

COMPUTING KEY ROTATION PERIOD FOR BLOCK CIPHER-BASED ENCRYPTION SCHEMES SYSTEM AND METHOD

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to US Provisional Application No. 62/751,068 filed on October 26, 2018, the contents of which are incorporated by reference in their entirety for all purposes.

BACKGROUND

[0002] Formal security analyses are used in modern cryptography. These security analyses often relate an adversary's available resources to determine if the adversary will be successful in breaking a particular cryptographic scheme. However, traditional techniques that are used to analyze the ability of an adversary to break a cryptographic scheme are often based upon the computation power (running time) of the adversary. While this can be effective in some instances, computational power may not be an accurate representation of an adversary's ability to corrupt a secure system. As a result, cryptographic keys may be replaced more often than necessary, which can waste computer resources.

[0003] Embodiments of the invention address these and other problems individually and collectively.

BRIEF SUMMARY

[0004] One embodiment is related to a method comprising: obtaining, by a key management computer, a key rotation period based on at least an adversarial storage limit; generating, by the key management computer, a first cryptographic key; and generating, by the key management computer, a second cryptographic key to replace the first cryptographic key according to the key rotation period.

[0005] Another embodiment is related to a key management computer comprising: a processor; and a computer-readable medium coupled to the processor, the computer-readable medium comprising code executable by the processor for implementing a method comprising: obtaining a key rotation period

based on at least an adversarial storage limit; generating a first cryptographic key; and generating a second cryptographic key to replace the first cryptographic key according to the key rotation period.

5 **[0006]** Further details regarding embodiments of the invention can be found in the Detailed Description and the Figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1 shows a block diagram of a system according to embodiments.

[0008] FIG. 2 shows a block diagram of a key management computer according to embodiments.

10 **[0009]** FIG. 3 shows a block diagram illustrating a process of determining a key rotation period according to embodiments.

[0010] FIG. 4 shows a flow diagram illustrating a process for key rotation according to embodiments.

DETAILED DESCRIPTION

15 **[0011]** Prior to discussing embodiments, some terms can be described in further detail.

[0012] An “adversarial storage limit” can include a data storage limit of an adversary. An adversarial storage limit can be an amount of data which an adversary (e.g., an adversarial computer, device, etc.) may store. In some
20 embodiments, an adversarial storage limit may be a limit of memory. In some embodiments, a memory limit may be discussed in terms of fast random-access memory that can be used for fast computation. As an illustrative example, the Amazon EC2 P3 may have a storage limit of 2^{40} Bytes of memory (e.g., approximately 1.5 TBytes).

25 **[0013]** A “key” or “cryptographic key” may include a piece of information that is used in a cryptographic algorithm to transform input data into another representation. A cryptographic algorithm can be an encryption algorithm that transforms original data into an alternate representation, or a decryption algorithm that transforms

encrypted information back to the original data. Examples of cryptographic algorithms may include triple data encryption standard (TDES), data encryption standard (DES), advanced encryption standard (AES), etc.

5 **[0014]** A “key identifier” may include any combination of letters, numbers, and/or symbols used to uniquely identify at least one key. A key identifier may be assigned randomly, consecutively, or according to any pattern or algorithm. In some embodiments, a key identifier can be derived from a key.

10 **[0015]** A “block cipher” can include a deterministic process operating on groups of bits, called blocks, with an unvarying transformation that is specified by a cryptographic key. In some embodiments, a secure block cipher can be suitable for the encryption of a single block under a fixed cryptographic key. Subsequent cryptographic keys can be created to perform further encryptions after the first cryptographic key has expired. In some embodiments, a multitude of modes of operation can be employed to allow repeated use of a cryptographic key in a secure way to allow for confidentiality and authenticity of any encrypted data.

15 **[0016]** A “block size” can include an amount of bits included in a block. Block ciphers can operate on a fixed length string of bits, where the length of the string of bits can be referred to as the block size. A block size can include any suitable size (e.g., a block size may be 64 bits, 128 bits, 192 bits, 256 bits, etc.).

20 **[0017]** A “security value” can include a value indicative of a security level. In some embodiments, the security value may be referred to as a security margin. In some embodiments, a security value can be of any suitable value which can result in an associated cryptographic process being secure. For example, the security value can be equal to a value of 2^{-10} , or any other value.

25 **[0018]** A “key rotation period” can include repeating interval at which a first cryptographic key may be rotated for a second cryptographic key. A key rotation period may include an amount of time a cryptographic key may be used before being replaced by another cryptographic key. For example, a key rotation period may be a time of 1 day for a first cryptographic key. After 1 day of use of the first
30 cryptographic key, a key management computer can replace the first cryptographic key with a second cryptographic key. In some embodiments, a key rotation period

need not be time based. For example, a key rotation period may specify an encryption data limit for the first cryptographic key. For example, the key rotation period may be an encryption data limit of 2^{70} bits. After a key management computer uses a first cryptographic key to encrypt 2^{70} bits of data, the key management computer can begin utilizing a second cryptographic key. As another example, a key rotation period may specify a cipher call limit for the first cryptographic key. For example, the key rotation period may specify a cipher call limit for the first cryptographic key equal to, for example, 2^{63} calls. After the cipher calls have been made, then a key management computer can rotate to the next (e.g., second) cryptographic key.

[0019] A "plaintext" may include text that is in a plain form. For example, this may include text that is readable by a human or a computer without any processing, such as the phrase "hello, how are you?" In some embodiments, a plaintext may also include a text that is in an unencrypted form. Numbers or other symbols may also qualify as plaintext.

[0020] A "ciphertext" may include text that is in an encrypted form. For example, this could refer to text that can be decrypted before it is readable by a human or computer. Ciphertext may be generated by any number of cryptographic algorithms or cryptosystems, such as RSA (Rivest–Shamir–Adleman), AES (Advanced Encryption Standard), etc.

[0021] A "server computer" may include a powerful computer or cluster of computers. For example, the server computer can be a large mainframe, a minicomputer cluster, or a group of servers functioning as a unit. In one example, the server computer may be a database server coupled to a Web server. The server computer may be coupled to a database and may include any hardware, software, other logic, or combination of the preceding for servicing the requests from one or more client computers. The server computer may comprise one or more computational apparatuses and may use any of a variety of computing structures, arrangements, and compilations for servicing the requests from one or more client computers.

[0022] A "processor" may include any suitable data computation device or devices. A processor may comprise one or more microprocessors working together

to accomplish a desired function. The processor may include a CPU comprising at least one high-speed data processor adequate to execute program components for executing user and/or system-generated requests. The CPU may be a microprocessor such as AMD's Athlon, Duron and/or Opteron; IBM and/or Motorola's PowerPC; IBM's and Sony's Cell processor; Intel's Celeron, Itanium, Pentium, Xeon, and/or XScale; and/or the like processor(s).

[0023] A "memory" may include any suitable device or devices that can store electronic data. A suitable memory may comprise a non-transitory computer readable medium that stores instructions that can be executed by a processor to implement a desired method. Examples of memories may comprise one or more memory chips, disk drives, etc. Such memories may operate using any suitable electrical, optical, and/or magnetic mode of operation.

I. INTRODUCTION

[0024] Traditional techniques that are used to analyze the ability of an adversary to break a cryptographic scheme is often based upon the computation power (running time) of the adversary. While this can be effective in some instances, computational power may not be an accurate representation of an adversary's ability to corrupt a secure system. As a result, cryptographic keys may be replaced more often than necessary, which can waste computer resources.

[0025] Embodiments allow for a determination of a key rotation period for rotating a cryptographic key. The key rotation period can be based on, at least, an adversarial storage limit. Determining a key rotation period based upon an adversarial storage limit can be a more accurate way of determining an appropriate key rotation period.

[0026] The adversarial storage limit can be obtained and/or determined in any suitable manner. For example, in some embodiments, the adversarial storage limit can be obtained, by the key management computer, from a security team operating the key management computer. In other embodiments, the adversarial storage limit can be determined by obtaining an initial adversarial storage limit and then determining a current adversarial storage limit using Moore's law, or other suitable mechanism for determining an estimated increase in technical development over

time. For example, a determined adversarial storage limit may be increased by the key management computer by a predetermined amount, relative to a previously determined adversarial storage limit, every 6 months.

5 [0027] After obtaining the key rotation period, the key management computer can generate a first cryptographic key. In some embodiments, the key management computer can store the first cryptographic key in a cryptographic key database, which may be accessed by one or more key recipient computers. In other embodiments, the key management computer can provide the first cryptographic key to a key recipient computer.

10 [0028] After generating the first cryptographic key, the key management computer can generate a second cryptographic key to replace the first cryptographic key according to the key rotation period. For example, if the key rotation period is equal to 5 calls of a corresponding block cipher (e.g., 5 uses of the block cipher), then a second cryptographic key can be used to replace the first cryptographic key
15 once the requirements outlined by the key rotation period are fulfilled (e.g., after 5 calls have been performed with the first cryptographic key).

II. SYSTEMS

[0029] Embodiments can include any suitable system as described herein. In some embodiments, a suitable system may be capable of performing methods as
20 described herein. One such system is illustrated in FIG. 1 and described in further detail below.

A. Network Overview

[0030] FIG. 1 shows a system 100 comprising a number of components according to embodiments. The system 100 comprises key management computer
25 102, a first key recipient computer 104, and an nth key recipient computer 106. The system 100 further comprises an adversary computer 108. The key management computer 102 can be in operative communication with any suitable number of other computers. For example, the key management computer 102 can be in operative communication with the first key recipient computer 104 and the nth key recipient
30 computer 106. Any of key management computer 102 and/or first key recipient

computer 104 to nth key recipient computer 106 can act as a server computer, a client computer, or both.

[0031] For simplicity of illustration, a certain number of components are shown in FIG. 1. It is understood, however, that some embodiments may include more than one of each component.

[0032] The computers in the system 100 may be in operative communication with each other through any suitable communication channel or communications network. Suitable communications networks may be any one and/or the combination of the following: a direct interconnection; the Internet; a Local Area Network (LAN); a Metropolitan Area Network (MAN); an Operating Missions as Nodes on the Internet (OMNI); a secured custom connection; a Wide Area Network (WAN); a wireless network (e.g., employing protocols such as, but not limited to a Wireless Application Protocol (WAP), I-mode, and/or the like); and/or the like. Messages between the computers, networks, and devices may be transmitted using a secure communications protocols such as, but not limited to, File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure Hypertext Transfer Protocol (HTTPS), Secure Socket Layer (SSL), ISO (e.g., ISO 8583) and/or the like.

[0033] The key management computer 102 can generate, manage, and/or transmit and receive symmetric, and/or public-private cryptographic keys. For example, the key management computer 102 can generate and rotate cryptographic keys for its own use to encrypt data stored by the key management computer 102 and/or decrypt data accessed by the key management computer 102. In some embodiments, the key management computer 102 can generate and rotate cryptographic keys that are used to encrypt and decrypt communications between the key management computer 102 and any of the first key recipient computer 104 to the nth key recipient computer 106. In some embodiments, the key management computer 102 can generate and rotate cryptographic keys that the key management computer 102 itself does not use, but are provisioned to the first key recipient computer 104 to the nth key recipient computer 106. The cryptographic keys provisioned by the key management computer 102 can be used by the first key recipient computer 104 to the nth key recipient computer 106 to encrypt and decrypt communications amongst themselves, or to encrypt and decrypt storage data.

[0034] In some embodiments, the key management computer 102 can obtain a key rotation period based on at least an adversarial storage limit. The key management computer 102 can then generate a first cryptographic key. After generating the first cryptographic key, the key management computer 102 can
5 generate a second cryptographic key to replace the first cryptographic key according to the key rotation period.

[0035] A key recipient computer (e.g., the first key recipient computer 104 to the nth key recipient computer 106) can be capable of receiving the first cryptographic key and/or the second cryptographic key from the key management
10 computer 102. In some embodiments, the key recipient computer can be capable of generating a cryptographic key request message. The cryptographic key request message may be a request message comprising a request for one or more cryptographic keys. The key recipient computer can then provide the cryptographic key request message to the key management computer 102. Upon receiving the
15 cryptographic key request message, the key management computer 102 can determine whether or not to provide the one or more cryptographic keys to the key recipient computer. The key management computer 102 can then generate and provide a cryptographic key response message to the key recipient computer. The key recipient computer can then encrypt any suitable data with the one or more
20 cryptographic keys received in the cryptographic key response message.

[0036] The adversary computer 108 can include any suitable computer, device, etc. operated by an adversarial entity. The adversary computer 108 can be in operative communication with the Nth key recipient computer 106. The adversary computer 108 may attempt to perform malicious actions with the Nth key recipient
25 computer 106. In some embodiments, the adversary computer 108 can, for example, attempt to corrupt a cryptographic scheme employed by the Nth key recipient computer 106.

B. Key Management Computer

[0037] FIG. 2 shows a block diagram of a key management computer 200
30 according to embodiments. The exemplary key management computer 200 may comprise a processor 204. The processor 204 may be coupled to a memory 202, a

network interface 206, and a computer readable medium 208. The computer readable medium 208 can comprise an adversarial storage limit determination module 208A, a key rotation period determination module 208B, and a cryptographic key generation module 208C. The key management computer 200 may be in
5 operative communication with a cryptographic key database 220.

[0038] The memory 202 can be used to store data and code. The memory 202 may be coupled to the processor 204 internally or externally (e.g., cloud based data storage), and may comprise any combination of volatile and/or non-volatile memory, such as RAM, DRAM, ROM, flash, or any other suitable memory device.

10 For example, the memory 202 can store cryptographic keys, and values associated with various parameters such as, but not limited to, a block cipher block size, a security values, an adversarial storage limit (e.g., in bits), a key rotation period (e.g., a data limit for a cryptographic key), etc.

[0039] The computer readable medium 208 may comprise code, executable
15 by the processor 204, for performing a method comprising: obtaining, by a key management computer, a key rotation period based on at least an adversarial storage limit; generating, by the key management computer, a first cryptographic key; and generating, by the key management computer, a second cryptographic key to replace the first cryptographic key according to the key rotation period.

20 **[0040]** The adversarial storage limit determination module 208A may comprise code or software, executable by the processor 204, for determining a storage limit of an adversary (e.g., an adversary computer). The adversarial storage limit determination module 208A, in conjunction with the processor 204, can obtain the adversarial storage limit (e.g., as input) from a security team operating the key
25 management computer 200. For example, the security team can analyze commercially available computers, devices, components, etc. The security team can determine an average adversarial storage limit, a high end value of the adversarial storage limit, and/or any other suitable adversarial storage limit based on the analysis of the commercially available computers. In other embodiments, the
30 adversarial storage limit determination module 208A, in conjunction with the processor 204, can periodically increase the adversarial storage limit starting from a predetermined value. The adversarial storage limit determination module 208A, in

conjunction with the processor 204, can periodically increase the adversarial storage limit based on Moore's law, or other suitable estimated time-increase in memory capabilities. For example, the adversarial storage limit determination module 208A, in conjunction with the processor 204, can periodically increase the adversarial storage limit by a predetermined amount (e.g., adding a predetermined amount, doubling the adversarial storage limit, etc.) at a period of 3 months.

[0041] The key rotation period determination module 208B can include may comprise code or software, executable by the processor 204, for determining a cryptographic key rotation period. The key rotation period determination module 208B, in conjunction with the processor 204, can determine a cryptographic key rotation period based on at least an adversarial storage limit. In some embodiments, the key rotation period determination module 208B, in conjunction with the processor 204, can further determine the key rotation period based on the security value and the block cipher block size, in addition to the adversary storage limit.

[0042] For example, in some embodiments, the key rotation period determination module 208B, in conjunction with the processor 204, can determine the key rotation period q equal to:

$$q = \frac{\varepsilon * 2^n}{n * m} \quad (A)$$

where ε can be a security value, n can be a block size of a corresponding block cipher, and m can be an adversarial storage limit. The security value ε can include a value indicative of a security level. The security value ε can be of any suitable value which can result in an associated cryptographic process being secure. For example, the security value ε can be equal to a value of 2^{-10} (which may be unitless), or any other value. In some cases, the security value can be characterized as a "security margin," which as is known in the art, is a measure for how much better a system needs to get at analyzing a cipher to break it. A security margin can be selected by a security expert, based upon a number of factors including the type of systems used, etc.

[0043] The block size n can include an amount of bits included in a block of a block cipher. Block ciphers can operate on a fixed length string of bits, where the length of the string of bits can be referred to as the block size n . The block size n

can include any suitable size (e.g., the block size n may be 64 bits, 128 bits, 192 bits, 256 bits, etc.). The adversarial storage limit m can include a data storage limit of an adversary. The adversarial storage limit m can be an amount of data which the adversary (e.g., an adversarial computer, device, etc.) may store. As an illustrative example, the commercially available Amazon EC2 P3 sold by Amazon may have a storage limit m of 2^{40} Bytes of memory (e.g., approximately 1.5 TBytes or 1.2×10^{13} bits). In some embodiments, the units of 2^n may be considered as bits^2 , since there may be 2^n bits^2 of the block cipher.

[0044] The cryptographic key generation module 208C can include may comprise code or software, executable by the processor 204, for generating cryptographic keys. The cryptographic key generation module 208C, in conjunction with the processor 204, can generate any suitable cryptographic keys. In some embodiments, the cryptographic key generation module 208C, in conjunction with the processor 204, can generate symmetric cryptographic keys. A symmetric cryptographic key can be a cryptographic key that is used for symmetric encryption. For example, a symmetric key may be used for both encryption of plain text and decryption of cipher text. A transmitting device may encrypt a message using the symmetric key and a receiving device may decrypt the encrypted message using the same symmetric key. The symmetric key, in practice, represents a shared secret between two or more computing devices that can be used to maintain private information. In some embodiments, the cryptographic key generation module 208C, in conjunction with the processor 204, can generate cryptographic keys based on AES, RSA, or any other suitable cryptosystem as known to one of skill in the art.

[0045] The network interface 206 may include an interface that can allow the key management computer 200 to communicate with external computers. The network interface 206 may enable the key management computer 200 to communicate data to and from another device (e.g., resource provider computers, authorization computers, key recipient computers, etc.). Some examples of the network interface 206 may include a modem, a physical network interface (such as an Ethernet card or other Network Interface Card (NIC)), a virtual network interface, a communications port, a Personal Computer Memory Card International Association (PCMCIA) slot and card, or the like. The wireless protocols enabled by the network

interface 206 may include Wi-Fi™. Data transferred via the network interface 206 may be in the form of signals which may be electrical, electromagnetic, optical, or any other signal capable of being received by the external communications interface (collectively referred to as “electronic signals” or “electronic messages”). These
5 electronic messages that may comprise data or instructions may be provided between the network interface 206 and other devices via a communications path or channel. As noted above, any suitable communication path or channel may be used such as, for instance, a wire or cable, fiber optics, a telephone line, a cellular link, a radio frequency (RF) link, a WAN or LAN network, the Internet, or any other suitable
10 medium.

[0046] In some embodiments, the key management computer 200 can be operatively coupled to the cryptographic key database 220. The cryptographic key database 220 may securely store data, such as cryptographic keys. The
cryptographic key database 220 may be a conventional, fault tolerant, relational,
15 scalable, secure database such as those commercially available from Oracle™ or Sybase™.

III. METHODS

[0047] Embodiments can use the systems and apparatuses described above to securely obtain a key rotation period based on at least an adversarial storage limit.
20 FIGS. 3-4 describe some examples of such methods. In some embodiments, the key management computer may include the key management computer 102 or 200 of FIGS. 1 and 2, respectively. The key recipient computer may include the first key recipient computer 104 or the Nth key recipient computer 106 of FIG. 1. Additional details and explanation of the key rotation technique will now be described.

A. Block Ciphers

[0048] A block cipher can encrypt a block of plaintext as a whole and can produces a ciphertext block of equal length. For example, a block cipher can take a fixed-length block of text of length b bits and a key as an input and produce a b-bit block of ciphertext. In some embodiments, a block size of 64 or 128 bits can be
30 used.

[0049] A multitude of modes of operation can be employed to allow repeated use of a cryptographic key in a secure way. For example, there may be one mode, two modes, three modes, four modes, five modes, etc. In some embodiments, the situation in which five modes are utilized by a key management computer can allow
 5 for a wide variety of applications of encryption for which a block cipher could be used. The five modes of operation can include 1) electronic codebook (ECB), 2) cipher block chaining (CBC), 3) cipher feedback (CFB), 4) output feedback (OFB), and 5) counter (CTR).

[0050] Electronic codebook (ECB) can be considered the simplest mode, in
 10 which plaintext can be handled one block at a time. Each block of plaintext can be encrypted using the same key. The term codebook is used because, for a given cryptographic key, there may be a unique cipher text for every b-bit block of plaintext. As an illustrative example, the codebook may be a large group of bits in which there is an entry for every possible b-bit plaintext showing its corresponding
 15 ciphertext. For a message longer than b bits, the key management computer can break the message into b-bit blocks, encrypting the blocks as appropriate, and then in some embodiments, padding the last block.

[0051] In some embodiments, decryption may be performed one block at a time using the same cryptographic key. For example, a plaintext can consist of a
 20 sequence of b-bit blocks ($P_1, P_2, \dots, P_n$) the corresponding sequence of ciphertext blocks can be denoted as $C_1, C_2, \dots, C_n$.

[0052] In some embodiments, ECB can be defined as follows, where E can be an encryption function, D can be a decryption function, and K can be a cryptographic key,

$$25 \quad C_i = E(K, P_i) \quad i = 1, \dots, N \quad (B)$$

$$P_i = D(K, C_i) \quad i = 1, \dots, N \quad (C)$$

[0053] The ECB mode can be useful, for example, for a short amount of data such as an encryption key. A characteristic of ECB is that if the same b-bit block of plaintext appears more than once in the message, it always produces the same
 30 ciphertext.

B. Security Analysis

[0054] A security analysis can relate an attacker's success advantage in breaking a cryptographic protocol to the attacker's available resources. Current techniques in symmetric cryptography deal with a running time of an adversary.

- 5 However, memory (in terms of fast random-access memory that can be used for fast computation) can be order of magnitude more constrained than computation. For example, an Amazon EC2 P3 instance (e.g., a commercially available cloud service sold by Amazon) has 2^{50} FLOPS and 2^{40} bytes of memory.

Component	Amount
Computation (FLOPS)	1.0 Pflops $\approx 2^{50}$
Memory (bytes)	1.5 Tbytes $\approx 2^{40}$

Table 1: Capacity of Amazon EC2 P3 instance

- 10 **[0055]** Block ciphers are one of the most widely used cryptographic primitives. A block cipher can usually be assumed to behave as a random function. In an analysis, this can be captured by a PRP/PRF switching lemma (i.e., pseudo random permutation (PRP) and pseudo random function (PRF)), which can state the following: A can be a random-access machine with m -bits of memory making q
- 15 oracle queries. Let F_n and P_n be a uniform random function and permutation on n -bit strings, respectively. Then,

$$Pr[A^{F_n} \Rightarrow 1] - Pr[A^{P_n} \Rightarrow 1] \leq \frac{q(q+1)}{2^{n+1}} \quad (D)$$

- [0056]** Techniques described herein provide a bound for the left-hand-side (of the above equation) in terms of the size of memory, m , of the random access
- 20 machine A . A framework for analyzing memory-restricted adversaries can thus be developed and a clean positive result for PRP/PRF switching, in terms of time-memory trade-off can be provided.

- [0057]** A word random-access-machine (word RAM) model of computation can closely resemble modern computers. As such, symmetric cryptographic
- 25 primitives can be analyzed in the ideal model, where adversaries can be

computationally unbounded and the “running time” of the adversary is captured by the number of queries from ideal objects. As a result, the following adversary model can be derived.

[0058] An oracled random-access machine (RAM), A , having access to oracle $O : \{0,1\}^n \rightarrow \{0,1\}^n$, and with memory bound m is a finite sequence of instructions, $f_0, f_1, \dots, f_q$, where for each i , f_i is a function of the form $f_i : \{0,1\}^m \rightarrow \{0,1\}^m$.

[0059] Fixing an oracle $O : \{0,1\}^n \rightarrow \{0,1\}^n$, an execution of $A = f_0, f_1, \dots, f_q$ can consist of a sequence of memory content $M_{0,0}, M_{1,0}, \dots, M_{q,0}, M_{q,1}$ which can be derived as follows.

$$O^m = M_{0,1} \xrightarrow{f_0} M_{0,1} \xrightarrow{O} M_{1,0} \xrightarrow{f_1} M_{1,1} \xrightarrow{O} \dots \xrightarrow{O} M_{q,0} \xrightarrow{f_q} M_{q,1} \quad (E)$$

[0060] By convention, it can be the case that $m \geq n$. Memory $M_{i,1}$ can be derived from $M_{i,0}$ by applying f_i . Memory $M_{i,0}$ can be derived from $M_{i-1,1}$ by applying O to the last n -bit of $M_{i-1,1}$. Finally, the last memory content, $M_{q,1}$, can be the output of A^O (denoted $A^O \Rightarrow M_{q,1}$).

[0061] The above can yield the following theorem 1:

$$\varepsilon = \Pr[A^{F_n} \Rightarrow 1] - \Pr[A^{P_n} \Rightarrow 1] \leq \frac{q \cdot n \cdot m}{2^n} \quad (F)$$

[0062] The quantity being bounded (e.g., a security value ε) in Theorem 1 (e.g., Equation F) can be a quantity used in block ciphers, such as AES and 3DES, for encryption. In these applications, q can correspond to a number of block cipher calls of the encryption scheme. In some embodiments, the right hand side of the above theorem 1 may be further divided by n bits of the block size to determine q' , which can correspond to the data limit of the encryption scheme, representing the maximum amount of data to be encrypted under a single key (e.g., until a new key is needed). The technique described herein allows a key management computer to relax the bound on the data limit when an appropriate bound on available memory of potential attackers is known.

[0063] FIG. 3 shows a block diagram illustrating a process of determining a key rotation period according to embodiments. FIG. 3 includes cryptographic

parameters 302, risk analysis parameters 304, a model 306, and a key rotation period 308.

[0064] The cryptographic parameters 302 can include parameters for the model 306 which may relate to the cryptographic scheme utilized by the key management computer. For example, the cryptographic parameters 302 can include a block size n (e.g., in bits). The risk analysis parameters 304 can include parameters for the model 306 which may be based on a performed risk analysis and may depend upon the environment which the key management computer operates within. For example, the risk analysis parameters 304 can include a security value ϵ (e.g., unitless) and an adversarial storage limit m (e.g., in bits). The cryptographic parameters 302 and the risk analysis parameters 304 can be input into the model 306 by the key management computer.

[0065] The model 306 can include a way in which the key management computer can obtain the key rotation period 308. For example, the model 306 can include an equation relating the cryptographic parameters 302 (e.g., the block size n) and the risk analysis parameters 304 (e.g., the security value ϵ and the adversarial storage limit m) to the key rotation period 308 (e.g., the key rotation period q). The key rotation period can include a value which can represent when to replace a first cryptographic key with a second cryptographic key (e.g., rotate the keys). The key rotation period q (e.g., number of calls, or in units of bits when the key rotation period is q') of FIG. 3 can specify a number of calls to a block cipher with the associated cryptographic key before rotating to a subsequent cryptographic key.

[0066] In some embodiments, the model 306 can further include an additional instance of the parameter or the block size n when determining the key rotation period q' . In this case, the key rotation period 308 can include a key rotation period q' (e.g., in bits) which can specify an amount of data encrypted by the cryptographic key before rotating to a subsequent cryptographic key. As an illustrative example, to determine the key rotation period q' , the model may be $q' = \frac{\epsilon * 2^n}{m}$.

[0067] In some embodiments, the model can include additional terms if a more specific mode of block cipher operation is chosen beyond ECB mode as described herein. For example, other modes may result in a model 306 in which an

additional term may be added to the term of, for example, $\frac{\varepsilon \cdot 2^n}{m}$. The additional term may directly depend from the mode of operation selected.

C. Key Rotation Process

[0068] FIG. 4 shows a flow diagram illustrating a process 400 for key rotation, according to embodiments. The method illustrated in FIG. 4 will be described in the context of a key management computer obtaining a key rotation period and generating cryptographic keys which may be replaced based on the key rotation period. It is understood, however, that the invention can be applied to other circumstances regarding key rotation periods obtained based on at least an adversary storage limit. The process 400 of FIG. 4 may be performed by a key management computer (e.g., the key management computer 200 of FIG. 2).

[0069] Prior to step 402, the key management computer can obtain an adversarial storage limit m . The key management computer can obtain the adversarial storage limit m in any suitable manner, as described herein. In some embodiments, the adversarial storage limit m can be received as an input to process 400, or can be determined, for example, based on an assessment of the storage limit of available system components (e.g., available to an adversarial entity operating an adversary computer).

[0070] For example, the key management computer can obtain the adversarial storage limit m (e.g., as input) from a security team operating the key management computer. The key management computer can receive, as input, an adversarial storage limit m with a value of 1.5 TBytes. In some embodiments, the adversarial storage limit m may be a value less than 1.5 TBytes (e.g., 1 TBytes). In other embodiments, the adversarial storage limit m may be a value greater than 1.5 TBytes (e.g., 2 TBytes). For example, a larger value of an adversarial storage limit m may allow for greater security since an adversary may not be able (e.g., it may be more difficult) to break the encryption. Whereas, a smaller value of an adversarial storage limit m may allow for less security, but also a longer key rotation period, thus fewer cryptographic keys may be generated.

[0071] In other embodiments, the key management computer can periodically increase the adversarial storage limit m starting from a predetermined value. The key management computer can periodically increase the adversarial storage limit m based on, for example, Moore's law, or other suitable estimated time-increase in memory capabilities. For example, the key management computer can periodically increase the adversarial storage limit m by a predetermined amount (e.g., adding a predetermined amount, doubling the adversarial storage limit m , etc.) at a period of 2 months. For example, the key management computer can increase the value of the adversarial storage limit m from a starting value of 1.2 TBytes by a value of 0.05 TBytes, 0.1 TBytes, 0.5 TBytes, 0.75 TBytes, 1.0 TByte after a period of 1 month, 2 months, 6 months, 1 year, 2 years, etc. For example, in some embodiments, the adversarial storage limit m may be 1 petabyte or any other suitable value.

[0072] At step 402, the key management computer can obtain a key rotation period based on at least the adversarial storage limit m . For example, the key management computer can retrieve the adversarial storage limit m from a memory and can then obtain the key rotation period based on the adversarial storage limit m according to the process illustrated in FIG. 3. In other embodiments, the key management computer can further determine the key rotation period based on the security value ε and the block cipher block size, in addition to the adversary storage limit.

[0073] As an illustrative example, the key management computer can obtain an adversarial storage limit m of 2^{40} bits from memory. The key management computer can also obtain a value of 2^{-10} for the security value ε and a value of 128 for the block size n .

[0074] In some embodiments, the key rotation period can specify an encryption data limit for the first cryptographic key. In other embodiments, the key rotation period can specify a cipher call limit for the first cryptographic key.

[0075] In some embodiments, after obtaining the key rotation period, the key management computer can store the key rotation period in the memory.

[0076] At step 404, the key management computer can generate a first cryptographic key. The first cryptographic key can be used for data encryption

and/or secured communications. For example, the cryptographic key may be a symmetric cryptographic key and may be generated according to block cipher cryptographic techniques.

5 **[0077]** In some embodiments, after generating the first cryptographic key, the key management computer can provide the first cryptographic key to a key recipient computer. The key recipient computer can utilize the first cryptographic key in any suitable manner. For example, the key recipient computer can encrypt data with the first cryptographic key.

10 **[0078]** In other embodiments, after generating the first cryptographic key, the key management computer can store the first cryptographic key in a cryptographic key database, which may be accessible by a plurality of key recipient computers.

[0079] In yet other embodiments, after generating the first cryptographic key, the key management computer can encrypt data with the first cryptographic key.

15 **[0080]** At step 406, the key management computer can generate a second cryptographic key to replace the first cryptographic key according to the determined key rotation period. For example, the second cryptographic key can be generated at, or before, the requirements indicated by the key rotation period have been met (e.g., an amount of bits have been encrypted, a number of block cipher calls have been made, etc.) since the generation and/or use of the first cryptographic key.

20 **[0081]** In some embodiments, the key management computer can receive a cryptographic key request message from the key recipient computer. The key management computer can then provide a cryptographic key response message comprising the second cryptographic key to the key recipient computer. The key recipient computer can then, for example, encrypt additional data with the second
25 cryptographic key.

[0082] In other embodiments, the key management computer can store the generated cryptographic keys into a cryptographic key database. A key recipient computer can retrieve the first cryptographic key and/or the second cryptographic key from the cryptographic key database at any suitable time.

[0083] Next, a comparative analysis between a previous best known analysis of determining a key rotation period for a cryptographic protocol and embodiments of the discloser is discussed. Embodiments provide for an improved key rotation period. As an illustrative example, the following Table 2 and Table 3 illustrate preliminary results and analysis of the data limit q in terms of transport layer security (TLS) (e.g., a cryptographic protocol which provides communication security over a computer network). The previously used TLS 1.3 is shown below:

$$\varepsilon = Adv_A^{prf} \leq \frac{q(q-1)}{2^{n+1}} \quad (G)$$

TLS 1.3 (with previously best known analysis)	
Security (Adv) (e.g., ε)	2^{-10}
Memory bound (m)	N/A
Key rotation period(q)	$2^{59.5}$

Table 2: Key rotation period of previous methods

[0084] The following Table 3 shows TLS 1.3 with embodiments to determine the key rotation period q' .

$$\varepsilon = Adv_A^{prf} \leq \frac{q' \cdot m}{2^n} \quad (H)$$

TLS 1.3 (with embodiments)	
Security (Adv) (e.g., ε)	2^{-10}
Memory bound (m)	2^{40}
Key rotation period (q')	2^{70}

Table 3: Key rotation period of embodiments

[0085] Thus, embodiments allow for a larger data limit for the key rotation period q' (e.g., a value of 2^{70} bits rather than $2^{59.5}$ bits) before rotating cryptographic

keys. Therefore, a given cryptographic key can be more efficient and able to encrypt more data before expiring.

[0086] The key management computer can determine a key rotation period, which may be based on a data limit, such as an encryption data limit. The key management computer can determine the key rotation period based on a block size of the block cipher in use and an adversarial storage limit m . In some embodiments, the key rotation period can also be based on a security value ε . For example, as shown in FIG. 3, the key management server can determine the key rotation period by computing $q' = \varepsilon * 2^n / m$, where q is the key rotation period in bits, n is the block size in bits, ε is the security value ε , and m is the adversarial storage limit m in bits.

[0087] In some embodiments, the key management computer can generate cryptographic keys, as described herein, that can have a key rotation period based on at least an adversarial storage limit m . The key management server can issue and/or provision the cryptographic keys to other computers. For example, the key management server can provision a symmetric key to a first key recipient computer and a second key recipient computer. The first key recipient computer and the second key recipient computer can then establish a secure communication channel with one another using the symmetric keys received from the key management server. In some embodiments, the key management computer can transmit the data limit (e.g., key rotation period) for the cryptographic key to the key recipient computers. The key recipient computers can then track the amount of data encrypted with the cryptographic key and can be capable of requesting a new cryptographic key as and/or when the amount of data encrypted approaches and/or reaches the data limit.

[0088] Embodiments of the invention have a number of advantages. For example, key rotation periods determined according to embodiments can allow for more data to be encrypted before the associated cryptographic key expires. In this way, the key management server can save computational power as it may not need to create cryptographic keys as often. For example, in reference to the above transport layer security (TLS) example, embodiments provide for a larger key rotation period than in the previously best known analysis. Thus, a key management

computer may not need to generate as many cryptographic keys as was done with the previously best known methods and systems.

[0089] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C, C++, C#, Objective-C, Swift, or scripting language such as Perl or Python using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer readable medium for storage and/or transmission, suitable media include random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a compact disk (CD) or DVD (digital versatile disk), flash memory, and the like. The computer readable medium may be any combination of such storage or transmission devices.

[0090] Such programs may also be encoded and transmitted using carrier signals adapted for transmission via wired, optical, and/or wireless networks conforming to a variety of protocols, including the Internet. As such, a computer readable medium according to an embodiment of the present invention may be created using a data signal encoded with such programs. Computer readable media encoded with the program code may be packaged with a compatible device or provided separately from other devices (e.g., via Internet download). Any such computer readable medium may reside on or within a single computer product (e.g., a hard drive, a CD, or an entire computer system), and may be present on or within different computer products within a system or network. A computer system may include a monitor, printer, or other suitable display for providing any of the results mentioned herein to a user.

[0091] The above description is illustrative and is not restrictive. Many variations of the invention will become apparent to those skilled in the art upon review of the disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the pending claims along with their full scope or equivalents.

[0092] One or more features from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the invention.

[0093] As used herein, the use of "a," "an," or "the" is intended to mean "at least one," unless specifically indicated to the contrary.

5

WHAT IS CLAIMED IS:

1 1. A method comprising:
2 obtaining, by a key management computer, a key rotation period based
3 on at least an adversarial storage limit;
4 generating, by the key management computer, a first cryptographic
5 key; and
6 generating, by the key management computer, a second cryptographic
7 key to replace the first cryptographic key according to the key rotation period,
8 wherein the key rotation period is determined according to the following
9 formula:

$$q = \frac{\varepsilon * 2^n}{n * m}$$

11 where q is the key rotation period, ε is a security value, n is a block
12 size of a corresponding block cipher, and m is the adversarial storage limit.

1 2. The method of claim 1 further comprising:
2 providing, by the key management computer, the first cryptographic
3 key to a key recipient computer, wherein the key recipient computer encrypts data
4 with the first cryptographic key.

1 3. The method of claim 2 further comprising:
2 receiving, by the key management computer, a cryptographic key
3 request message from the key recipient computer; and
4 providing, by the key management computer, a cryptographic key
5 response message comprising the second cryptographic key to the key recipient
6 computer, wherein the key recipient computer encrypts additional data with the
7 second cryptographic key.

1 4. The method of claim 1, wherein the key rotation period specifies
2 an encryption data limit for the first cryptographic key.

1 5. The method of claim 1, wherein the key rotation period specifies
2 a cipher call limit for the first cryptographic key.

1 6. The method of claim 1, wherein obtaining the key rotation period
2 based on at least the adversarial storage limit further comprises:

3 obtaining, by the key management computer, the key rotation period
4 based on at least the adversarial storage limit from a memory of the key
5 management computer.

1 7. The method of claim 6 further comprising:
2 obtaining, by the key management computer, the adversarial storage
3 limit;
4 determining, by the key management computer, the key rotation period
5 based on at least the adversarial storage limit; and
6 storing, by the key management computer, the key rotation period in
7 the memory.

1 8. The method of claim 1 further comprising:
2 storing, by the key management computer, the first cryptographic key
3 and the second cryptographic key in a cryptographic key database, wherein a key
4 recipient computer retrieves the first cryptographic key and/or the second
5 cryptographic key from the cryptographic key database.

1 9. The method of claim 1 further comprising:
2 updating, by the key management computer, the adversarial storage
3 limit after a predetermined amount of time.

1 10. A key management computer comprising:
2 a processor; and
3 a computer-readable medium coupled to the processor, the computer-
4 readable medium comprising code executable by the processor for implementing a
5 method comprising:

6 obtaining a key rotation period based on at least an adversarial
7 storage limit;

8 generating a first cryptographic key; and

9 generating a second cryptographic key to replace the first
10 cryptographic key according to the key rotation period,

wherein the key rotation period is determined according to the following formula:

$$q = \frac{\varepsilon * 2^n}{n * m}$$

where q is the key rotation period, ε is a security value, n is a block size of a corresponding block cipher, and m is the adversarial storage limit.

11. The key management computer of claim 10 further comprising:
an adversarial storage limit determination module;
a key rotation period determination module; and
a cryptographic key generation module.

12. The key management computer of claim 10 further comprising:
a memory coupled to the processor, and wherein obtaining the key rotation period based on at least the adversarial storage limit further comprises:
obtaining the key rotation period from the memory.

13. The key management computer of claim 12, wherein the method further comprises:
obtaining the adversarial storage limit;
determining the key rotation period based on at least the adversarial storage limit; and
storing the key rotation period in the memory.

14. The key management computer of claim 10, wherein the method further comprises:
updating the adversarial storage limit after a predetermined amount of time.

15. The key management computer of claim 10, wherein the key rotation period specifies an encryption data limit for the first cryptographic key.

16. The key management computer of claim 10, wherein the key rotation period specifies a cipher call limit for the first cryptographic key.

1 17. The key management computer of claim 10, wherein the first
2 cryptographic key is a first symmetric key and wherein the second cryptographic key
3 is a second symmetric key.

1 18. The key management computer of claim 10, wherein the method
2 further comprises:

3 storing the first cryptographic key and the second cryptographic key in
4 a cryptographic key database.

1 19. A key management computer comprising:

2 a processor; and

3 a computer-readable medium coupled to the processor, the computer-
4 readable medium comprising code executable by the processor for implementing a
5 method comprising:

6 obtaining a key rotation period based on at least an adversarial
7 storage limit;

8 generating a first cryptographic key; and

9 generating a second cryptographic key to replace the first
10 cryptographic key according to the key rotation period,

11 wherein the key rotation period is determined according to the following
12 formula:

$$q' = \frac{\varepsilon * 2^n}{m}$$

14 where q' is the key rotation period, ε is a security value, n is a
15 block size of a corresponding block cipher, and m is the adversarial storage
16 limit.