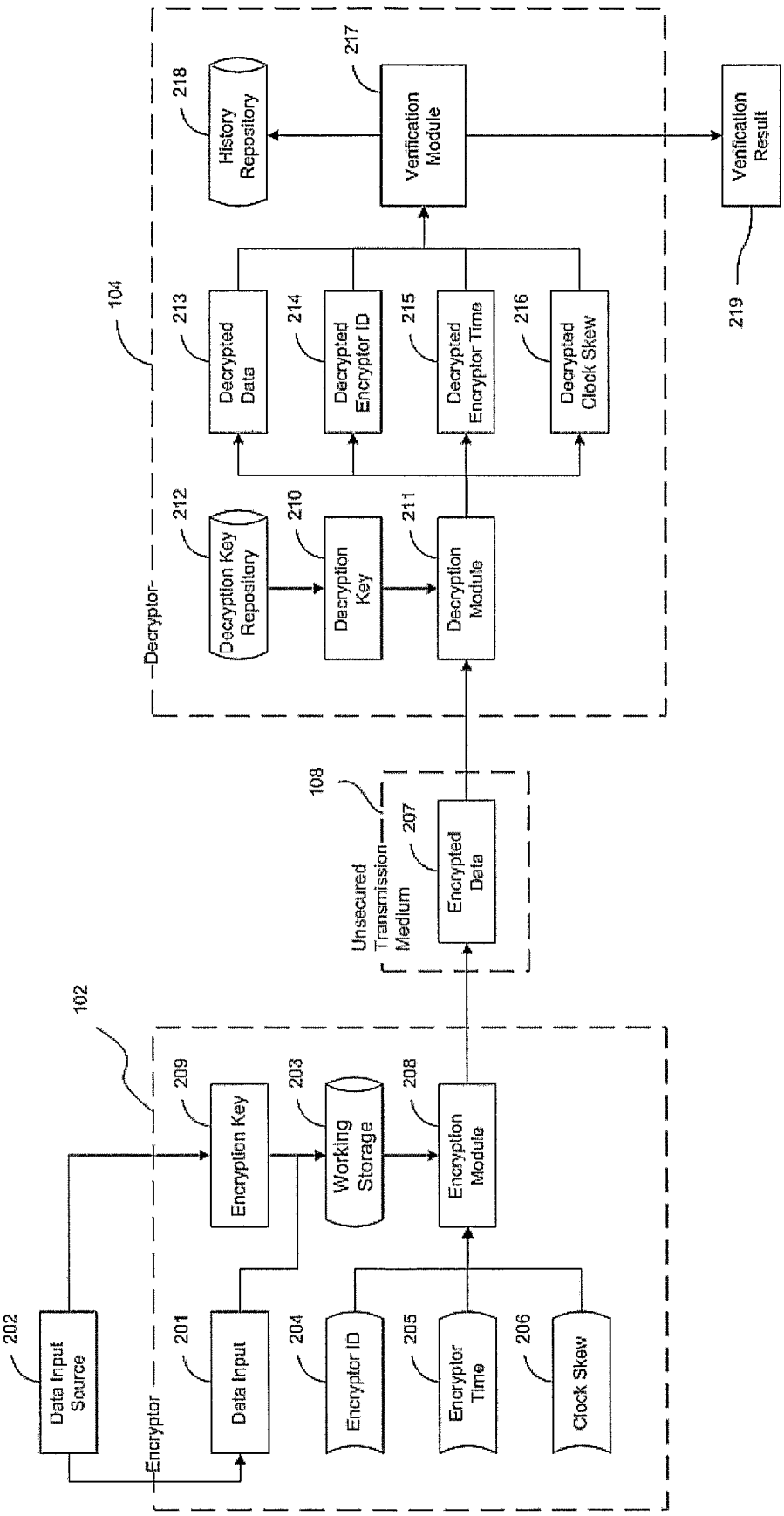

Encryptor
Data Input Source
202
Data Input
201
Encryption Key
209
Working Storage
203
Encryption Module
208
Encryptor ID
204
Encryptor Time
205
Clock Skew
206
102
Unsecured Transmission Medium
109
Encrypted Data
207
Decryptor
Decryption Key Repository
212
Decryption Key
210
Decryption Module
211
Decrypted Data
213
Decrypted Encryptor ID
214
Decrypted Encryptor Time
215
Decrypted Clock Skew
216
104
History Repository
218
Verification Module
217
Verification Result
219

FIG. 2

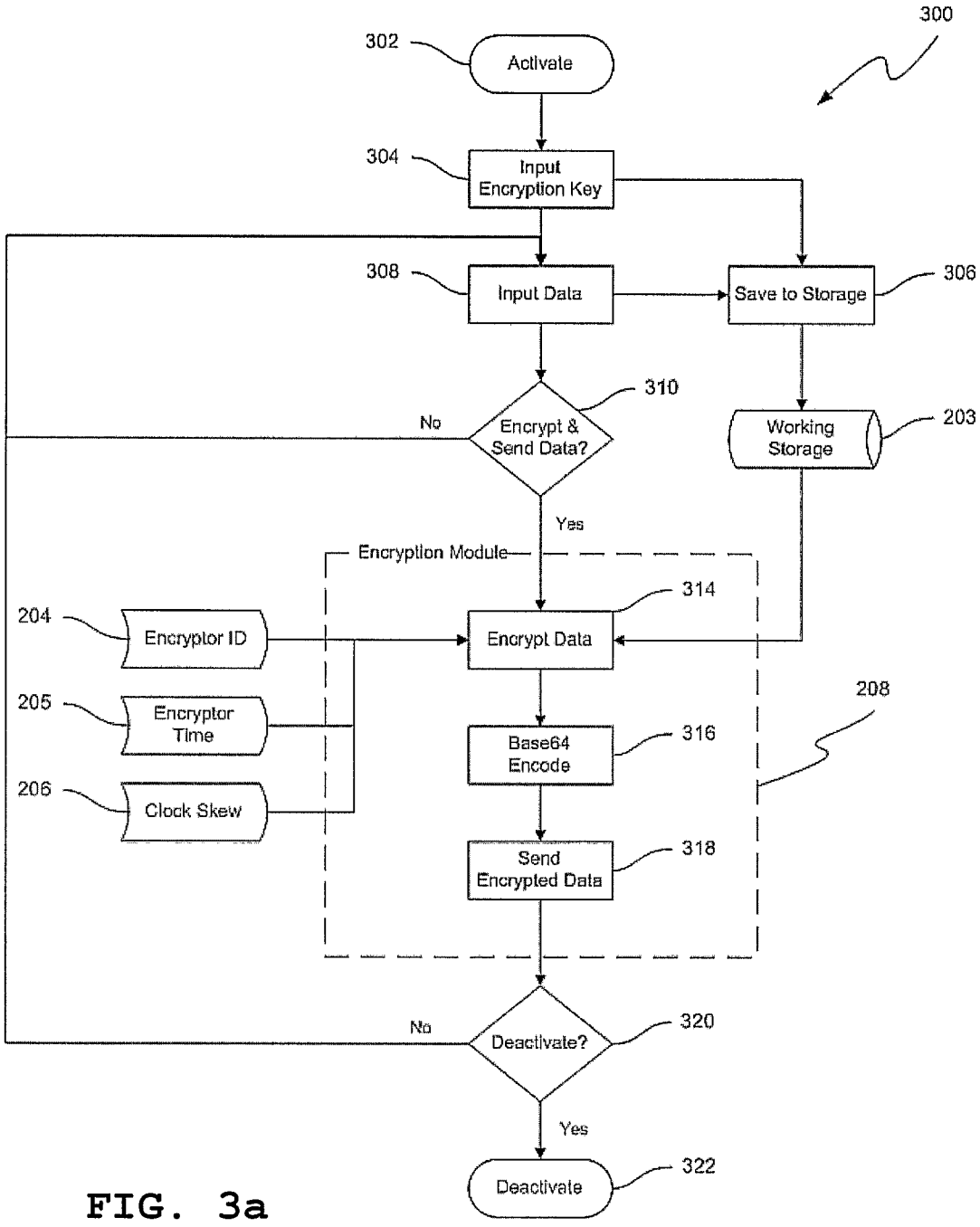

300
302
Activate
304
Input Encryption Key
308
Input Data
306
Save to Storage
310
No
Encrypt & Send Data?
203
Working Storage
Yes
Encryption Module
314
204
Encryptor ID
Encrypt Data
205
Encryptor Time
208
316
Base64 Encode
206
Clock Skew
318
Send Encrypted Data
No
Deactivate?
320
Yes
322
Deactivate

FIG. 3a

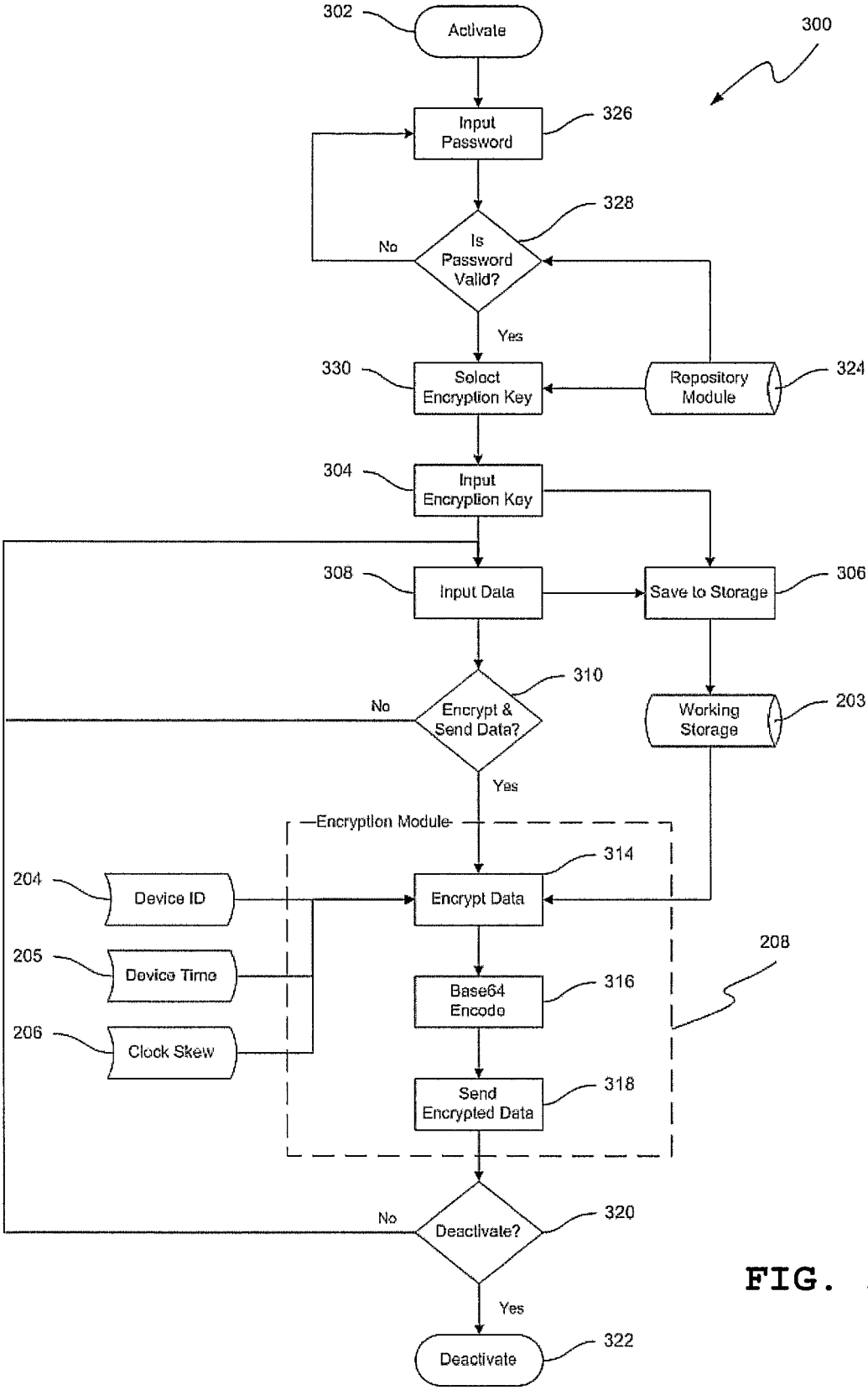

302
Activate
300
Input Password
326
328
No
Is Password Valid?
Yes
330
Select Encryption Key
Repository Module
324
304
Input Encryption Key
308
Input Data
Save to Storage
306
310
No
Encrypt & Send Data?
Working Storage
203
Yes
Encryption Module
314
204
Device ID
Encrypt Data
208
205
Device Time
Base64 Encode
316
206
Clock Skew
Send Encrypted Data
318
No
Deactivate?
320
Yes
Deactivate
322

FIG. 3b

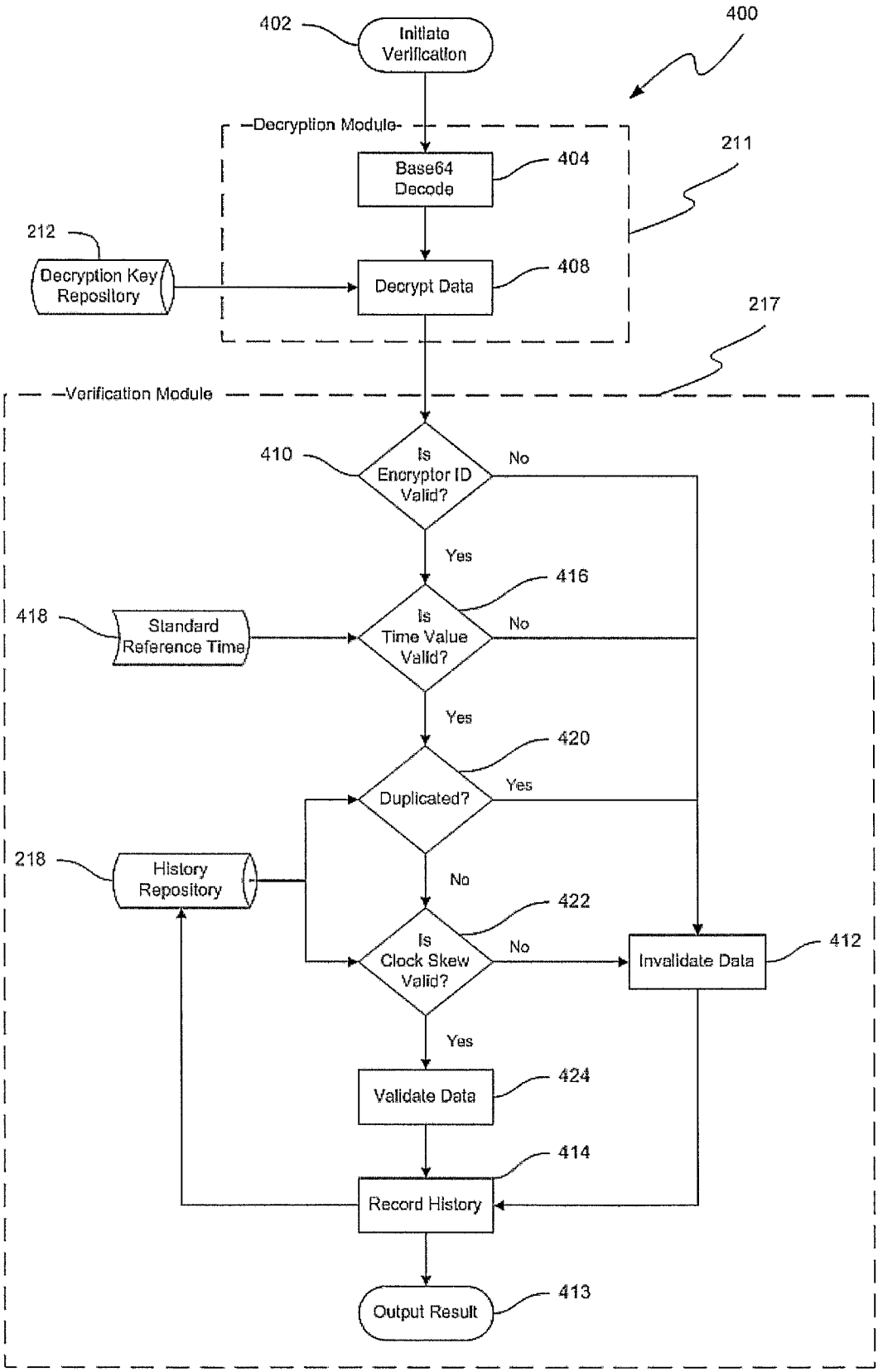

402
Initiate Verification
400
Decryption Module
Base64 Decode
404
211
212
Decryption Key Repository
Decrypt Data
408
217
Verification Module
410
Is Encryptor ID Valid?
No
Yes
416
418
Standard Reference Time
Is Time Value Valid?
No
Yes
420
Duplicated?
Yes
218
History Repository
No
422
Is Clock Skew Valid?
No
412
Invalidate Data
Yes
Validate Data
424
414
Record History
Output Result
413

FIG. 4a

Feedback Challenge

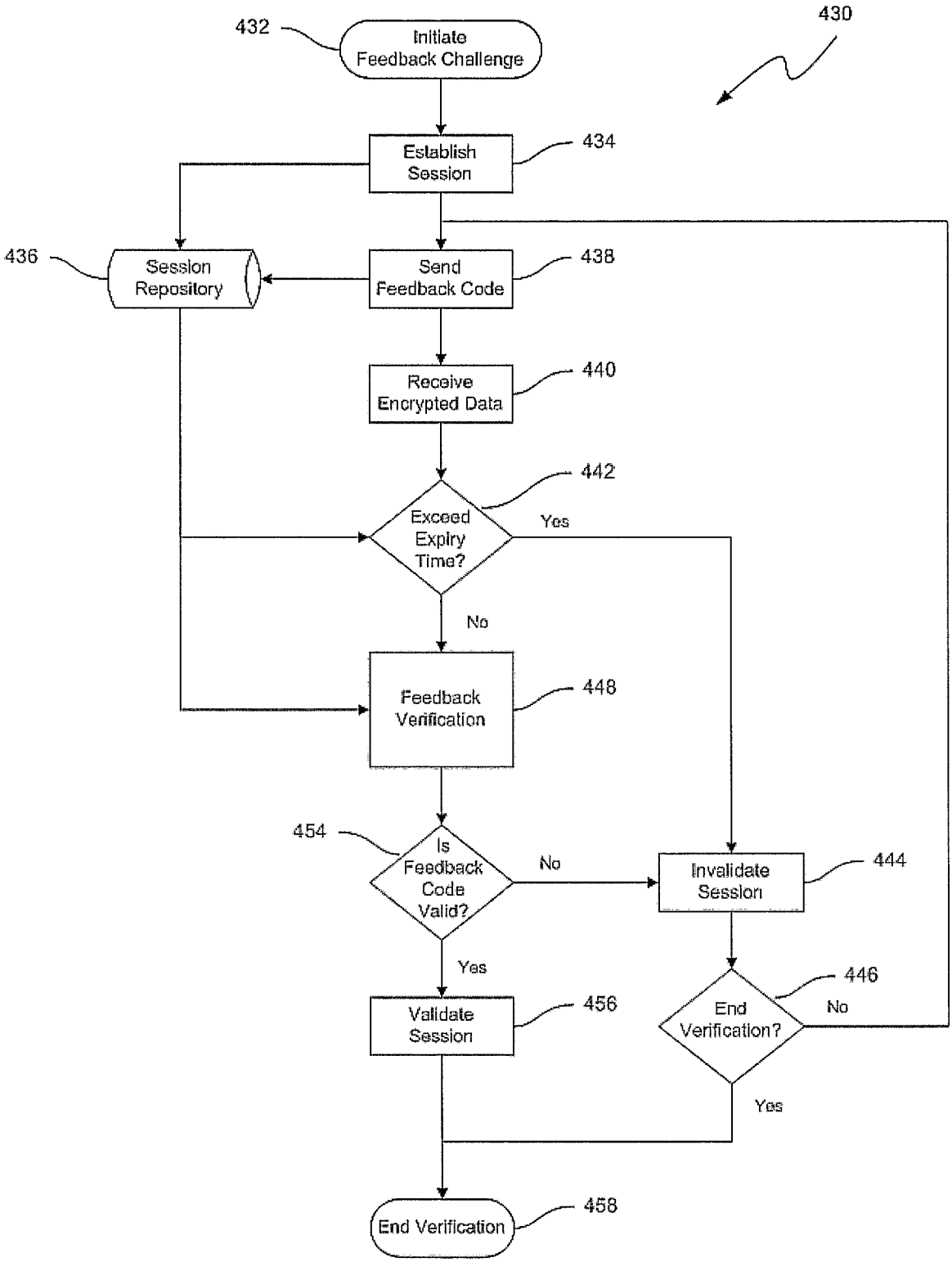


FIG. 4b

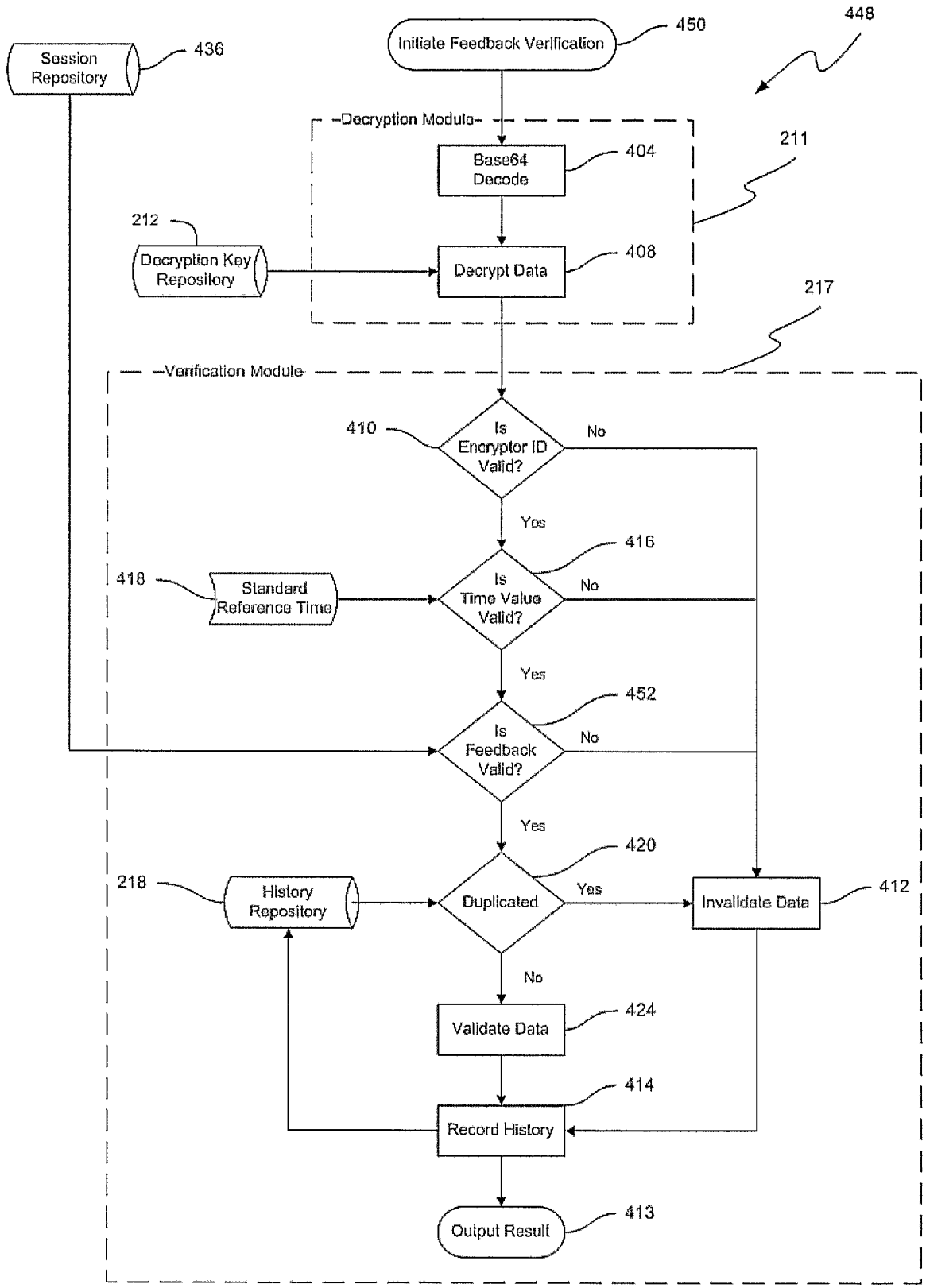

448
Initiate Feedback Verification
450
Session Repository
436
Decryption Module
211
Base64 Decode
404
212
Decryption Key Repository
Decrypt Data
408
217
Verification Module
410
Is Encryptor ID Valid?
No
Yes
416
418
Standard Reference Time
Is Time Value Valid?
No
Yes
452
Is Feedback Valid?
No
Yes
420
218
History Repository
Duplicated
Yes
Invalidate Data
412
No
Validate Data
424
414
Record History
Output Result
413

FIG. 4c

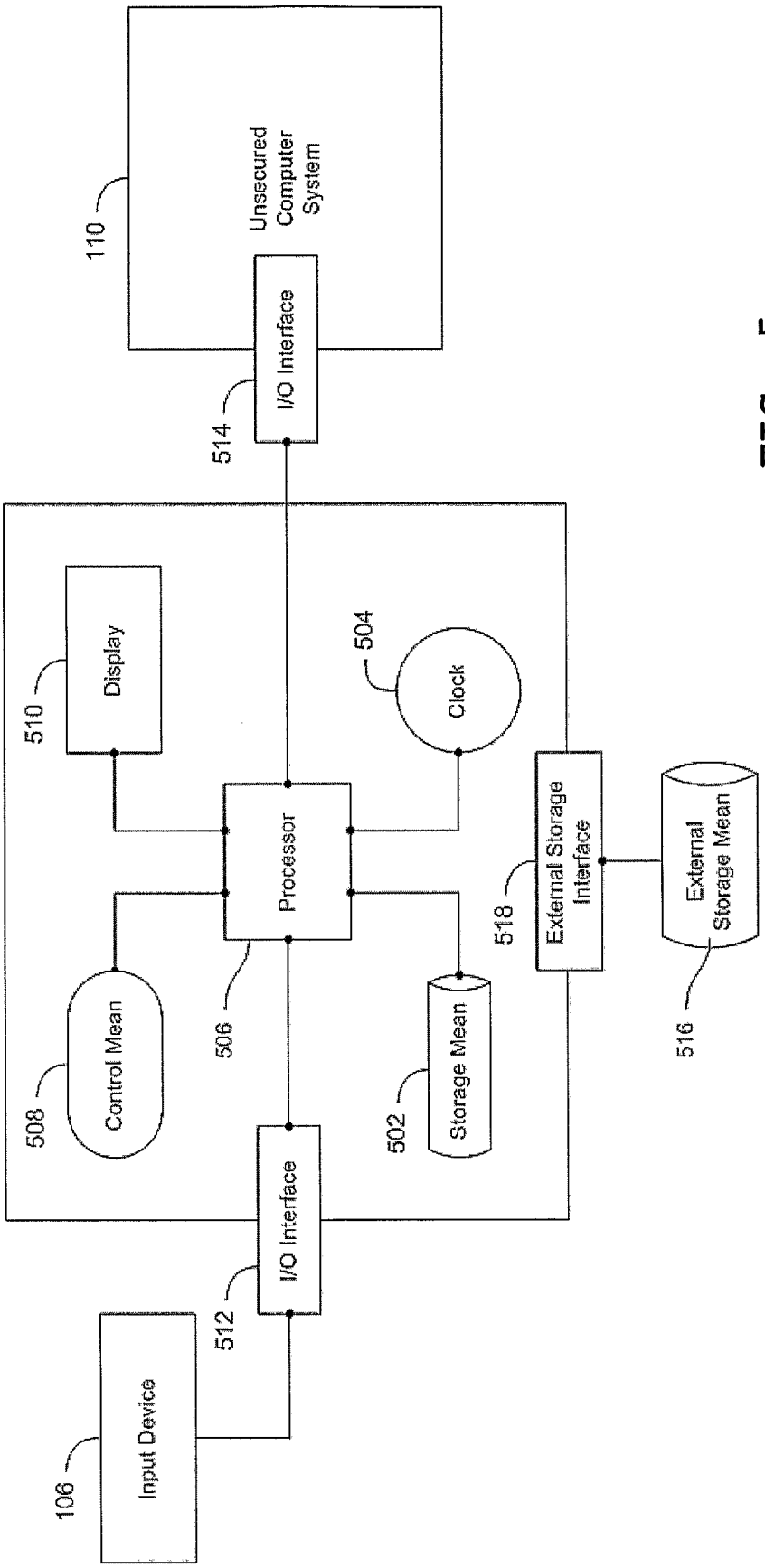

110
Unsecured Computer System
514
I/O Interface
510
Display
504
Clock
Processor
506
508
Control Mean
502
Storage Mean
518
External Storage Interface
516
External Storage Mean
512
I/O Interface
106
Input Device

FIG. 5