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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2017 (28.12.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/223478 A1

(51) International Patent Classification:
H04L 9/08 (2006.01)

(21) International Application Number:
PCT/US2017/039043

(22) International Filing Date:
23 June 2017 (23.06.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/192,755 24 June 2016 (24.06.2016) US

(71) Applicant: **NTT INNOVATION INSTITUTE, INC.** [US/US]; 1950 University Avenue, Suite 600, East Palo Alto, CA 94303 (US).

(72) Inventor: **YAMAMOTO, Go**; 1950 University Avenue, Suite 600, East Palo Alto, CA 94303 (US).

(74) Agent: **LOHSE, Timothy, W.**; DLA Piper LLP (US), 200 University Avenue, East Palo Alto, CA 94303 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: KEY MANAGEMENT SYSTEM AND METHOD

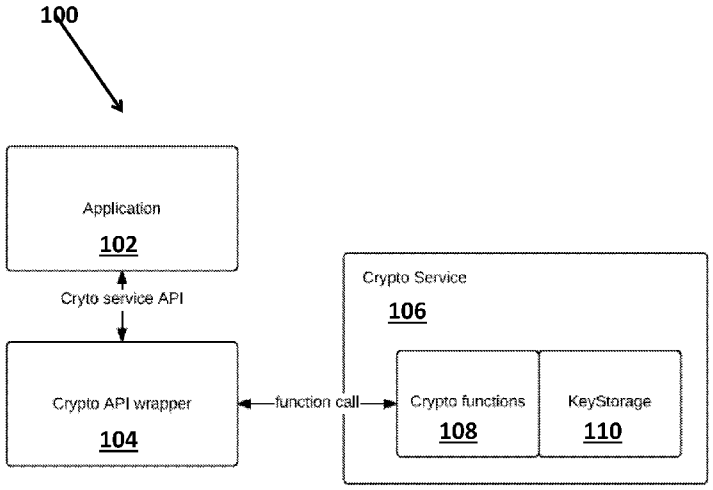

FIGURE 1

(57) Abstract: A system and method for private key management in a public key encryption system are disclosed. In one embodiment, the system and method may utilize a "fake" private key to provide the private key management.

WO 2017/223478 A1

KEY MANAGEMENT SYSTEM AND METHOD

Go Yamamoto

Field

The disclosure relates generally to a public key encryption system and method and in particular to the management of the private keys in the public key encryption system.

Background

Encryption is a well known technique used to obscure data or information, contained in a message, from unauthorized third parties. There are many different types of encryption that have been used. One popular type of encryption is public key encryption in which a public key and a private key are generated so that the private key and public key have a mathematical relationship that is computationally difficult to calculate at least from the private key to the public key. In other words, given only the public key, it is difficult to determine the private key. As a result, the public key can be publicly distributed (such as stored in a public key ring or key server) and the private key is kept securely by the user. When the user wants to send an encrypted message using public key encryption, the user finds the recipient's public key and encrypts the message using the public key of the recipient. When the recipient receives the message, the recipient uses his/her private key to decrypt the message. Similarly, when someone wants to send a message to the user, the message is encrypted using the public key of the user and the user decrypts the message using the secure private key. The advantage of public key encryption is that a private key of the user never has to be revealed or communicated to any third party.

Figure 1 illustrates an ideal encryption key system 100 in which the system includes an application 102 that has the capability to encrypt/decrypt messages using public key encryption. The application may utilize a crypto service API to connect to a crypto API wrapper 104 that manages the public key encryption process of the application including the storage of the private key. In the ideal system in Figure 1, the system also has a crypto service 106 that is accessible by a function call. The crypto service 106 may include crypto functions 108 and a key storage 110. The crypto service 106 may be securely hosted and protected from hackers and the like. In this ideal system, the crypto service 106 stores and manages the private key of the user and may also

perform the decryption functions for the applications so that the private key is maintained on the secure crypto service 106.

Figure 2 illustrates a typical encryption key system 200 that has the same application 202. In most typical implementations of the public key system, the application 202 may include a key storage 204 in which the private key of the user is stored and managed. As with the ideal system, the application may connect using a crypto function API to a crypto library 206 that has a plurality of crypto functions 208 to perform the encryption/decryption using the private key. In the typical system, to perform the encryption/decryption, the private key must be communicated from the key storage 204 in the application to the crypto library 206 which exposes the private key. In addition, with the system of Figure 2, it is very difficult to update the private keys since they are stored on each application 202. Furthermore, in an Internet of Things (IoT) type system, the management of the private keys and thus the public key encryption system is not feasible. Thus, it is desirable to provide a system and method for managing the private keys of a public key encryption system.

Brief Description of the Drawings

Figure 1 illustrates an ideal encryption key system;

Figure 2 illustrates a typical encryption key system;

Figure 3 illustrates a computer system network having a plurality of computing devices that uses public key encryption;

Figure 4 illustrates a encryption key system with encryption key management;

Figure 5 illustrates a decryption module of the system in Figure 2;

Figure 6 illustrates a first implementation of a decryption module that may be used with the encryption system in Figure 4; and

Figure 7 illustrates a second implementation of a decryption module that may be used with the encryption system in Figure 4.

Detailed Description of One or More Embodiments

The disclosure is particularly applicable to private key management in a public key encryption system and method using for an Internet of Things (IoT) network and it is in this context that the disclosure will be described. It will be appreciated, however, that the private key management system may be used to manage the private keys of other encryption systems and may be used for any sized network or computer installation.

Figure 3 illustrates a computer system network 300 having a plurality of computing devices 302, such as Device 1,…., Device N+1 as shown in Figure 3, that each use public key encryption. The computer system network may be an Internet of Things (IoT) system in which each computing device 302 communicates through and across a communications path 303 to other computing devices 302. In this system, each computing device 302 in the IoT has a processor and executes an application having encryption/decryption capabilities 304 or a background encryption/decryption routine. The encryption/decryption capabilities may, for example, allow the computing devices 302 to securely communicate with each other. The computer network system 300 may also have a crypto service 306 connected to the communications path 303 that manages the private keys of the encryption/decryption element of each computing device 302 as described below in more detail.

Each computing device 302 may be a processor based device that has one or more processors, memory, a persistent storage device, such as a disk drive or flash, a display (optional for some of the computing devices like a thermostat) and communications circuits that allow each computing device 302 to communicate with/to other elements/devices of services on the computer system network 300. For example, each computing device 302 may be a personal computer, a laptop or tablet computer, a device that controls a function of a household like a thermostat or a door lock, an appliance in a household such as a refrigerator or oven and the like. In one embodiment, each computing device may execute the encryption/decryption element (having a plurality of lines of computer code) to provide the encryption/decryption capability. In another embodiment, the encryption/decryption element may be a piece of hardware, such as an integrated circuit, field programmable gate array, a microcontroller, a microprocessor and the like, that performs the encryption/decryption capabilities.

The crypto service 306 may be implemented in hardware or software. If the crypto service 306 is implemented in software, it may be a plurality of lines of computer code that may be executed by a processor of the computer system that hosts the crypto service 306. The computer system may be a server computer, an application server, a blade service, a cloud computing resource and the like. If the crypto service 306 is implemented in hardware, it may be an integrated circuit, field programmable gate array, a microcontroller, a microprocessor and the like that performs the key management processes described below.

In the computer system 300 shown in Figure 3 or any other computer system, the private keys of each computing device 302 must be managed and maintained. The management and maintenance of the private keys of each computing device 302 can be very onerous for a large computer system or computer network like the IoT shown in Figure 3. The system described below with reference to Figure 4 as well as the decryption modules described below with reference to Figures 5-6 provide that private key management function as well as the encryption/decryption capabilities using the private key management function.

Figure 4 illustrates an encryption key system with key management. The encryption key system permits decrypting cyphertext to plaintext using key information that is stored on a remote source (such as the crypto service 306 in Figure 4) without changing interfaces of the decryption modules for each of the one or more computing device(s). As with other encryption solutions, so that the encryption key system is compatible with existing encryption systems, the application 304 on each computing device is present and the application has encryption/decryption capacities. The application 304 may have a key storage 400 that stores the “fake” private keys of the encryption key system.

The application 304 may interface with a known crypto function API to a crypto library 402. The crypto library 402 may be separate from the application 304 and located on the same computer system as the application 304 or may be located on a remote computing resource of the computer system network. The crypto library 402 may further include a metadata decoder 404 and a crypto API wrapper 406 that operate to perform the encryption/decryption requested by the application 304 and manage the transmission of the data. The meta data decoder 404 may receive the “fake” private key from the storage 400 in the application 304 and then locate the actual

private key in a remote key storage 410. The crypto API wrapper 406 may provide an interface for the encryption/decryption capabilities of the application 304 to the known crypto APIs that implement the encryption and decryption functions.

The encryption key system with key management may further comprise the crypto service element 306 that may be remote from the application 304 on the same computer system as the application 304 or on a different computer system in the computer system network 300. The crypto service element 306 may further comprise crypto functions 408, key storage 410 and a meta data encoder 412. The crypto functions 408 may store and distribute various crypto functions and manage the private key that may be stored in the key storage 410. As with any public key encryption system, the crypto functions 408 may include an encryption module/component to encrypt a message using the public key of the intended recipient and a decryption module/component for decrypting an incoming encrypted message using the private key of the user based on requests from the application 304 communicated using the crypto function APIs. The meta data encoder 412 may generate a "fake" private key 414 that corresponds to the actual private key for each application 304 (assuming each application has its own encryption/decryption element). The crypto service 306 may provide the fake private key 414 to key storage 400 of the encryption/decryption element of the application 304.

The key storage 410 may perform some of the key management functions in that it may store the private key for each entity as well as the corresponding fake private key so that the system is able to obtain the proper private key based on the fake private key to decrypt an encrypted piece of content.

The crypto library 402 and the crypto service 306 may each be implemented in hardware or software. When each of the crypto library 402 and the crypto service 306 are implemented in software, each of the crypto library 402 and the crypto service 306 may be a plurality of lines of computer code that may be executed by a processor of the computing resource/device on which the crypto library 402 or the crypto service 306 is hosted or stored. When each of the crypto library 402 and the crypto service 306 are implemented in hardware, each of the crypto library 402 and the crypto service 306 may be an integrated circuit, a field programmable gate array, a microcontroller, a microprocessor and the like that performs the functions of each component.

In operation, a sender of a message uses a public key of the user of the application 304 (or a public key associated with the application 304) to encrypt the message. When the application 304 receives a message to be decrypted, it retrieves the private key from the key storage 400 (which is the fake private key 414) and passes the fake private key 414 onto the crypto library 402 so that the meta data decoder 404 can decode the fake private key and request decryption using the private key from the key storage 410 where the actual private key is stored. The private key is then used to decrypt the message by the crypto functions 408 and the plaintext message is returned to the application 304. The fake private key 414 is a pointer to the actual private key stored in remote key storage 410. When the private key associated with the user or application needs to be updated: 1) a new private key may be generated by the crypto functions 408; 2) the private key may be stored in the remote key storage 410; 3) the meta data encoder 412 may generate a new fake private key 414 that acts as a pointer to the actual private key; and 4) the new fake private key may be sent to the application 304 and stored in the key storage 400. The application 304 uses the fake private key as if it is the actual private key so that management of the private key is easier. Similarly, for a new user or application, the system may: 1) generate a public key and private key pair; 2) generate a fake private key using the meta data encoder 412; 3) store the private key in the key storage 410; and 4) send the fake private key onto the application or new user for use as described above.

Using the system in Figures 3 and 4, the private keys associated with each computing device 302 and application 304 may be easily managed without changing interfaces of the decryption modules so that the system is compatible with existing public key encryption systems and their modules.

Figure 5 illustrates a decryption module 500 of the typical system in Figure 2 that may include a parse/validation component 502 and a decryption component/module 504 as shown. The decryption module 504 receives private Key information and Ciphertext as inputs and output an error signal and Plaintext. The key information is processed by parser to obtain a set of cryptographic keys that are expressed by a mathematically comprehensive form, and the set is validated to determine if the set retains integrity of valid keys. The private key(s) that are required for decryption process are chosen from the set and sent to Decryption process 504. The

decryption process 504 takes the ciphertext and Private Keys, and outputs plaintext by processing the decryption algorithm of the cipher.

Figure 6 illustrates a first implementation of a decryption module 600 that may be used with the encryption system in Figures 3 and 4. The decryption module 600 may be part of the application 304 on the device or it may be part of the crypto service 306. The decryption module 600 may have a parser/validator component 602, a decryption component 604, a meta-information extractor component 606 and a remote decryption client 608. Each of these elements of the decryption module 600 may be implemented in hardware or software. As already described above for the encryption system, the system has the remote crypto service module 306 that has key storage 410 for the actual private keys of the entities (in which entities include users, applications and computing devices that have the encryption/decryption capabilities) and a remote decryption server that may incorporate the crypto functions 408 of the system. The decryption module takes Key information and Ciphertext (encrypted content) as inputs and outputs Plaintext (the decrypted content that was decrypted using the private key). Any error signals from Parse, Validation component 602 and the decryption component 604 are trapped by meta-information extractor 606 before the module aborts. The meta information extractor 606 extracts server information from the Key information. If the Server information is successfully obtained, server information is sent to the remote decryption client 608, and the Remote decryption client 608 contacts the remote server and performs the decryption process using a decryption protocol and the remote decryption server that has the crypto functions 408. The remote decryption server may request the private key for the decryption based on the key information from the fake private key and then perform the decryption using the actual private key from the key storage 410. Then the Remote decryption client 608 outputs Plaintext as the output from the module.

For example, consider a decryption module for a RSA cryptosystem. Suppose Key information consists of N, E, D, where N is the public modulus of RSA cipher and E is the public exponent, and D is the private key. To encrypt plaintext m, we compute m^E mod N to obtain ciphertext. To decrypt ciphertext c, we compute c^D mod N. We can implement meta-information extractor as shown below.

(1) Check if 2^(DE) = 2 mod N. If true, then output error signal and abort.

(2) Parse D as a string to obtain URL of remote decryption server. If not successful output error signal and abort.

(3) Call remote decryption server with the URL on input.

Users can perform remote decryption without changing the interface of the decryption module. To perform remote decryption,

(1) Get public key N, E, and string S that presents the URL of remote decryption server.

(2) Encode S as a big number to obtain D.

(3) Use N, E, D as the Key information.

As long as RSA cryptosystem is considered secure, it is practically impossible to collide D obtained above with the true public key showing that the above processes work.

If Key information carries more entries, then we can use them to embed server information. For example, if Key information has entries P and Q, prime numbers that satisfy N=PQ, then we can encode some part of URL, or the whole URL, to P or/and Q. The above system would also operate with other public key algorithms and/or elliptic curve cryptographies such as ECDH key agreement or EC ElGamal cipher.

Figure 7 illustrates a second implementation of a decryption module that may be used with the encryption system in Figure 4 that has similar elements to those shown in Figure 6 that have a similar function (designated with the same reference numbers) and the description will not be repeated here. The decryption module 600 may be part of the application 304 on the device or it may be part of the crypto service 306. In this second implementation, the meta-information extractor 606 may be placed before the Parser and Validator 602. In this configuration, the meta-information extractor 606 may be implemented as shown below.

(1) Parse D as a string to obtain URL of remote decryption server. If not successful call Parse and Validate.

(2) Call remote decryption server with the URL on input.

This implementation works practically because it is very unlikely that the true private key has integrity as a URL string, since the private key is supposed to be chosen as a uniformly random number.

If Key information carries more entries, then the system can use them to embed server information. For example, if Key information has entries P and Q, prime numbers that satisfy N=PQ, then we can encode some part of URL, or the whole URL, to P or/and Q. In this situation we can implement meta-information extractor as shown below.

(1) Check D=0 and parse P as a string to obtain URL of remote decryption server. If not successful call Parse and Validate.

(2) Call remote decryption server with the URL on input.

The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the disclosure to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the disclosure and its practical applications, to thereby enable others skilled in the art to best utilize the disclosure and various embodiments with various modifications as are suited to the particular use contemplated.

The system and method disclosed herein may be implemented via one or more components, systems, servers, appliances, other subcomponents, or distributed between such elements. When implemented as a system, such systems may include an/or involve, inter alia, components such as software modules, general-purpose CPU, RAM, etc. found in general-purpose computers,. In implementations where the innovations reside on a server, such a server may include or involve components such as CPU, RAM, etc., such as those found in general-purpose computers.

Additionally, the system and method herein may be achieved via implementations with disparate or entirely different software, hardware and/or firmware components, beyond that set forth above. With regard to such other components (e.g., software, processing components, etc.) and/or computer-readable media associated with or embodying the present inventions, for example, aspects of the innovations herein may be implemented consistent with numerous general purpose or special purpose computing systems or configurations. Various exemplary computing systems, environments, and/or configurations that may be suitable for use with the innovations herein may include, but are not limited to: software or other components within or embodied on

personal computers, servers or server computing devices such as routing/connectivity components, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, consumer electronic devices, network PCs, other existing computer platforms, distributed computing environments that include one or more of the above systems or devices, etc.

In some instances, aspects of the system and method may be achieved via or performed by logic and/or logic instructions including program modules, executed in association with such components or circuitry, for example. In general, program modules may include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular instructions herein. The inventions may also be practiced in the context of distributed software, computer, or circuit settings where circuitry is connected via communication buses, circuitry or links. In distributed settings, control/instructions may occur from both local and remote computer storage media including memory storage devices.

The software, circuitry and components herein may also include and/or utilize one or more type of computer readable media. Computer readable media can be any available media that is resident on, associable with, or can be accessed by such circuits and/or computing components. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and can accessed by computing component. Communication media may comprise computer readable instructions, data structures, program modules and/or other components. Further, communication media may include wired media such as a wired network or direct-wired connection, however no media of any such type herein includes transitory media. Combinations of the any of the above are also included within the scope of computer readable media.

In the present description, the terms component, module, device, etc. may refer to any type of logical or functional software elements, circuits, blocks and/or processes that may be implemented in a variety of ways. For example, the functions of various circuits and/or blocks can be combined with one another into any other number of modules. Each module may even be implemented as a software program stored on a tangible memory (e.g., random access memory, read only memory, CD-ROM memory, hard disk drive, etc.) to be read by a central processing unit to implement the functions of the innovations herein. Or, the modules can comprise programming instructions transmitted to a general purpose computer or to processing/graphics hardware via a transmission carrier wave. Also, the modules can be implemented as hardware logic circuitry implementing the functions encompassed by the innovations herein. Finally, the modules can be implemented using special purpose instructions (SIMD instructions), field programmable logic arrays or any mix thereof which provides the desired level performance and cost.

As disclosed herein, features consistent with the disclosure may be implemented via computer-hardware, software and/or firmware. For example, the systems and methods disclosed herein may be embodied in various forms including, for example, a data processor, such as a computer that also includes a database, digital electronic circuitry, firmware, software, or in combinations of them. Further, while some of the disclosed implementations describe specific hardware components, systems and methods consistent with the innovations herein may be implemented with any combination of hardware, software and/or firmware. Moreover, the above-noted features and other aspects and principles of the innovations herein may be implemented in various environments. Such environments and related applications may be specially constructed for performing the various routines, processes and/or operations according to the invention or they may include a general-purpose computer or computing platform selectively activated or reconfigured by code to provide the necessary functionality. The processes disclosed herein are not inherently related to any particular computer, network, architecture, environment, or other apparatus, and may be implemented by a suitable combination of hardware, software, and/or firmware. For example, various general-purpose machines may be used with programs written in accordance with teachings of the invention, or it may be more convenient to construct a specialized apparatus or system to perform the required methods and techniques.

Aspects of the method and system described herein, such as the logic, may also be implemented as functionality programmed into any of a variety of circuitry, including programmable logic devices ("PLDs"), such as field programmable gate arrays ("FPGAs"), programmable array logic ("PAL") devices, electrically programmable logic and memory devices and standard cell-based devices, as well as application specific integrated circuits. Some other possibilities for implementing aspects include: memory devices, microcontrollers with memory (such as EEPROM), embedded microprocessors, firmware, software, etc. Furthermore, aspects may be embodied in microprocessors having software-based circuit emulation, discrete logic (sequential and combinatorial), custom devices, fuzzy (neural) logic, quantum devices, and hybrids of any of the above device types. The underlying device technologies may be provided in a variety of component types, e.g., metal-oxide semiconductor field-effect transistor ("MOSFET") technologies like complementary metal-oxide semiconductor ("CMOS"), bipolar technologies like emitter-coupled logic ("ECL"), polymer technologies (e.g., silicon-conjugated polymer and metal-conjugated polymer-metal structures), mixed analog and digital, and so on.

It should also be noted that the various logic and/or functions disclosed herein may be enabled using any number of combinations of hardware, firmware, and/or as data and/or instructions embodied in various machine-readable or computer-readable media, in terms of their behavioral, register transfer, logic component, and/or other characteristics. Computer-readable media in which such formatted data and/or instructions may be embodied include, but are not limited to, non-volatile storage media in various forms (e.g., optical, magnetic or semiconductor storage media) though again does not include transitory media. Unless the context clearly requires otherwise, throughout the description, the words "comprise," "comprising," and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in a sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively. Additionally, the words "herein," "hereunder," "above," "below," and words of similar import refer to this application as a whole and not to any particular portions of this application. When the word "or" is used in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in the list, all of the items in the list and any combination of the items in the list.

Although certain presently preferred implementations of the invention have been specifically described herein, it will be apparent to those skilled in the art to which the invention pertains that variations and modifications of the various implementations shown and described herein may be made without departing from the spirit and scope of the invention. Accordingly, it is intended that the invention be limited only to the extent required by the applicable rules of law.

While the foregoing has been with reference to a particular embodiment of the disclosure, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the disclosure, the scope of which is defined by the appended claims.

Claims:

1. A key management method for public key encryption, comprising:

storing, remotely from an entity, a plurality of set of keys for entities in the system that have decryption capabilities, the set of keys for each entity having a private key of the entity and a fake private key corresponding to the private key of the entity;

receiving, from the entity, a piece of ciphertext content and a fake private key; and

obtaining, when the piece of ciphertext content is being decrypted for a particular entity, the private key of the entity based on the key private key.

2. The method of claim 1 further comprising generating, by the cryptographic component, a new private key for the particular entity and a fake private key that corresponds to the new private key for the particular entity and sending, by the cryptographic component, the fake private key to the particular entity.

3. The method of claim 2 further comprising decrypting the piece of ciphertext content using the private key of the entity.

4. The method of claim 1, wherein obtaining the private key for the particular entity further comprises extracting a uniform resource locator of the cryptographic component from key information and retrieving the new private key of the particular entity based on the uniform resource locator of the cryptographic component and the private fake key.

5. The method of claim 1, wherein each entity is a computing device with application that has decryption capabilities.

6. The method of claim 1, wherein each entity is one of an application having a decryption capability, a device having a decryption capability and a user that uses an application with a decryption capability.

7. The method of claim 1, wherein the fake private key for the particular entity further comprises a pointer to new private key for the particular entity.

8. A key management system, comprising:

one or more entities, each entity having a public key decryption capability using a private key of the entity and a key store;

a cryptographic component, connected to the entities by a computer network, having a metadata encoder that generates a new private key for a particular entity and generates a fake private key that corresponds to the new private key; and

wherein the cryptographic component sends the fake private key to the particular entity.

9. The system of claim 8, wherein the cryptographic component performs a decryption process based on a request from the particular client wherein the request includes a piece of ciphertext and the fake private key and the cryptographic component obtains the new private key of the particular entity based on the fake private key to permit the decryption of the ciphertext using the new private key of the particular entity.

10. The system of claim 9, wherein each entity has a decryption module that performs the decryption of the ciphertext using the new private key of the particular entity.

11. The system of claim 10, wherein the decryption module further comprises a parser module that receives key information and a meta-information extractor module obtains the new private key for the particular entity based on the fake private key.

12. The system of claim 11, wherein the meta-information extractor module extracts a uniform resource locator of the cryptographic component from the key information and retrieves the new private key of the particular entity based on the uniform resource locator of the cryptographic component.

13. The system of claim 8, wherein each entity is a computing device with application that has decryption capabilities.

14. The system of claim 8, wherein each entity is one of an application having a decryption capability, a device having a decryption capability and a user that uses an application with a decryption capability.

15. The system of claim 8, wherein the fake private key for the particular entity further comprises a pointer to new private key for the particular entity.

16. A public key encryption system, comprising:

one or more entities, each entity having a public key decryption capability using a private key of the entity and a key store;

a cryptographic component, connected to the entities by a computer network, having a metadata encoder that generates a new private key for a particular entity and generates a fake private key that corresponds to the new private key; and

wherein the cryptographic component sends the fake private key to the particular entity.

17. The system of claim 16, wherein the cryptographic component performs a decryption process based on a request from the particular client wherein the request includes a piece of ciphertext and the fake private key and the cryptographic component obtains the new private key of the particular entity based on the fake private key to permit the decryption of the ciphertext using the new private key of the particular entity.

18. The system of claim 17, wherein each entity has a decryption module that performs the decryption of the ciphertext using the new private key of the particular entity.

19. The system of claim 18, wherein the decryption module further comprises a parser module that receives key information and a meta-information extractor module obtains the new private key for the particular entity based on the fake private key.

20. The system of claim 19, wherein the meta-information extractor module extracts a uniform resource locator of the cryptographic component from the key information and retrieves the new private key of the particular entity based on the uniform resource locator of the cryptographic component.

21. The system of claim 16, wherein each entity is a computing device with application that has decryption capabilities.

22. The system of claim 16, wherein each entity is one of an application having a decryption capability, a device having a decryption capability and a user that uses an application with a decryption capability.

23. The system of claim 16, wherein the fake private key for the particular entity further comprises a pointer to new private key for the particular entity.

24. A key management method, comprising:

generating a private key for an entity;

storing the generated private key in a key management component having a key storage remote from the entity;

generating, at a key management component, a fake private key that acts as a pointer to the generated private key; and

sending the fake private key to the entity to store in a key storage, wherein the entity decrypts a piece of ciphertext by making a request using the fake private key.

25. The method of claim 24 further comprising generating a public key for the entity wherein the entity is a new entity.

26. The method of claim 24, wherein generating the private key for an entity further comprises generating a new private key for an existing entity.

27. The method of claim 24, wherein the entity is a computing device having a decryption capability.

28. The method of claim 24, wherein the entity is one of an application having a decryption capability, a device having a decryption capability and a user that uses an application with a decryption capability.

29. The method of claim 24, wherein the fake private key for the particular entity further comprises a pointer to new private key for the particular entity.

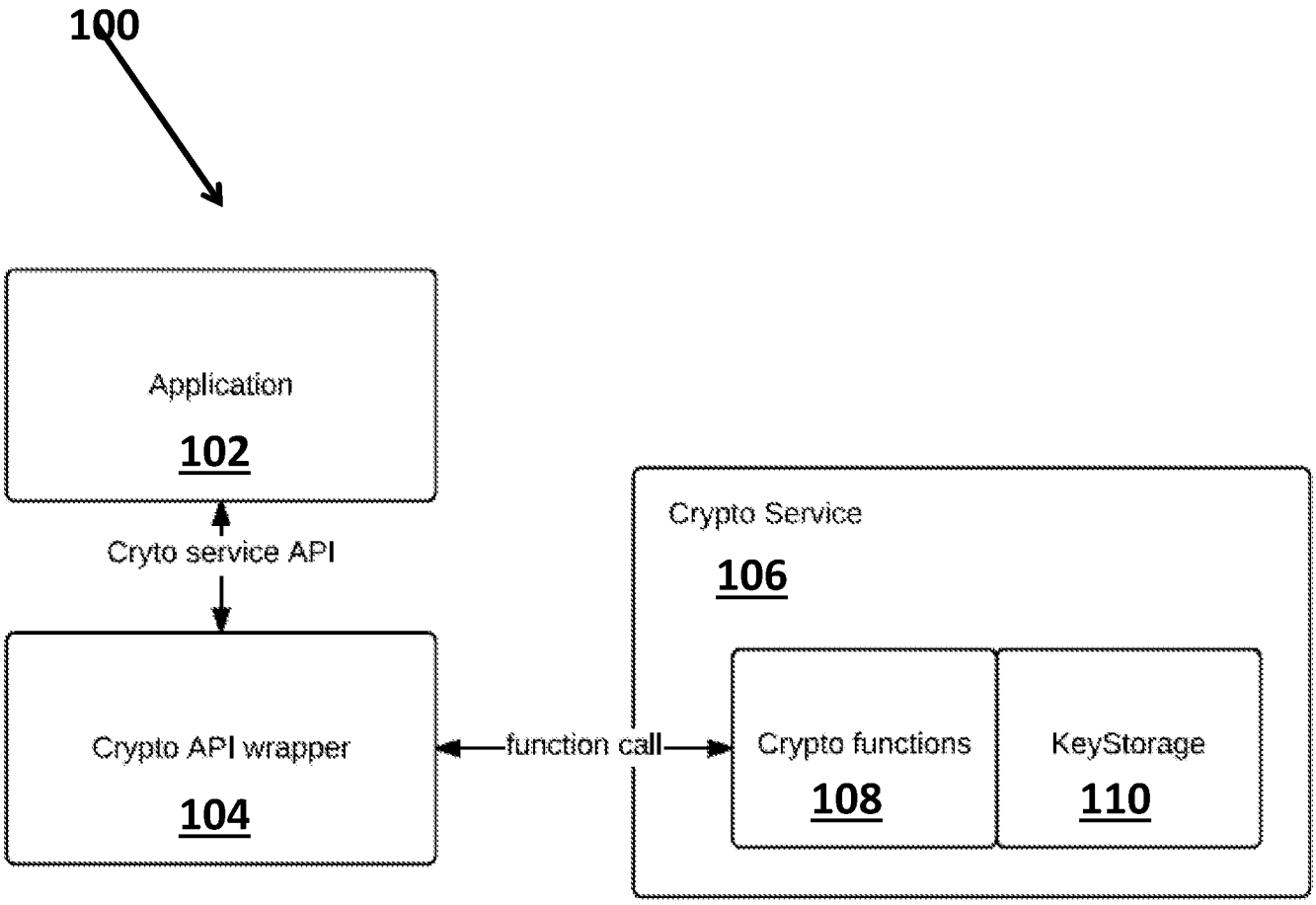
100
Application
102
Cryto service API
Crypto API wrapper
104
function call
Crypto Service
106
Crypto functions
108
KeyStorage
110

FIGURE 1

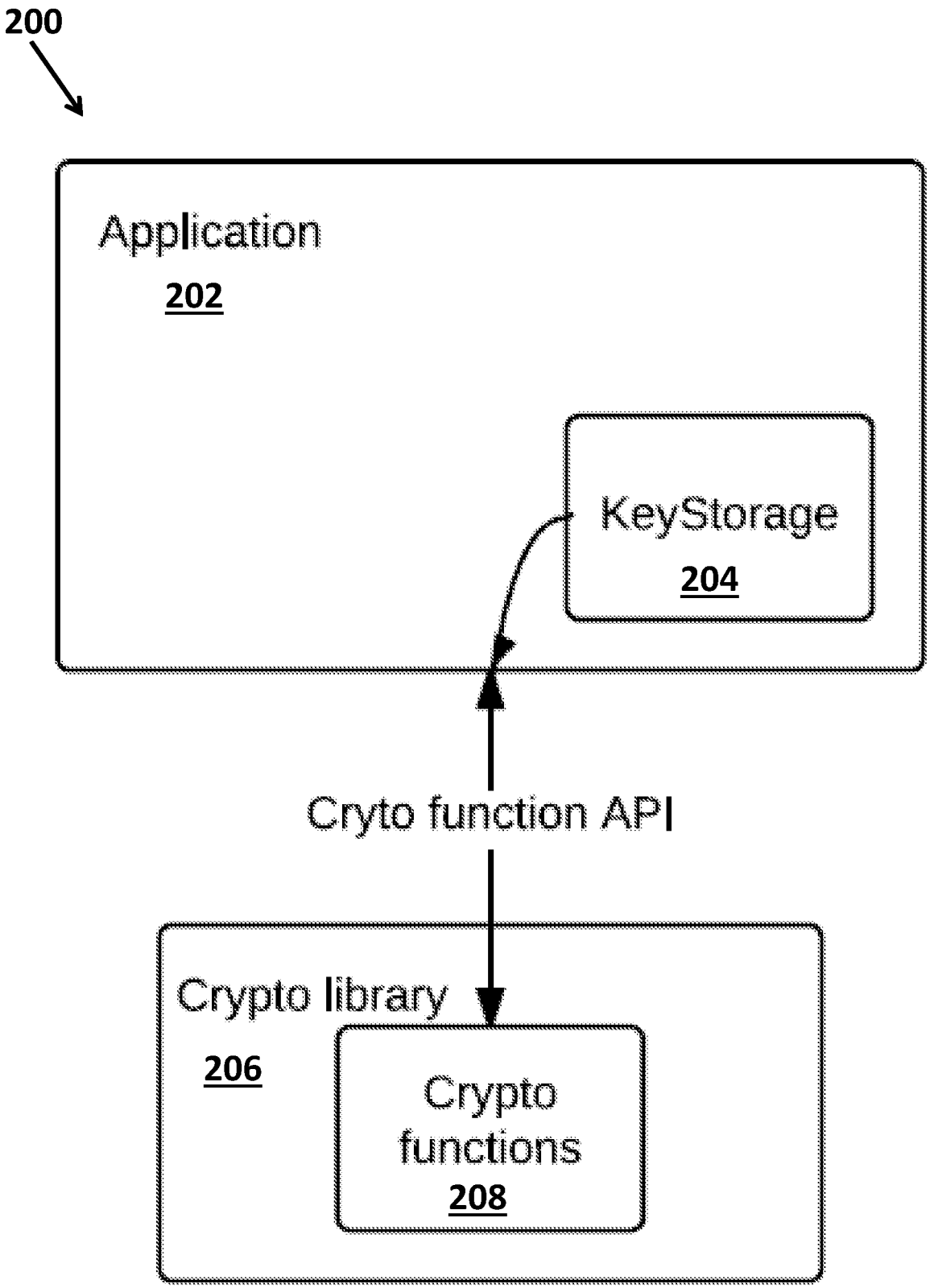
200
Application
202
KeyStorage
204
Cryto function API
Crypto library
206
Crypto functions
208

FIGURE 2

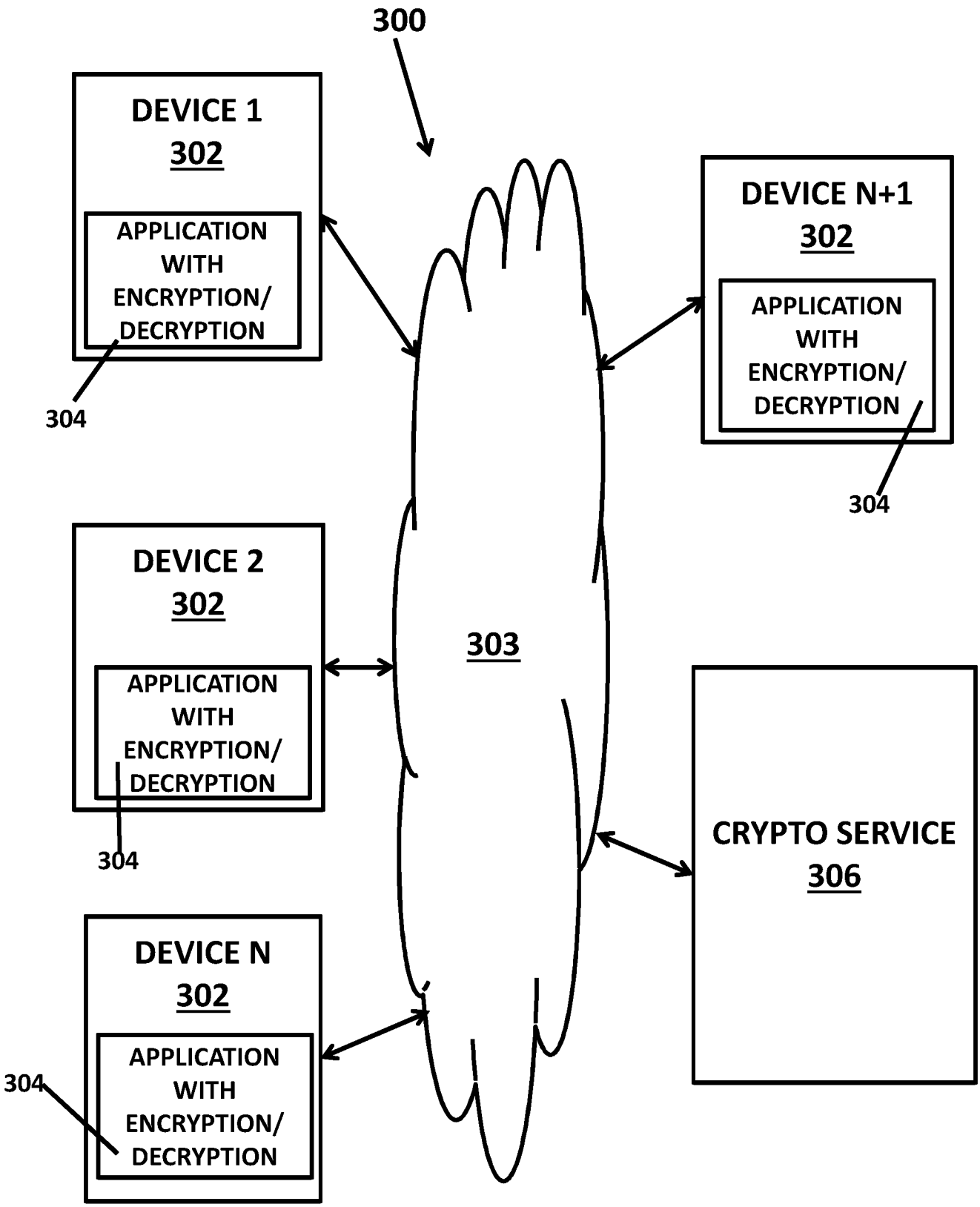
300
DEVICE 1
302
APPLICATION WITH ENCRYPTION/ DECRYPTION
304
DEVICE N+1
302
APPLICATION WITH ENCRYPTION/ DECRYPTION
304
DEVICE 2
302
APPLICATION WITH ENCRYPTION/ DECRYPTION
304
303
CRYPTO SERVICE
306
DEVICE N
302
APPLICATION WITH ENCRYPTION/ DECRYPTION
304

FIGURE 3

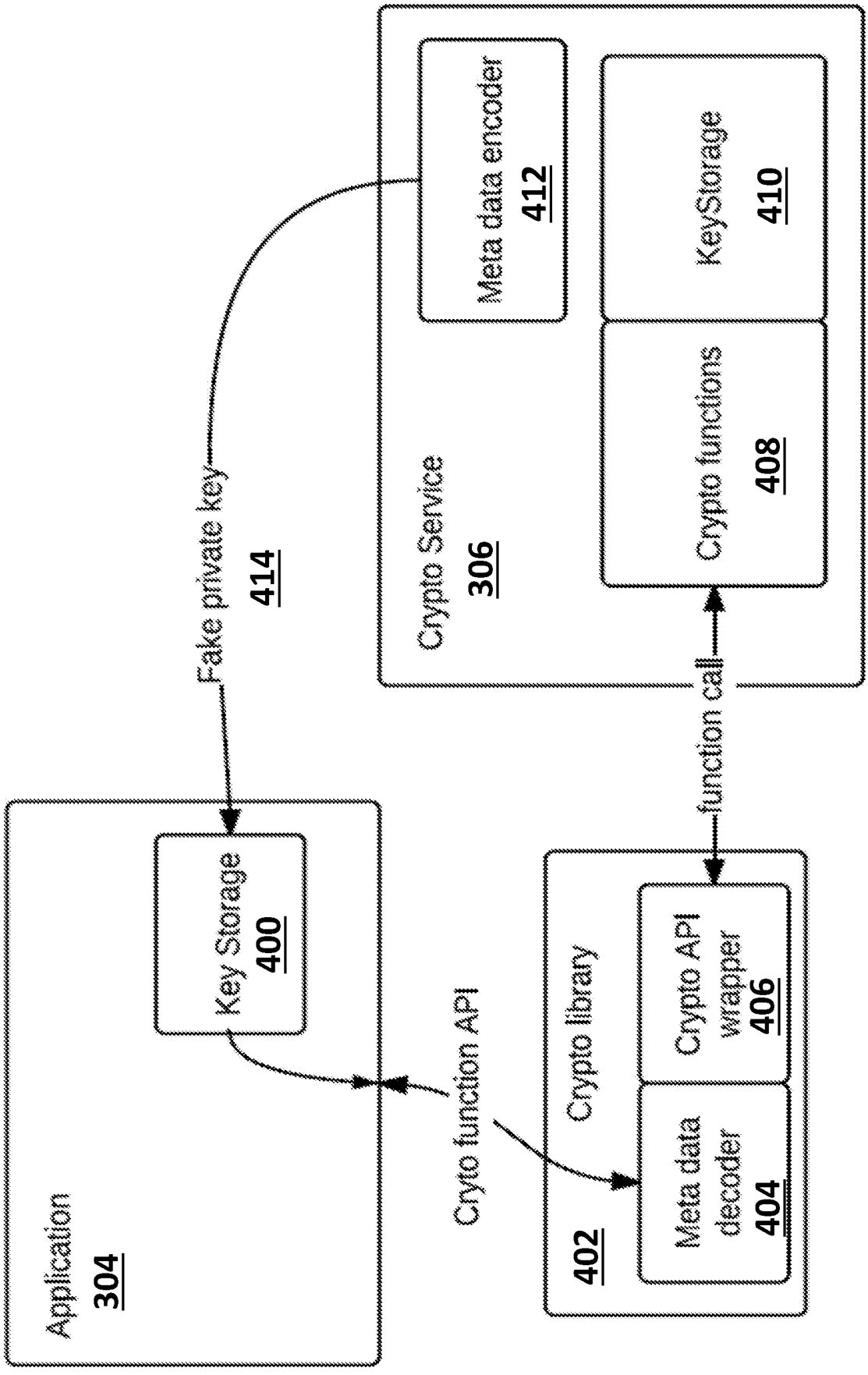
Application
304
Key Storage
400
Fake private key
414
Cryto function API
Crypto Service
306
Meta data encoder
412
Crypto functions
408
KeyStorage
410
function call
Crypto library
402
Meta data decoder
404
Crypto API wrapper
406

FIGURE 4

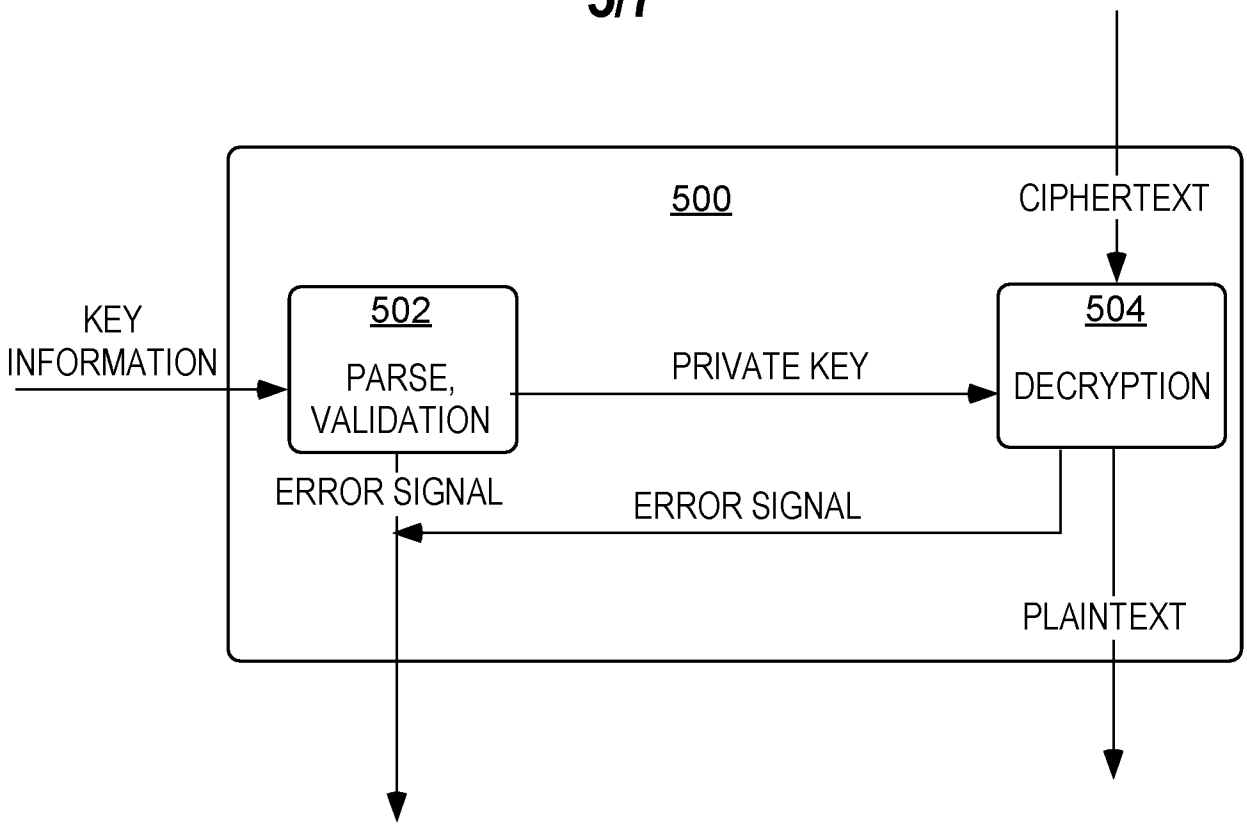
500
CIPHERTEXT
KEY
INFORMATION
502
PARSE,
VALIDATION
PRIVATE KEY
504
DECRYPTION
ERROR SIGNAL
ERROR SIGNAL
PLAINTEXT

FIGURE 5

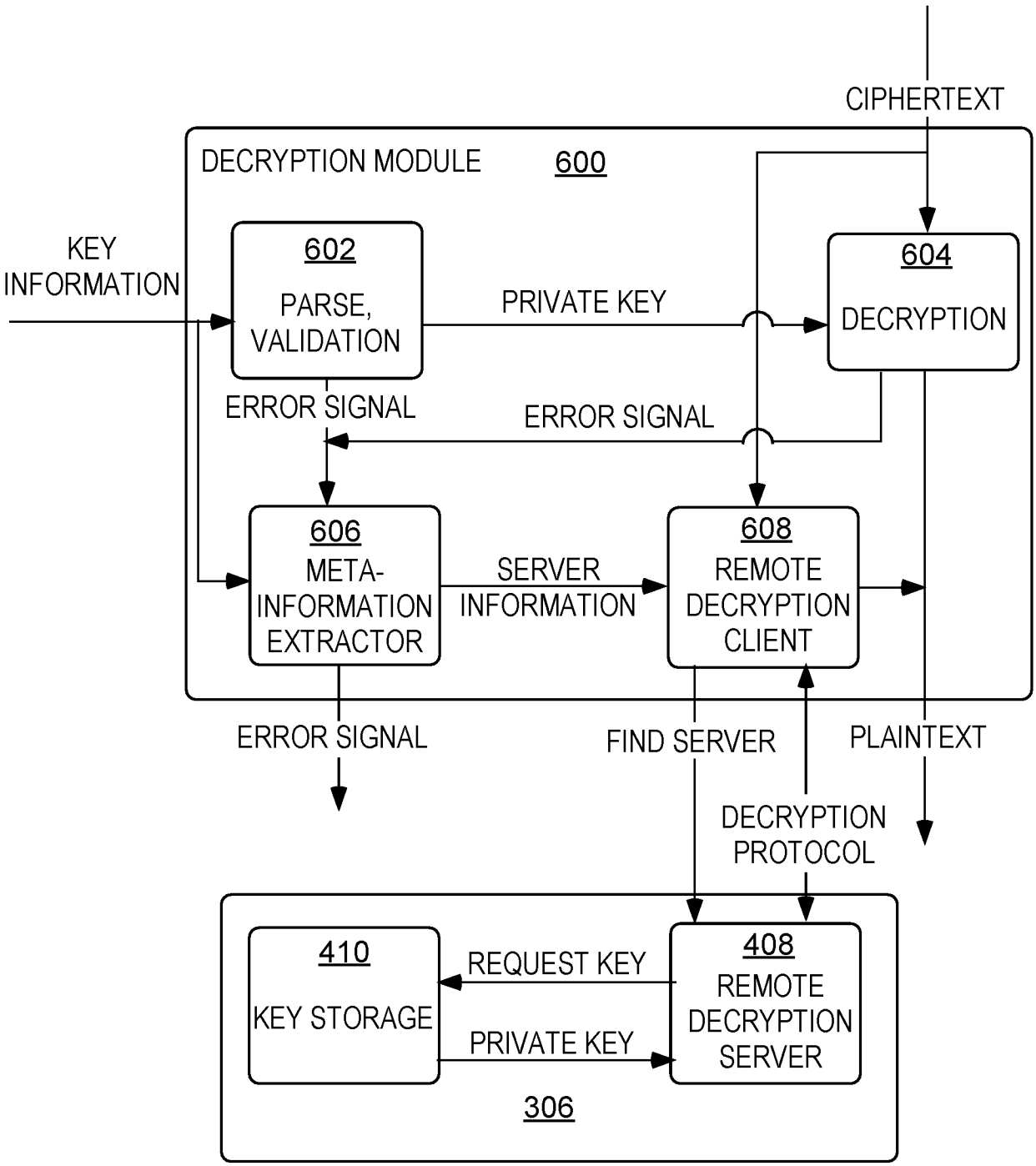
CIPHERTEXT
DECRYPTION MODULE
600
KEY INFORMATION
602
PARSE, VALIDATION
PRIVATE KEY
604
DECRYPTION
ERROR SIGNAL
ERROR SIGNAL
606
META-INFORMATION EXTRACTOR
SERVER INFORMATION
608
REMOTE DECRYPTION CLIENT
ERROR SIGNAL
FIND SERVER
PLAINTEXT
DECRYPTION PROTOCOL
410
KEY STORAGE
REQUEST KEY
PRIVATE KEY
408
REMOTE DECRYPTION SERVER
306

FIGURE 6

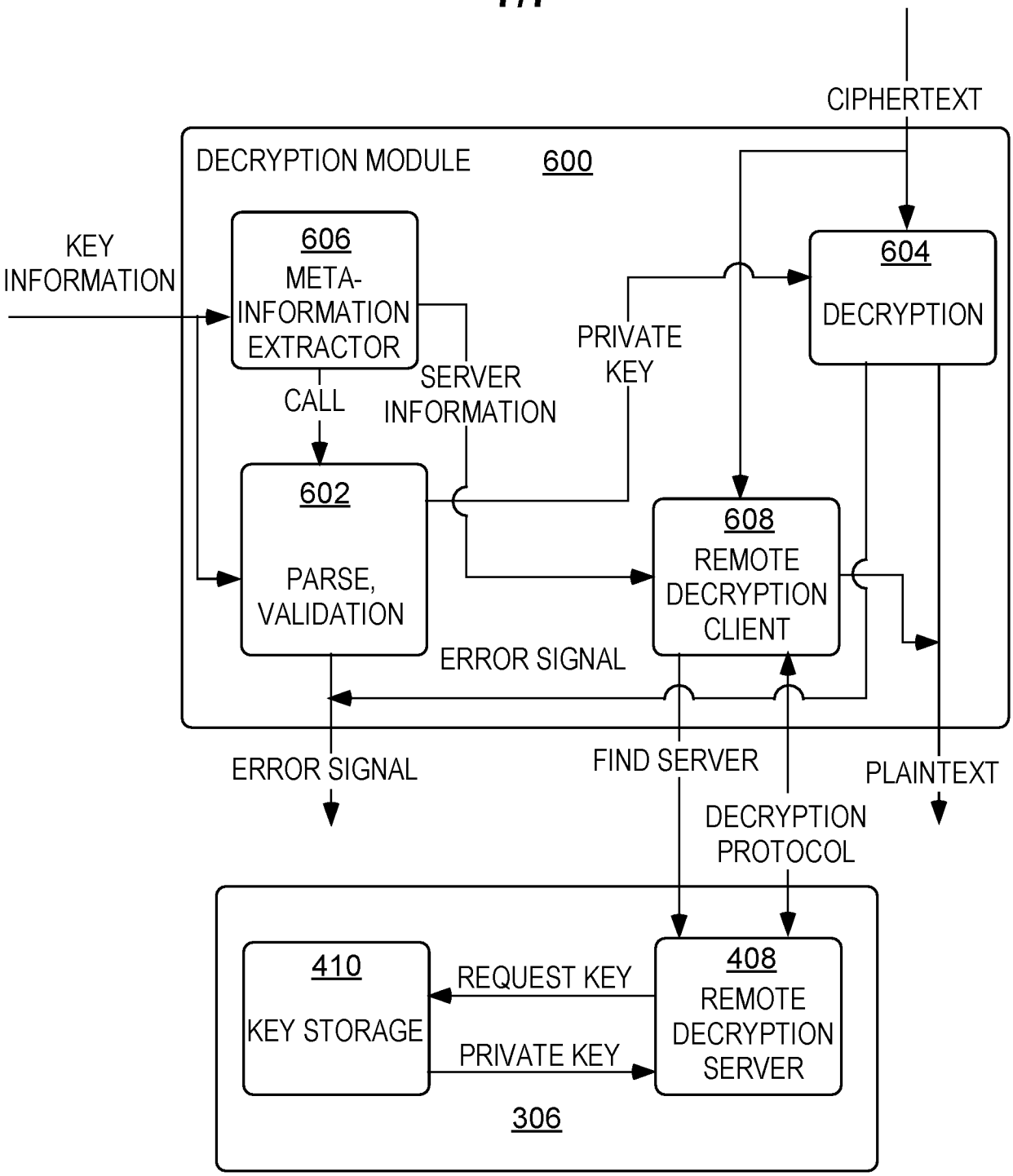
CIPHERTEXT
DECRYPTION MODULE
600
KEY INFORMATION
606
META-INFORMATION EXTRACTOR
604
DECRYPTION
PRIVATE KEY
SERVER INFORMATION
CALL
602
PARSE, VALIDATION
608
REMOTE DECRYPTION CLIENT
ERROR SIGNAL
ERROR SIGNAL
FIND SERVER
PLAINTEXT
DECRYPTION PROTOCOL
410
KEY STORAGE
REQUEST KEY
PRIVATE KEY
408
REMOTE DECRYPTION SERVER
306

FIGURE 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 17/39043

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 9/08 (2017.01)
CPC - H04L 9/08, H04L 9/0883, H04L 9/0836, H04L 9/0891, H04L 63/0428, H04L 9/30, H04L 63/0442, H04L 9/0825, H04L 9/083, G06F 21/10

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)
See Search History Document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/0128535 A1 (Cheng) 01 July 2004 (01.07.2004), entire document especially paras [0024], [0026], [0027], [0028], [0029], [0031]	1-29
A	US 2007/0136607 A1 (Launchbury et al.) 14 June 2007 (14.06.2007), entire document	1-29
A	US 2003/0188181 A1 (Kunitz et al.) 02 October 2003 (02.10.2003), entire document	1-29
A	US 2009/0287706 A1 (Bourges-Waldegg et al.) 19 November 2009 (19.11.2009), entire document	1-29
A	US 4,941,176 A (Matyas et al.) 10 July 1990 (10.07.1990), entire document	1-29

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
28 August 2017 (28.08.2017)	15 SEP 2017
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)