Technical Disclosure Commons

Defensive Publications Series

September 2021

SYSTEM AND METHOD FOR REVOKING COMPROMISED CRYPTOGRAPHIC KEY FROM DATABASE

DANIEL MASNY
Visa

GAVEN WATSON
Visa

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
MASNY, DANIEL and WATSON, GAVEN, "SYSTEM AND METHOD FOR REVOKING COMPROMISED CRYPTOGRAPHIC KEY FROM DATABASE", Technical Disclosure Commons, (September 27, 2021)
https://www.tdcommons.org/dpubs_series/4624

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

TITLE: "SYSTEM AND METHOD FOR REVOKING COMPROMISED CRYPTOGRAPHIC KEY FROM DATABASE"

VISA

DANIEL MASNY

GAVEN WATSON

TECHNICAL FIELD

The present subject matter relates to field of data security, more particularly, but not exclusively to a system and method for revoking compromised cryptographic keys.

BACKGROUND

In the process of establishing a working key, security of the working key is dependent on the security of all algorithms and keys involved in the process. The working keys such as Personal Identification Number (PIN) generation key are dependent on the security of the Key-Encryption-Key (KEK) used during its exchange, which in turn is dependent on the security of any other key used in its exchange. A PIN generation key is a cryptographic key used in the generation of a PIN (a secret number which is used with a bank car to withdraw money from a cash machine or Automated Teller Machine (ATM)). Similarly, a KEK is a cryptographic key that is used for encrypting another key. A KEK, maybe exchanged under another asymmetric encryption key in the form of a key encapsulation mechanism (KEM). The Key Encapsulation Mechanisms (KEMs) are a class of encryption techniques designed to secure symmetric cryptographic key material for transmission using asymmetric (public-key) algorithms. In case of all keys their security depends on the security of the algorithm under which they were exchange. In a hybrid scheme, using multiple exchange methods, there are dependencies between the keys and multiple key encapsulation mechanisms.

Additionally, each cryptographic algorithm requires a choice of parameters which determine a security level. It is challenging to find optimal parameter sets for a desired security level and there may be a case that specific choices induce weaknesses. For example, the case when choosing an elliptic curve or a ring for lattice-based cryptography.

In present scenarios, the dependencies of the working key are often not stored in a central database and after the working key is established, information of how the working key was exchanged is potentially lost. Therefore, there is a security concern that if the security of the establishing process of the working key was compromised then the working key needs to be replaced. Also, as soon as the security of a cryptographic algorithm is compromised, the algorithm and all keys that are used in the cryptographic algorithm should be replaced.

The information disclosed in this background of the disclosure section is only for enhancement of understanding of the general background of the invention and should not be taken as an acknowledgement or any form of suggestion that this information forms the prior art already known to a person skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, serve to explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

Figure 1 illustrates exemplary environment of a revoking system for revoking compromised cryptographic key from database, in accordance with some embodiments of the present disclosure;

Figure 2 illustrates an example graph showing method for revoking compromised cryptographic key from database, in accordance with some embodiments of the present disclosure;

Figure 3 illustrates flow diagram showing method for revoking compromised cryptographic key from database, in accordance with some embodiments of the present disclosure; and

Figure 4 illustrates a block diagram of an exemplary computer system for implementing embodiments consistent with the present disclosure.

The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

The present disclosure proposes a system and method for revoking compromised cryptographic key. The proposed system provides a database which stores information based on how the cryptographic key was established and dependencies of the cryptographic keys. The proposed system provides data security by revoking keys which have a dependency on the compromised cryptographic key.

Figure 1 illustrates an exemplary environment 100 of a revocation system 101 which is configured to revoke the compromised cryptographic key from a database 102. In an embodiment, the database 102 may be a centralized database which allows to efficiently revoke compromised cryptographic keys by storing dependencies of the cryptographic keys. The cryptographic key is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa. In an embodiment, the database 102 does not necessarily store the cryptographic key directly. The database 102 stores an identifier which does not leak information about the cryptographic keys and the identifier may be easily derived from the cryptographic keys. In an embodiment, the database 102 may be referred as a graph-based key revocation which stores all the dependencies in form of a graph. The cryptographic keys and encapsulation mechanisms are nodes, and the dependencies are edges in the graph. In an embodiment, each cryptographic key is assigned a key identifier. The key identifier is a salted hash of the cryptographic key. In cryptography, a salt is random data that is used as an additional input to a one-way function such as a hash function to randomize the input, salts are often used to safeguard passwords in storage. In an embodiment, the node in the graph only stores the key identifier and not the cryptographic key. The key identifier may be published without compromising confidentiality of the cryptographic key. The environment 100 for revoking the compromised cryptographic key from the database 102 providing includes the database 102, a communication network 103 and the revocation system 101. For example, consider **Figure 2**, which illustrates an example graph showing method for revoking the compromised cryptographic key from the database 102. In an embodiment, the graph in **Figure 2** comprises four layers. In an embodiment, top layer 201 consists of nodes which represent different key encapsulation schemes. The different key encapsulation schemes may include, but is not limited to, Rivest-Shamir-Adleman (RSA) 2048, Post-Quantum Cryptography (PQC) Key Encapsulation Mechanism (KEM) 128 and so on. In an embodiment, for each parameter choice of the key encapsulation there is a separate node. In an embodiment, second layer 202 consists of nodes for asymmetric keys. Asymmetric encryption comprises public-private key pairs to encrypt and decrypt data or text. In an embodiment, **Figure 2** shows three asymmetric keys such as asymmetric key 1, asymmetric key 2 and asymmetric key 3. In an embodiment, third layer 203 consists of nodes for the KEKs such as KEK 1 and KEK 2. In an embodiment, fourth layer 204 consists of nodes for the working keys such as PIN key 1, PIN key 2 and PIN key 3. In an embodiment, the asymmetric key which is been generated using one or more key encapsulations consists of multi-edge from the key encapsulation mechanism to the asymmetric key. For example, in **Figure 2** the asymmetric key 2 is generated from the RSA 2048 and the PQC KEM 128 and thus has two edges. In an embodiment, for the KEK which has been transferred using the asymmetric key, consists of an edge from the asymmetric key to the KEK. For example, in **Figure 2** there is an edge from the asymmetric key 1 to the KEK 1. In an embodiment, there is a directed edge from the KEK to each of the working key which has been encrypted under the KEK. For example, in Figure 2 the PIN key 1 and the PIN key 2 have a directed edge to the KEK 1. In another embodiment, authenticity may be included by including signatures and Certification Authorities (CA) in the graph. The CAs may be included by adding another top-level category in addition to the top layer 201.

Further, in an embodiment, during a key revocation or key encapsulation mechanism revocation, the dependencies may be easily traced in the graph. The revocation system 101 may be configured to start the process of revocation from the node of the compromised key or key encapsulations. Upon identifying the compromised key, the revocation system 101 iterates over child nodes and revokes the corresponding keys to provide data security. In an embodiment, in case of the multi-edge scenario, degree of the multi-edge is reduced, and the child node is not considered compromised.

In an embodiment, the revocation system 101 and the database 102 may communicate via the communication network 103, for revoking the compromised cryptographic key from the database 102. The communication network 103 may include, without limitation, a direct interconnection, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, and the like. In an embodiment, the revocation system 101 may be implemented in a server configured to revoke the compromised cryptographic key from the database 102. In an embodiment, such server may be a dedicated server or a cloud-based server.

Further, the revocation system 101 may include one or more processor 104, I/O interface 105, and a memory 106. In some embodiments, the memory 106 may be communicatively coupled to the one or more processors 104. The memory 106 stores instructions, executable by the one or more processors 104, which, on execution, may cause the revocation system 101 to revoke the compromised cryptographic key from the database 102, as disclosed in the present disclosure. In an embodiment, the memory 106 may include one or more modules 107 and data 108. The one or more modules 107 may be configured to perform the steps of the present disclosure using the data 108, to revoke the compromised cryptographic key from the database 102. In an embodiment, each of the one or more modules 107 may be a hardware unit which may be present outside the memory 106 and coupled with the revocation system 101. The revocation system 101 may be implemented in a variety of computing systems, such as a laptop computer, a desktop computer, a Personal Computer (PC), a notebook, a smartphone, a tablet, e-book readers, a server, a network server, a cloud-based server and the like. In an embodiment, the revocation system 101 may be a dedicated server or may be a cloud-based server.

Figure 3 illustrates flow diagram showing method for revoking compromised cryptographic key from the database 102, in accordance with some embodiments of the present disclosure.

At block 301 of **Figure 3**, the revocation system 101 may be configured to retrieve the key identifier of the compromised cryptographic key from the database 102.

At block 302 of **Figure 3**, the revocation system 101 may be configured to trace the dependencies of the key identifier in the graph by iterating over child nodes. In an embodiment, the child nodes may be the nodes dependent on the key identifier.

At block 303 of **Figure 3**, the revocation system 101 revokes the compromised cryptographic key and the corresponding keys from the database 102 and provides data security.

Advantages of the present disclosure

Embodiments of the present disclosure discloses a system and method to efficiently revoke compromised cryptographic key from a database by storing dependency information in the database.

Embodiments of the present disclosure provides data security by providing a database which stores the identifier of the cryptographic key and not the cryptographic key itself.

Computing System

Figure 4 illustrates a block diagram of an exemplary computer system 400 for implementing embodiments consistent with the present disclosure. In an embodiment, the computer system 400 is used to implement the revocation system 101 for revoking compromised cryptographic key from database. The computer system 400 may include a central processing unit (“CPU” or “processor”) 402. The processor 402 may include at least one data processor for executing processes in Virtual Storage Area Network. The processor 402 may include specialized processing units such as, integrated system (bus) controllers, memory management control units, floating point units, graphics processing units, digital signal processing units, etc.

The processor 402 may be disposed in communication with one or more input/output (I/O) devices 409 and 410 via I/O interface 401. The I/O interface 401 may employ communication protocols/methods such as, without limitation, audio, analog, digital, monaural, RCA, stereo, IEEE-1394, serial bus, universal serial bus (USB), infrared, PS/2, BNC, coaxial, component, composite, digital visual interface (DVI), high-definition multimedia interface (HDMI), radio frequency (RF) antennas, S-Video, VGA, IEEE 802.n /b/g/n/x, Bluetooth, cellular (e.g., code-division multiple access (CDMA), high-speed packet access (HSPA+), global system for mobile communications (GSM), long-term evolution (LTE), WiMax, or the like), etc.

Using the I/O interface 401, the computer system 400 may communicate with one or more I/O devices 409 and 410. For example, the input devices 409 may be an antenna, keyboard, mouse, joystick, (infrared) remote control, camera, card reader, fax machine, dongle, biometric reader, microphone, touch screen, touchpad, trackball, stylus, scanner, storage device, transceiver, video device/source, etc. The output devices 410 may be a printer, fax machine, video display (e.g., cathode ray tube (CRT), liquid crystal display (LCD), light-emitting diode (LED), plasma, Plasma Display Panel (PDP), Organic light-emitting diode display (OLED) or the like), audio speaker, etc.

In some embodiments, the computer system 400 may consist of the revocation system 101. The processor 402 may be disposed in communication with a communication network 411 via a network interface 403. The network interface 403 may communicate with the communication network 411. The network interface 403 may employ connection protocols including, without limitation, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc. The communication network 411 may include, without limitation, a direct interconnection, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, etc. Using the network interface 403 and the communication network 411, the computer system 400 may communicate with a database 412 to complete any revocation of compromised cryptographic key from the database 412. The network interface 403 may employ connection protocols include, but not limited to, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc.

The communication network 411 includes, but is not limited to, a direct interconnection, an e-commerce network, a peer to peer (P2P) network, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, Wi-Fi, and such. The first network and the second network may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), etc., to communicate with each other. Further, the first network and the second network may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, etc.

In some embodiments, the processor 402 may be disposed in communication with a memory 405 (e.g., RAM, ROM, etc. not shown in **Figure 4**) via a storage interface 404. The storage interface 404 may connect to memory 405 including, without limitation, memory drives, removable disc drives, etc., employing connection protocols such as, serial advanced technology attachment (SATA), Integrated Drive Electronics (IDE), IEEE-1394, Universal Serial Bus (USB), fibre channel, Small Computer Systems Interface (SCSI), etc. The memory drives may further include a drum, magnetic disc drive, magneto-optical drive, optical drive, Redundant Array of Independent Discs (RAID), solid-state memory devices, solid-state drives, etc.

The memory 405 may store a collection of program or database components, including, without limitation, user interface 406, an operating system 407, web browser 408 etc. In some embodiments, computer system 400 may store user/application data, such as, the data, variables, records, etc., as described in this disclosure. Such databases may be implemented as fault-tolerant, relational, scalable, secure databases such as Oracle ® or Sybase®.

The operating system 407 may facilitate resource management and operation of the computer system 400. Examples of operating systems include, without limitation, APPLE MACINTOSH® OS X, UNIX®, UNIX-like system distributions (E.G., BERKELEY SOFTWARE DISTRIBUTION™ (BSD), FREEBSD™, NETBSD™, OPENBSD™, etc.), LINUX DISTRIBUTIONS™ (E.G., RED HAT™, UBUNTU™, KUBUNTU™, etc.), IBM™ OS/2, MICROSOFT™ WINDOWS™ (XP™, VISTA™/7/8, 10 etc.), APPLE® IOS™, GOOGLE® ANDROID™, BLACKBERRY® OS, or the like.

In some embodiments, the computer system 400 may implement a web browser 408 stored program component. The web browser 408 may be a hypertext viewing application, such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari, etc. Secure web browsing may be provided using Hypertext Transport Protocol Secure (HTTPS), Secure Sockets Layer (SSL), Transport Layer Security (TLS), etc. Web browsers 408 may utilize facilities such as AJAX, DHTML, Adobe Flash, JavaScript, Java, Application Programming Interfaces (APIs), etc. In some embodiments, the computer system 400 may implement a mail server stored program component. The mail server may be an Internet mail server such as Microsoft Exchange, or the like. The mail server may utilize facilities such as ASP, ActiveX, ANSI C++/C#, Microsoft .NET, Common Gateway Interface (CGI) scripts, Java, JavaScript, PERL, PHP, Python, WebObjects, etc. The mail server may utilize communication protocols such as Internet Message Access Protocol (IMAP), Messaging Application Programming Interface (MAPI), Microsoft Exchange, Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), or the like. In some embodiments, the computer system 400 may implement a mail client stored program component. The mail client may be a mail viewing application, such as Apple Mail, Microsoft Entourage, Microsoft Outlook, Mozilla Thunderbird, etc.

Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term "computer-readable medium" should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, Compact Disc (CD) ROMs, DVDs, flash drives, disks, and any other known physical storage media.
media.

The described operations may be implemented as a method, system or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The described operations may be implemented as code maintained in a "non-transitory computer readable medium", where a processor may read and execute the code from the computer readable medium. The processor is at least one of a microprocessor and a processor capable of processing and executing the queries. A non-transitory computer readable medium may include media such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, DVDs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, Flash Memory, firmware, programmable logic, etc.), etc. Further, non-transitory computer-readable media may include all computer-readable media except for a transitory. The code implementing the described operations may further be implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.).

The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments. Also, the words "comprising," "having," "containing," and "including," and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items or meant to be limited to only the listed item or items. It must also be noted that as used herein and in the appended claims, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise.

Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., are non-transitory. Examples include random access memory (RAM), read-only memory (ROM), volatile memory, non-volatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the embodiments of the disclosure is intended to be illustrative, but not limiting, of the scope of the disclosure.

With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

SYSTEM AND METHOD FOR REVOKING COMPROMISED CRYPTOGRAPHIC KEY FROM DATABASE

ABSTRACT

The present disclosure provides a system and a method for revoking compromised cryptographic key from a database. The proposed system provides a database which is a graph that stores information on dependencies between a working key and cryptographic algorithms and keys used in establishing the working keys. The proposed system upon identifying the compromised keys revokes the compromised cryptographic keys and corresponding keys from the database. The proposed system provides data security and an efficient revocation process of the compromised cryptographic keys.

Figure 3

1/4

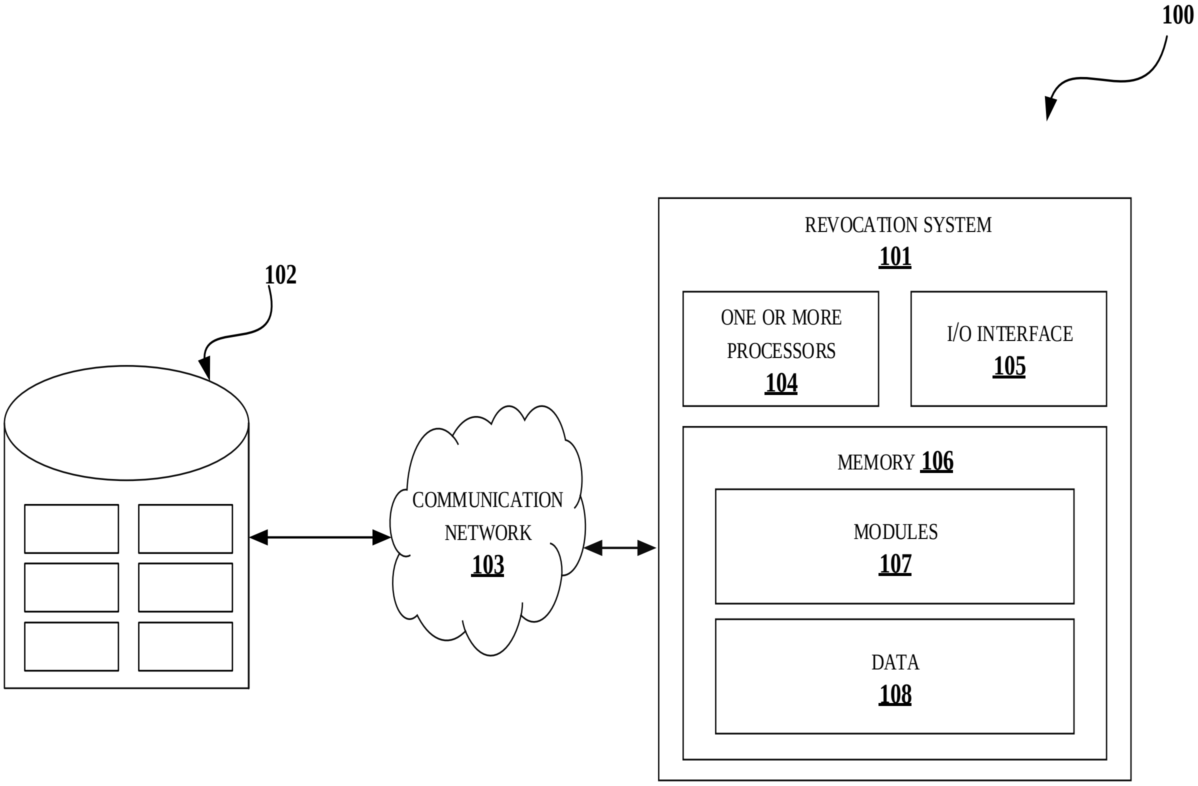
100
102
COMMUNICATION NETWORK
103
REVOCATION SYSTEM
101
ONE OR MORE PROCESSORS
104
I/O INTERFACE
105
MEMORY 106
MODULES
107
DATA
108

Figure 1

2/4

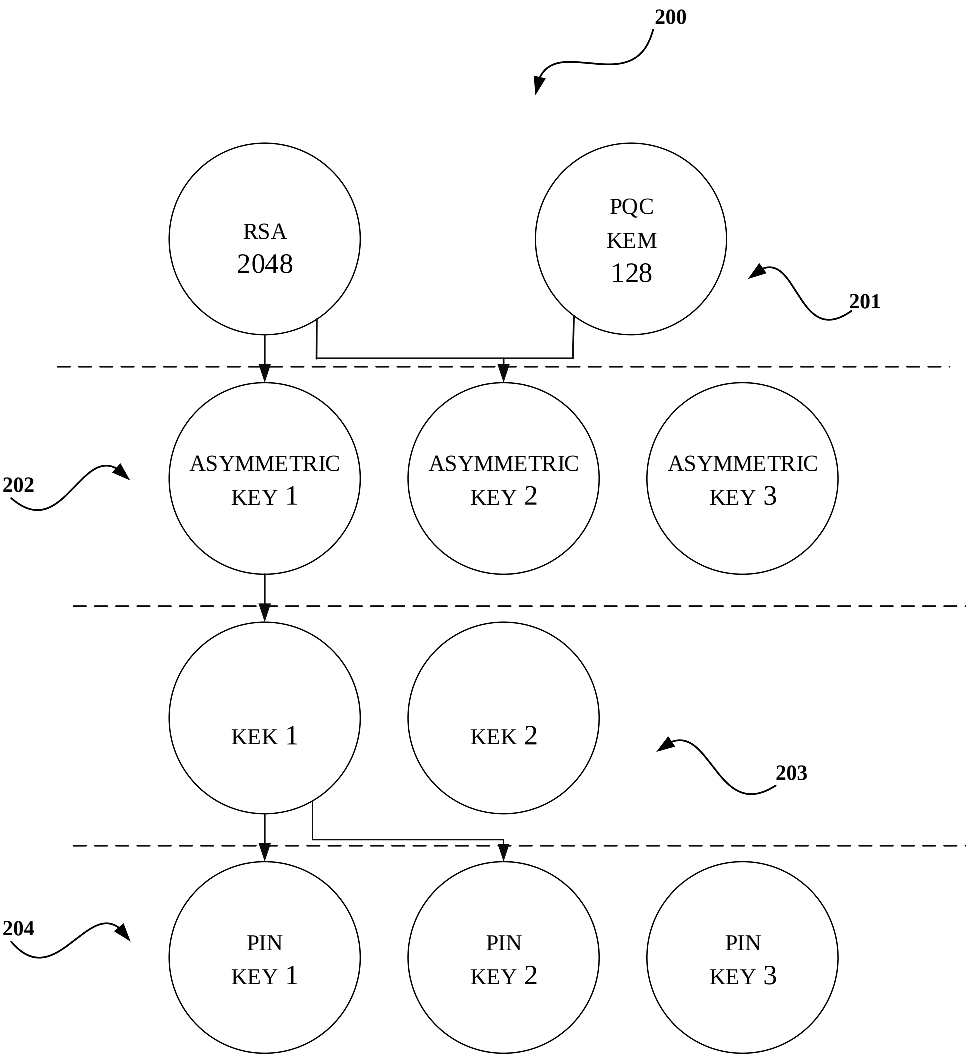
200
RSA
2048
PQC
KEM
128
201
202
ASYMMETRIC
KEY 1
ASYMMETRIC
KEY 2
ASYMMETRIC
KEY 3
KEK 1
KEK 2
203
204
PIN
KEY 1
PIN
KEY 2
PIN
KEY 3

Figure 2

3/4

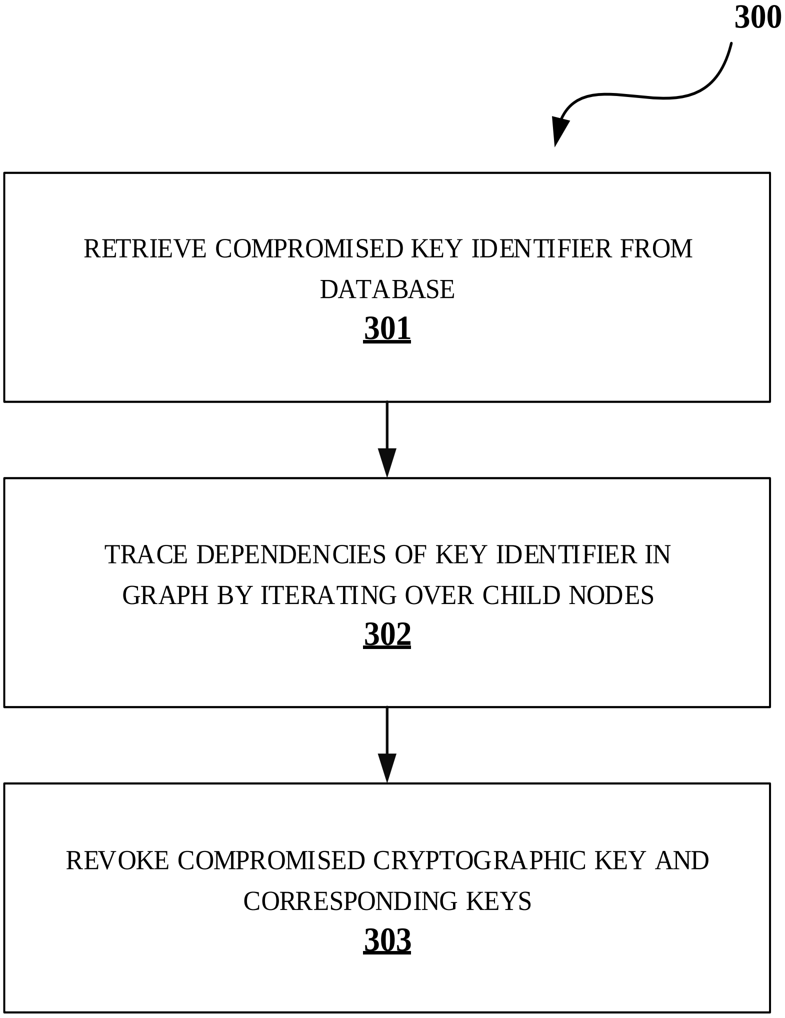
300
RETRIEVE COMPROMISED KEY IDENTIFIER FROM DATABASE
301
TRACE DEPENDENCIES OF KEY IDENTIFIER IN GRAPH BY ITERATING OVER CHILD NODES
302
REVOKE COMPROMISED CRYPTOGRAPHIC KEY AND CORRESPONDING KEYS
303

Figure 3

4/4

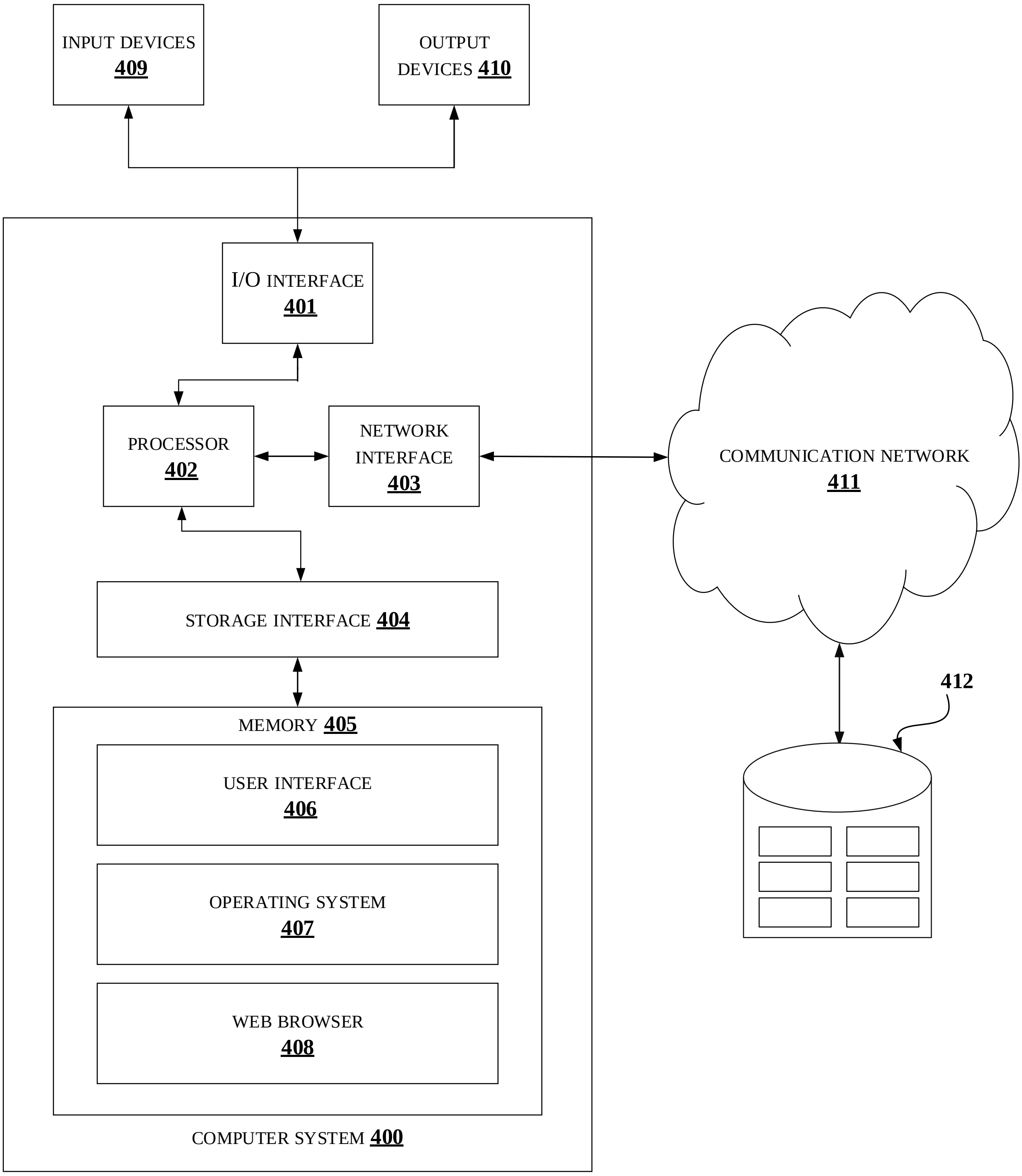
INPUT DEVICES 409
OUTPUT DEVICES 410
I/O INTERFACE 401
PROCESSOR 402
NETWORK INTERFACE 403
COMMUNICATION NETWORK 411
STORAGE INTERFACE 404
412
MEMORY 405
USER INTERFACE 406
OPERATING SYSTEM 407
WEB BROWSER 408
COMPUTER SYSTEM 400

Figure 4