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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2018 (18.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/011825 A1

(51) International Patent Classification:
G06F 21/60 (2013.01) *H04L 9/28* (2006.01)
H04L 9/00 (2006.01)

(21) International Application Number:
PCT/IN2017/050289

(22) International Filing Date:
11 July 2017 (11.07.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201641023730 11 July 2016 (11.07.2016) IN

(71) Applicant: **INDIAN INSTITUTE OF SCIENCE** [IN/IN]; Office of Intellectual Property and Technology Licensing, Karnataka, Bangalore 560 012 (IN).

(72) Inventors: **SRIKANTH, Cherukupally**; Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore 560012 (IN). **VENI MADHAVAN, Conjeevaram Ekambaram**; Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore 560012 (IN). **KUMAR SWAMY, Hebbar Vadiraj**; Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore 560012 (IN).

(74) Agent: **LAKSHMIKUMARAN, Malathi** et al.; B6/10, Safdarjung Enclave, New Delhi 110029 (IN).

(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.

(54) Title: ENCRYPTION AND DECRYPTION OF MESSAGES

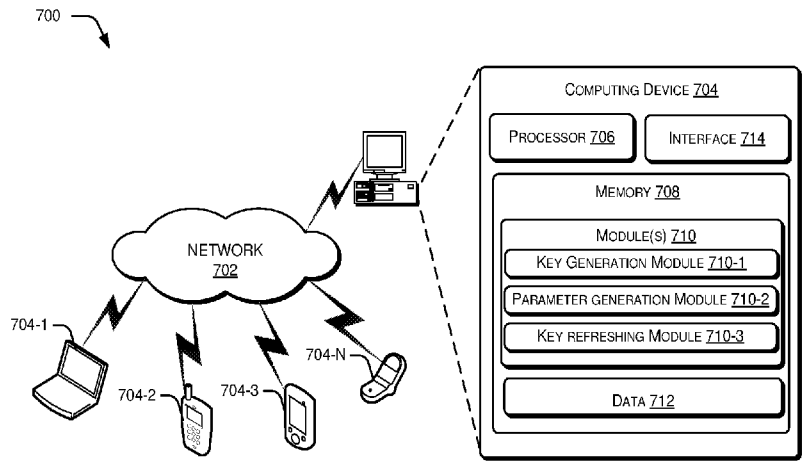

Fig. 7

(57) Abstract: A method and system for encrypting and decrypting messages, and user authentication is described. The method includes obtaining a message or cipher-text, one or more predetermined numbers, a predetermined key (e), and private keys (a0, d0) by a processor (706) of a computing device (704) and generating one or more sequences of arithmetic progressions based on first set of properties and the private keys (a0, d0). The processor (706) determines a master sequence based on the predetermined key (e) and determines sub-collections of arithmetic progression in the master sequence based on second set of properties. Further, one or more master arithmetic progressions are determined in each of the sub-collections based on the one or more predetermined numbers. The processor (706) then selects the leading terms and the common difference of the master arithmetic progressions of each master arithmetic progression and determines the master private keys for encrypting and decrypting.

WO 2018/011825 A1

[Continued on next page]

(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).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

ENCRYPTION AND DECRYPTION OF MESSAGES

TECHNICAL FIELD

[0001] The present subject matter, in general, relates to cryptography and, in particular, relates to encryption and decryption of messages.

BACKGROUND

[0002] Cryptography is a method used for secure transmission of data between two parties. Cryptography involves encryption of data in one form, such as a plaintext, to another form, such as a cipher-text, at a sender's end and decryption of the cipher-text to the plaintext, at a receiver's end. Cryptographic systems can be classified into two types, symmetric cryptography and asymmetric cryptography based on a key used for encryption and decryption of the data.

[0003] Symmetric or private key cryptography involves the same key for encryption and decryption of data. Symmetric key encryption can use either stream ciphers or block ciphers. Stream ciphers encrypt the digits (typically bytes) of a message one at a time and block ciphers take a number of bits and encrypt them as a single unit. Examples of computational models used for symmetric key cryptography include Rivest Cipher 4 (RC4), Advanced Encryption Standard (AES), and Common Scrambling Algorithm (CSA).

[0004] Asymmetric or public key cryptography involves two keys, a public key and a secret key or private key for encrypting and decrypting data, respectively. The secret key and the public key are, typically, mathematically linked and obtained using complex mathematical computations to ensure safety of the encrypted data. The public key is used for encryption of the plaintext, whereas the secret key is used for decryption of the cipher-text. Examples of computational models used for public key cryptography include Rivest Shamir Adleman (RSA), Diffie-Hellman key exchange (DH), and Elliptic curve cryptography (ECC).

[0005] Further, a cryptographic system involving a combination of the symmetric and asymmetric cryptography, known as hybrid cryptosystems, is also used in cryptography. For example, encryption of data may be performed using symmetric key and the symmetric key can be encrypted using the public key and sent to the receiver. The hybrid cryptosystems combine the advantages of public key cryptography and private key cryptography.

BRIEF DESCRIPTION OF DRAWINGS

[0006] The detailed description is described with reference to the accompanying figures. 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.

[0007] Fig. 1 illustrates a method for performing symmetric encryption, according to an embodiment of the present subject matter.

[0008] Fig. 2 illustrates a method for performing symmetric decryption, according to an embodiment of the present subject matter.

[0009] Fig. 3 illustrates a method for performing hybrid encryption, according to an embodiment of the present subject matter.

[0010] Fig. 4 illustrates a method for performing hybrid decryption, according to an embodiment of the present subject matter.

[0011] Fig. 5 illustrates a method for user authentication, in accordance with an embodiment of the present subject matter.

[0012] Fig. 6(a) and Fig. 6(b) illustrate a first and second computing device in a computing network, in accordance with an embodiment of the present subject matter.

[0013] Fig. 7 illustrates a computing device in a computing network, according to an embodiment of the present subject matter.

DETAILED DESCRIPTION

[0014] The present subject matter relates to cryptographic systems and methods for encryption and decryption of messages.

[0015] Generally, computational models associated with cryptography are intensive as they involve complex mathematical operations, such as arithmetic operations over large integers and modular reduction over these large integers. The computational models may also involve many inversion operations over large integers. Further, different cryptographic systems involve different computational models for ensuring security of transmitted data.

[0016] For example, the security of asymmetric cryptographic systems is based on computational complexity of fundamental problems like integer factoring and discrete logarithms. On the other hand, symmetric cryptographic systems, particularly block ciphers, are based on general principles of substitution and permutation transformations, and in the case of stream ciphers, feed-back shift registers are used. Further, both asymmetric cryptographic systems and symmetric cryptographic systems with different computational models are also used in conjunction.

[0017] Therefore, asymmetric cryptographic systems and symmetric cryptographic systems are implemented using different principles and computational models. Consequently, the characteristics, such as performance and strengths of asymmetric cryptographic and symmetric cryptographic systems are studied and analyzed on separate frameworks. Due to this, invulnerability to cryptanalytic attacks and reliability of cryptographic systems may not be easily determined.

[0018] In accordance with the present subject matter, cryptographic methods and systems using similar computational models have been provided.

[0019] The present subject matter includes a method for providing a secure communication of messages transmitted over a computing network comprising a plurality of computing devices. The computing network may include at least a first computing device (a sender) and a second computing device (a receiver) and each of

the plurality of computing devices in the computing network comprises at least a processor and a memory. The term "processor" may hereinafter be used for collectively referring to the processors of the first computing device and the second computing device, unless otherwise specified.

[0020] In one embodiment, the processor of the first computing device obtains a pair of private keys, one or more predetermined numbers, a predetermined key, and a message to be transmitted to the second computing device. The pair of private keys may be a pair of numbers based on which the processor can generate one or more sequences of arithmetic progressions.

[0021] In one embodiment, the processor of the first computing device retrieves the pair of private keys from the memory of the first computing device. For example, the pair of private keys, the predetermined key, and the one or more predetermined numbers may be agreed between the first computing device and the second computing device and may be stored in the memory of the first computing device and the second computing device. In another embodiment, the pair of private keys may be determined by the processor of the first computing device based on certain parameters, as will be discussed later.

[0022] The pair of private keys may be any pair of numbers which may be used for generating one or more sequences of arithmetic progressions. The processor generates the one or more sequences of arithmetic progressions based on the pair of private keys and a first set of properties. For example, one of the numbers in the pair of private keys may be used as a leading term and the other may be used as a common difference of a first arithmetic progression in the generated sequences. The number of sequences generated may be equivalent to the greatest common divisor of the pair of private keys.

[0023] In one embodiment, the first set of properties may include:- the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor

of the pair of private keys; the common difference of the arithmetic progressions are in non-increasing order; the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression; the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function. The predetermined function may be based on the leading terms and common differences of the first arithmetic progression.

[0024] Further, a master sequence is determined from the one or more sequences based on the predetermined key, by the processor. The processor determines sub-collections of the arithmetic progressions in the master sequence based on a second set of properties. The second set of properties may include:- (i) the difference between the common differences of consecutive arithmetic progressions, hereinafter referred to as a "second common difference", in the sub-collection is equal; (ii) each of the sub-collection is maximal, i.e., includes all arithmetic progressions from the sequence satisfying the property (i).

[0025] Further, one or more master arithmetic progressions are determined from each sub-collection. The master arithmetic progressions are arithmetic progressions that are selected from each sub-collection, based on the one or more predetermined numbers, by the processor. For example, the one or more predetermined numbers may indicate a position of the master arithmetic progressions in each sub-collection. From each sub-collection, the leading terms and the common differences of the one or more master arithmetic progressions may be selected by the processor accordingly. The selected leading terms and the common differences may be used for determining master private keys, which may in turn be used for encryption or decryption.

[0026] The master private keys are determined by the processor based on at least the leading terms and the common differences of the one or more master

arithmetic progressions. For example, the master private keys may be one of the terms of the one or more master arithmetic progressions. In one embodiment, the master private keys may be determined based on the leading terms, common differences, and the second common difference of the sub-collections.

[0027] Further, at the first computing device, the master private keys may be used for encrypting the messages into a cipher-text. In one embodiment, at the first computing device, the message may be encoded into a number M and then split into smaller numbers (m_1, m_2, ..., m_i). Each of the numbers (m_1, m_2, ..., m_i) may be encrypted into the cipher-text using the master private keys corresponding to each m_t. The cipher-text may be transmitted to the second computing device in the network.

[0028] In one embodiment, when the messages are very large, the pair of private keys may be refreshed for generating multiple sequences of arithmetic progressions. For example, the common difference of the first arithmetic progression may be kept constant and the leading term of the first arithmetic progression may be computed based on a key refreshing model.

[0029] Further, the processor of the second computing device obtains the cipher-text, the pair of private keys, the predetermined key, and the one or more predetermined numbers. The cipher-text is received from the first computing device. The processor may retrieve the pair of private keys, the predetermined key, and the one or more predetermined numbers from the memory. In one embodiment, the pair of private keys may be determined by the processor based on certain parameters.

[0030] On receiving the cipher-text, the processor of the second computing device decrypts the cipher-text into message using the master private keys. The master private keys may be determined by the processor of the second computing device in the same manner as determined by the processor of the first computing device. Further, on decrypting the cipher-text, a number M may be obtained which is decoded into the message transmitted by the first computing device.

[0031] Therefore, the present subject matter provides a secure communication of messages transmitted over a computing network using cryptographic methods described above. Further, the subject matter described can be extended to asymmetric and hybrid cryptosystems, and therefore, analysis of the different types of cryptosystems for precision, performance, strength, simplicity, etc., is performed on same framework. This enables a unified approach for determining the invulnerability to cryptanalytic attacks and reliability of cryptographic systems. Further, the refreshing of the pair of private keys enables encryption and decryption of messages of large sizes using the methods described.

[0032] These and other advantages of the present subject matter would be described in greater detail in conjunction with the following figures. While aspects of described encryption and decryption of messages can be implemented in any number of different computing systems, environments, and/or configurations, the embodiments are described in the context of the following exemplary system(s).

[0033] Fig. 1 illustrates a method 100 for performing symmetric encryption, in accordance with an embodiment of the present subject matter. The symmetric encryption may be performed in a computing network, where messages are transmitted between computing devices.

[0034] In one embodiment, the computing network may include at least a first computing device and a second computing device and each of the computing devices in the computing network comprises at least a memory and a processor. The first computing device may be a sender and the second computing device may be a receiver.

[0035] At block 102, the processor of the first computing device obtains a pair of private keys (a_0, d_0), a predetermined key (e), one or more predetermined numbers, and a message to be transmitted to the second computing device. The message to be transmitted to the second computing device may already be stored in

the memory of the first computing device, or an external memory connected to the first computing device, or may be provided by a user through an interface.

[0036] In one embodiment, the pair of private keys (a_0, d_0), the predetermined key (e), and the one or more predetermined numbers may be agreed between the first computing device and the second computing device. For example, the pair of private keys (a_0, d_0), the predetermined key (e), and the one or more predetermined numbers may be shared between the first computing device and the second computing device using a pre-shared key. The pre-shared key may be a shared secret, which may be previously shared between two computing devices using secure channels before it can be used. In another embodiment, the pair of private keys (a_0, d_0), the predetermined key (e), and the one or more predetermined numbers may be generated using a key management system.

[0037] The pair of private keys (a_0, d_0) may be a pair of numbers, where one of the pair of numbers may be a leading term and the other may be a common difference of an arithmetic progression. In one embodiment, the pair of private keys may be a pair of co-prime numbers.

[0038] In one embodiment, the pair of private keys (a_0, d_0) may be refreshed to a new pair of numbers based on a key refreshing model. The key refreshing model can be used for refreshing the value of the leading term using the equation as given below:

$$a_{0n} = (a_0)^z (\text{mod } d_0) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

[0039] where, a_{0n} may be a new private key, a_0 and d_0 refer to the pair of private keys, and z is an integer. In one embodiment, z is agreed between the first computing device and the second computing device. If z = 2, then the key refreshing model is the Blum-Blum-Shub method.

[0040] In another embodiment, the pair of private keys (a_0, d_0) may be refreshed using equation as given below:

$$a_{0n} = a_0 + z \ (\text{mod } d_0) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

[0041] In yet another embodiment, the pair of private keys (a_0, d_0) may be refreshed using equation as given below:

$$f = \Sigma_{i=1}^{t} LB_b(L_{i,1}) \pmod{2^b} \quad \text{.....(3)}$$

$$a_{0n} = (a_0)^f \pmod{d_0} \quad \text{.....(4)}$$

[0042] Here, $LB_b(x)$ is the lower order b bits of x for a small predetermined integer b, which is agreed upon by the first computing device and the second computing device. On computing f, the value of a_0 is refreshed to a new value using the equation (4).

[0043] Further, the processor generates one or more sequences of arithmetic progressions based on the pair of private keys. In one embodiment, as depicted at block 104, the one or more sequences of arithmetic progressions may be generated based on the pair of private keys (a_0, d_0) and a first set of properties. The number of sequences generated may be equivalent to the greatest common divisor of the pair of private keys. A master sequence is determined from the one or more sequences based on the predetermined key, by the processor.

[0044] In one embodiment, the first set of properties may include:- the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor of the pair of private keys; the common difference of the arithmetic progressions are in non-increasing order; the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression (represented in equation 5); the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function. The predetermined function may be based on the leading terms and common differences of the first arithmetic progression. In one embodiment, the predetermined function may be represented by an equation as given below:

$$d_{i+1} * a_i \equiv G \bmod (d_i) \ldots\ldots\ldots (5)$$

$$a_{i+1} = d_{i+1} + (a_i * d_{i+1} - G) / d_i \ldots . (6)$$

[0045] where, G is the greatest common divisor of the pair of private keys, a_i and d_i refer to the leading term and common difference, and i refers to the arithmetic progression in the sequence.

[0046] For example, if the pair of private keys is (11, 25), then the greatest common divisor G=1, which is the number of sequence generated. The sequence $\Phi(11, 25)$ of arithmetic progressions generated based on the first set of properties, may be as follows:

11, 36, 61, 86,...

23, 39, 55, 71,...

17, 24, 31, 38,...

17, 22, 27, 32,...

13, 16, 19, 22,...

5, 6, 7, 8,.....

[0047] The leading term and the common difference of second arithmetic progression in the sequence of arithmetic progressions may be determined by using the equation (5) and (6). The leading terms and the common differences of all other arithmetic progressions may be determined similarly.

[0048] Further, at block 106, sub-collections of the arithmetic progressions in the master sequence are determined based on a second set of properties. The second set of properties may include:- (i) the difference between the common differences of consecutive arithmetic progressions (second common difference) in the sub-collection is equal; (ii) each of the sub-collection is maximal, i.e., includes all arithmetic progressions from the sequence satisfying the property (i).

[0049] For example, the sub-collections of the sequence $\Phi(11, 25)$ determined based on the second set of properties may be as follows:

$$G_1 = <A(11, 25), A(23, 16), A(17, 7)>$$

$$G_2 = <A(17, 7), A(17, 5), A(13, 3), A(5, 1)>$$

[0050] where, G_1 and G_2 are sub-collections of the sequence $\Phi(11, 25)$. The second common difference of G_1 is 9 and the second common difference of G_2 is 2.

[0051] Further, one or more master arithmetic progressions are determined in each sub-collection based on the one or more predetermined numbers. In one embodiment, the one or more predetermined numbers may indicate position of the one or more master arithmetic progressions in each sub-collection of the sequence. For example, the predetermined number k=3 may indicate that the 3^{rd} arithmetic progression of each sub-collection would be the master arithmetic progression. In another embodiment, the one or more predetermined numbers may be denoted by k_{ab}, where k_{ab} represents the a^{th} arithmetic progression of the b^{th} sub-collection. For example, k_{27} may denote the 2^{nd} arithmetic progression in the 7^{th} sub-collection and k_{37} may denote the 3^{rd} arithmetic progression in the 7^{th} sub-collection.

[0052] At block 108, the leading terms and the common differences of the one or more master arithmetic progressions may be selected from each sub-collection. The selected leading terms and the common differences may be used for determining master private keys for encrypting the messages.

[0053] In one embodiment, the master private keys may be one of the terms from each of the one or more master arithmetic progressions for encrypting the messages. At block 110, the master private keys of the one or more master arithmetic progressions are determined based on at least the leading terms and the common differences of the one or more master arithmetic progressions. In one embodiment, a message may be encoded into a number M, which is split into smaller numbers (m_1, m_2, …, m_i), where $i \leq$ number of sub-collections (b). In another embodiment, the leading term, the common difference, and the second common difference may be used for determining the master private keys.

[0054] At block 112, the master private keys are used for encrypting the messages into a cipher-text. In one embodiment, the messages encoded into numbers (m_1, m_2, ..., m_i) may be encrypted into the cipher-text. The cipher-text may be transmitted to the second computing device in the computing network.

[0055] In a first embodiment, the message M is split into smaller numbers (m_1, m_2, ..., m_i) and encrypted into cipher-text using at least the leading terms and common difference. The number of binary bits of each smaller number may be approximately $\log_2 d_0$. For example, if i= number of sub-collections (b) and $1 \leq t \leq i$, the number m_t is encrypted into a cipher c_t, where $c_t = L_{t,k} + m_t * D_{t,k}$. Here, k refers to the predetermined number and $L_{t,k}$ and $D_{t,k}$ refer to the leading term and the common difference of the k^{th} arithmetic progression (master arithmetic progression) of t^{th} sub-collection. The cipher text C = (c_1, c_2, ..., c_i), is then transmitted to the second computing device.

[0056] In a second embodiment, the number m_t is encrypted into a cipher-text c_t, where $c_t \equiv m_t * L_{t,k} \pmod{D_{t,k}}$, for $1 \leq t \leq i$. The cipher c_t may be equal to $m_t * L_{t,k} + \alpha_t * D_{t,k}$, for some integer $\alpha_t < 0$. The cipher text C = (c_1, c_2, ..., c_i), is transmitted to the second computing device for decryption. In a third embodiment, the concatenation of the terms $L_{t,k} + r_t * D_{t,k}$, where $1 \leq t \leq i$, forms a key bit stream. Here, r_t is a small integer which is predetermined. The generated key bits may be XORed with the message bits formed by the number M to generate the cipher text C.

[0057] Further, in a fourth embodiment, the second common difference and more than one predetermined numbers may also be used for encryption. For example, two predetermined numbers (k_1, k_2) and the second common difference δ, may be used for encrypting into a cipher-text as

$$c_t = m_t * L_{t,k1} + D_{t,k2} + k_2 * \delta_t \quad \ldots\ldots\ldots\ldots (7)$$

[0058] Where $L_{t,k1}$ refers to the leading term of the k_1^{th} arithmetic progression in t^{th} sub-collection and $D_{t,k2}$ refers to the common difference of the k_2^{th} arithmetic

progression in t^{th} sub-collection. Further, the cipher-text, $C = (c_1, c_2, \ldots, c_i)$, may be transmitted to the second computing device for decryption.

[0059] Fig. 2 illustrates a method 200 for symmetric decryption according to an embodiment of the present subject matter. At block 202, the processor of the second computing device obtains the pair of private keys (a_0, d_0), the predetermined key (e), the one or more predetermined numbers, and the cipher-text.

[0060] The cipher-text may be received from the first computing device. In one embodiment, the pair of private keys (a_0, d_0) and the one or more predetermined numbers may be agreed between the first computing device and the second computing device. For example, the pair of private keys (a_0, d_0), the predetermined key (e), and the one or more predetermined numbers may be shared between the first computing device and the second computing device using pre-shared key. The pre-shared key may be a shared secret, which may be previously shared between two devices or parties using secure channels before it can be used. In another embodiment, the pair of private keys (a_0, d_0), the predetermined key (e), and the one or more predetermined numbers may be generated using the key management system.

[0061] In one embodiment, the pair of private keys (a_0, d_0) may be refreshed to a new pair of co-prime numbers based on a key refreshing model. The key refreshing model refreshes the value of the leading term using the equation (1), (2), or (4) as explained earlier.

[0062] At block 204, the processor of the second computing device generates the one or more sequences of arithmetic progressions based on the pair of private keys (a_0, d_0) and the first set of properties. The number of sequences generated is equivalent to the greatest common divisor of the pair of private keys (a_0, d_0). The one or more sequences of arithmetic progression may be generated in the same manner as generated by the processor of the first computing device. A master sequence is determined from the one or more sequences based on the predetermined key (e), by the processor.

[0063] At block 206, sub-collections of the arithmetic progressions in the master sequence are determined based on the second set of properties. The second set of properties may include:- (i) the difference between the common differences of consecutive arithmetic progressions in the sub-collection is equal; (ii) each of the sub-collection is maximal, i.e., includes all arithmetic progressions from the sequence satisfying the property (i).

[0064] Further, the one or more master arithmetic progressions may be determined based on the one or more predetermined numbers. For example, the one or more predetermined numbers may indicate position of the master arithmetic progressions in each sub-collection of the sequence. At block 208, the leading terms and the common differences of the one or more master arithmetic progression may be selected from each sub-collection.

[0065] The selected leading terms and the common differences may be used for determining the master private keys. The master private keys may be one of the terms from each of the one or more master arithmetic progressions. In one embodiment, the master private key may be determined based on at least the leading terms and the common differences of the one or more master arithmetic progressions, as depicted in block 210.

[0066] At block 212, the master private keys may be used for decrypting the cipher-text into the message. In all the embodiments, the cipher text (C) may be split into smaller ciphers (c_1, c_2, ..., c_i) before decryption. In the embodiments, on decrypting the cipher-text a number M may be obtained, which is decoded into the message transmitted by the first computing device.

[0067] In the first embodiment, the processor may receive cipher text C = (c_1, c_2, ..., c_i), from the first computing device, where $1 \leq t \leq i$. The processor may decrypt c_t using the leading term and common difference of the master arithmetic progression of each sub-collection. For example, c_t may be the m_t^{th} term of k_t^{th}

arithmetic progression (master arithmetic progression) of the t^{th} sub-collection in the master sequence. The processor decrypts c_t as $(c_t - L_{t,k}) / D_{t,k}$ to recover m_t.

[0068] In the second embodiment, the cipher text $C = (c_1, c_2, \ldots, c_i)$, received may have been determined by the first computing device as $c_t \equiv m_t * L_{t,k} \pmod{D_{t,k}}$. The message m_t may be recovered by computing $c_t * (L_{t,k})^{-1} \pmod{D_{t,k}}$. The value $(L_{t,k})^{-1} \pmod{D_{t,k}}$ may be computed by a parameter generation module and stored in the memory and accessed by the processor for decryption.

[0069] In the third embodiment, the processor of the second computing device decrypts the cipher text C by generating the same key bit stream, as generated by the first computing device under the third embodiment, which will be XORed with the cipher text C to recover the message M.

[0070] Further, in the fourth embodiment, the cipher text $C = (c_1, c_2, \ldots, c_i)$, may have been determined by the first computing device as $c_t = m_t * L_{t,k1} + D_{t,k2} + k_2 * \delta_t$. The cipher c_t may be decrypted by the processor as follows:

$$m_t = (c_t - (D_{t,k2} + k_2 * \delta_t)) / L_{t,k1} \ldots\ldots\ldots\ldots (8)$$

[0071] Where $L_{t,k1}$ refers to the leading term of the k_1^{th} arithmetic progression in t^{th} sub-collection and $D_{t,k2}$ refers to the common difference of the k_2^{th} arithmetic progression in t^{th} sub-collection.

[0072] Fig. 3 illustrates a method 300 for performing hybrid encryption in the computing network, according to an embodiment of the present subject matter. At block 302, the processor of the first computing device receives a public key (Δ), the predetermined key (e), and the one or more predetermined numbers. In one embodiment, the public key (Δ), the predetermined key (e), and the one or more predetermined numbers may be generated by a key management system based on a secret key pair (p, q) of the second computing device.

[0073] The key management system may be a computing device that can generate or reset keys, such as the pair of private keys, predetermined key, one or more predetermined numbers, secret keys, public keys, and the like. In one

embodiment, the key management system may be the second computing device that may generate the one or more predetermined numbers, the secret key pair (p, q), and its public key (Δ). The second computing device may then share the one or more predetermined numbers, the predetermined key (e), and the public key (Δ) to other computing devices in the computing network. In one embodiment, the secret key pair (p, q) may be a pair of prime numbers and the public key (Δ) may be a product of the prime numbers.

[0074] At block 304, on receiving the one or more predetermined numbers and the public key (Δ), the processor of first computing device selects a trap-door parameter (s), which is co-prime to the public key (Δ). In one embodiment, the trap-door parameter (s) may also be lesser than the public key (Δ) and greater than half of the public key (Δ).

[0075] At block 306, the processor determines the pair of private keys (a_0, d_0) and a trap-door information (v) using one or more of the one or more predetermined numbers, the public key (Δ), and the trap-door parameter (s). In one embodiment, the pair of private keys (a_0, d_0) may be determined based on the trap-door parameter (s), the public key (Δ), and the one or more predetermined numbers, and the trap-door information (v) may be calculated based on the public key (Δ) and the trap-door parameter (s).

[0076] For example, the pair of private keys (a_0, d_0) and the trap-door information (v) may be determined as follows:

$$d_0 = s + k*\Delta \quad \ldots\ldots(9)$$

$$a_0 \equiv - \Delta^{-1} \mathrm{mod}(d_0) \quad \ldots\ldots(10)$$

$$v = s - s^{-1} (\mathrm{mod}\ \Delta) \quad \ldots\ldots(11)$$

[0077] Where k may be one of the one or more predetermined numbers known to the first computing device and the second computing device. For example, k = 9, and the public key is as given below:

Δ=1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

[0078] The first computing device selects the trap-door parameter (s), which may be, for example,

s=1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

[0079] Based on the trap-door parameter (s) and the public key (Δ), the trap-door information (v) is determined using equation (11). The trap-door information may be given as

v=1212280430076947639827697457636793980653520623997874101784947348063026981930797816014755897873790260416711170302372568258414519156101494932843659542391145040410580119001028479636396923936383026785187491828170481306425830887541777883

[0080] Further, a_0 and d_0 may also be determined using equation (9) and (10) as follows:

a_0=659453193350359762527370398182314984794885234084248029940729784113994026778274024253012483532711122451883497976227528778388401353324627282855958077951410984568620427840365117141324717753206945732369087711643870789874004657342469254 3

d_0=1174452193507154842210355848057056951924236579225976772341847684741924445400702458222460585606560657575831500763938313258002869043252571791583319092779917715440464651200329909236078533026919195895122539742792866325415790246242985109 5

[0081] In one embodiment, the blocks 302, 304, and 306, may be used for key transfer from first computing device to second computing device. Further, the processor generates one or more sequences of arithmetic progressions based on the pair of private keys (a_0, d_0). In one embodiment, as depicted in block 308, the one or more sequences of arithmetic progressions may be generated based on the pair of private keys (a_0, d_0) and the first set of properties. The generation of the one or more sequences of arithmetic progressions may be performed in a manner as described with respect to Fig. 1. Further, a master sequence is determined by the processor based on the predetermined key (e).

[0082] At block 310, sub-collections of the arithmetic progressions in the master sequence are determined based on the second set of properties. The second set of properties may include:- (i) the difference between the common differences of consecutive arithmetic progressions in the sub-collection is equal; (ii) each of the sub-collection is maximal, i.e., includes all arithmetic progressions from the sequence satisfying the property (i).

[0083] Further, one or more master arithmetic progressions are determined based on the one or more predetermined numbers as described with respect to earlier figures. At block 312, the leading terms and the common differences of the one or more master arithmetic progression may be selected from each sub-collection. The selected leading terms and the common differences may be used for determining master private keys.

[0084] The master private keys may be one of the terms of the one or more master arithmetic progressions for encrypting the messages. The master private keys may be determined based on at least the leading terms and the common differences of the master arithmetic progressions, as depicted in block 314. In one embodiment, a message may be encoded into a number M, which is split into smaller numbers (m_1, m_2, …, m_i).

[0085] At block 316, the master private keys may be used for encrypting the messages into cipher-text. In one embodiment, the messages encoded into numbers (m_1, m_2, ..., m_i) may be encrypted into the cipher-text. The cipher-text and the trapdoor information (v) may be transmitted to the second computing device through the computing network.

[0086] Fig. 4 illustrates a method 400 for hybrid decryption over a computing network, in accordance with an embodiment of the present subject matter. At block 402, the processor of the second computing device receives the one or more predetermined numbers, the predetermined key (e), and the secret key pair (p, q). In one embodiment, the one or more predetermined numbers and the secret key pair (p, q) may be generated by the key management system and provided to the second computing device. For example, the secret key pair (p, q) may be

p=33478071698956898786044169848212690817704794983713768568912431388982888379387800228761471165253174308773781446799948915739666511279233734171433968102700927987363089 17

q=36746043666799590428244633799627952632279158164343087642676032283815739666511279233734171433968102700927987363089 17

[0087] At block 404, the processor receives the cipher-text and the trap-door information (v) from the first computing device. At block 406, on receiving the trap-door information (v), the processor determines the pair of private keys (a_0, d_0) based on the secret key pair (p, q).

[0088] In one embodiment, the processor determines the trap-door parameter (s) and based on the trap-door parameter (s), one or more predetermined numbers, and the secret key pair (p, q), the pair of private keys is determined. For example, the trap-door parameter (s) may be determined as follows:

$$s \equiv (v \pm (v^2 + 4)^{1/2} / 2) \bmod (p*q) \quad \text{.....(12)}$$

[0089] Based on the above equation, there may be two possible values of the trap-door parameter (s). If the trap-door parameter (s) is determined based on the values of v mentioned in description of Fig. 3, then the two values of s may be

s_1=672841774300488625929103855105905032286683665901754593642574778053
706849441651664555899036267528351248706733154416557966701957416588986
832249256511227923919305360290848265229431980910278027645478585250701940500477780763750310560378

s_2=116619456871527444207095889978945324833239675805379881476421368322
147547357194434736176562313202811193219197750800389469560623983377673354900376457637068954406921466060775676130877746957717732156828047585781626204197523048933169609 2

[0090] Where s_1 and s_2 are the two possible values of the trap door parameter. The processor may use s_1 or s_2 for determining the pair of private keys (a_0, d_0) using equations (9) and (10).

[0091] The pair of private keys (a_0, d_0) may be a pair of numbers, where one of the numbers may be a leading term and the other may be a common difference of arithmetic progression. In one embodiment, the pair of private keys may be a pair of co-prime numbers. In one embodiment, the above-mentioned steps (402, 404, 406) may be used for key recovery by computing devices. In one embodiment, the pair of private keys (a_0, d_0) may be refreshed to a new pair of numbers based on the key refreshing model. The key refreshing model refreshes the value of the leading term as explained in equation (1), (2), or (4).

[0092] At block 408, the processor generates the one or more sequences of arithmetic progressions based on at least the pair of private keys (a_0, d_0) and the first set of properties. The generation of the one or more sequences of arithmetic progressions may be performed in a manner as described with respect to Fig. 1. Further, the processor determines the master sequence from the one or more sequences of arithmetic progression.

[0093] At block 410, sub-collections of the arithmetic progressions in the master sequence are determined based on the second set of properties. The second set of properties may include:- (i) the difference between the common differences of

consecutive arithmetic progressions (second common difference) in the sub-collection is equal; (ii) each of the sub-collection is maximal, i.e., includes all arithmetic progressions from the sequence satisfying the property (i).

[0094] Further, the one or more master arithmetic progressions are determined based on the one or more predetermined numbers. For example, the predetermined numbers may indicate position of the master arithmetic progressions in each sub-collection of the master sequence. At block 412, the leading terms and the common differences of the master arithmetic progression may be selected from each sub-collection.

[0095] The selected leading terms and the common differences may be used for determining the master private keys. The master private keys may be one of the terms of the master arithmetic progressions. In one embodiment, the master private keys are determined based on at least the leading terms and the common differences of the master arithmetic progressions, as depicted in block 414.

[0096] At block 416, the master private keys may be used for decrypting the cipher-text into message. In one embodiment, on decrypting the cipher-text a number M may be obtained, which is decoded into the message transmitted by the first computing device.

[0097] Further, Fig. 5 illustrates a method 500 for user authentication, according to an embodiment of the present subject matter. At block 502, a processor of computing device in the computing network, generates a first sequence $\Phi(u, w)$ of arithmetic progressions based on the first set of properties and an authenticating key (u) and a master key (w).

[0098] The authenticating key (u) and the master key (w) may be pair of numbers stored in the memory of the computing device. In one embodiment, the first set of properties may include:- the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor of the pair of private keys (u, w); the

common difference of the arithmetic progressions are in non-increasing order; the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression; the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function. The predetermined function may be based on the leading terms and common differences of the first arithmetic progression, as depicted and explained earlier in equation (6).

[0099] At block 504, the processor selects leading terms of predetermined arithmetic progressions in the first sequence $\Phi(u, w)$ and stores it in the memory. The predetermined arithmetic progression may be based on a number provided as an input by the user of the computing device.

[00100] At block 506, the processor receives an input key (x) provided by a second user for authentication. At block 508, the processor generates a second sequence $\Phi(x, w)$ of arithmetic progressions based on the first set of properties and the input key (x) and master key (w).

[00101] At block 510, the processor determines whether the leading term of the corresponding arithmetic progression of the second sequence $\Phi(x, w)$ is equal to the stored leading terms of the predetermined arithmetic progressions of the first sequence $\Phi(u, w)$.

[00102] At block 512, the processor authenticates the second user based on the determination. If the leading term of the predetermined arithmetic progression of the sequence is not equal to the leading term of the corresponding arithmetic progression of the sequence, then the computing device does not provide authentication to the second user. If the leading term of the predetermined arithmetic progression of the sequence is equal to the leading term of the corresponding arithmetic progression of the sequence, then the computing device authenticates the second user.

[00103] Fig. 6(a) illustrates first computing device 602 and the second computing device 604 in the computing network during symmetric cryptography, in accordance with an embodiment of the present subject matter. As mentioned earlier, the symmetric cryptography involves the sharing of the common pair of private keys and the one or more predetermined numbers.

[00104] In one embodiment, the sharing may be done prior to the encryption process as the shared secret. The shared secret may be known only to the first computing device 602 and the second computing device 604 in the network and may include the pair of private keys, the predetermined key, and the one or more predetermined numbers as shown in the figure. In another embodiment, the pair of private keys and the predetermined numbers may be generated by key management system 606. The key management system 606 may receive a request from one or more computing devices for generating a pair of private keys. The pair of private keys may be generated by selecting a random string of numbers and providing to the requesting computing devices, for example, the first computing device 602 and the second computing device 604.

[00105] The first computing device 602 receives the pair of private keys, the predetermined key (e), and the one or more predetermined numbers for performing encryption of messages to be transmitted to the second computing device 604. In one embodiment, the first computing device 602 includes a memory, processor and an encryption module. The encryption module 608 may be executable by the processor for encrypting the message into the cipher-text. The encryption module 608 is configured to receive a pair of private key (a_0, d_0), one or more predetermined numbers, the predetermined key (e), and message to be transmitted to a second computing device 604.

[00106] Further, the encryption module generates one or more sequences of arithmetic progressions based on first set of properties and the pair of private keys (a_0, d_0). The number of sequences generated is equivalent to the greatest common

divisor of the pair of private keys (a_0, d_0). Further, the master sequence is determined based on the predetermined key (e). Within the master sequence, sub-collections of arithmetic progressions are determined based on a second set of properties, and one or more master arithmetic progressions are also determined in each of the sub-collection based on the one or more predetermined numbers.

[00107] The encryption module 608 may select leading terms and common differences of the master arithmetic progressions from each of the sub-collections and determine the master private keys based on the leading terms and the common differences of each master arithmetic progression.

[00108] Further, the encryption module 608 encrypts the message into a cipher-text using the master private keys. The cipher-text is transmitted to the second computing device 604.

[00109] The second computing device 604 includes memory, processor, and a decryption module 610 executable by the processor, according to an embodiment. The processor determines the master private keys in the manner as described for the first computing device 602. On receiving the cipher-text, the decryption module 610 decrypts the cipher-text into the message sent by the first computing device 602 based on the master private keys.

[00110] Fig. 6(b) illustrates the first computing device 602 and the second computing device 604 in the computing network during asymmetric cryptography, in accordance with an embodiment of the present subject matter. As mentioned earlier, the asymmetric cryptography involves two different keys, for example, the public key and the secret key.

[00111] As shown in the figure, the public key and the secret key are generated by the key management system. The first computing device may request for generating secret key pair and provide the secret key pair secretly to it. Further, it may also request the key management system to generate a public key based on the

secret key pair and provide it to other computing devices of the computing network, which request for the public key.

[00112] The encryption module 608 of the first computing device 602 receives the public key, the predetermined key (e), and the one or more predetermined numbers for determining the pair of private keys. Further, the encryption module 608 selects a trap-door parameter and based on the trap-door parameter, the public key and the one or more predetermined numbers, the encryption module determines the trap-door information (v). Based on the pair of private keys, the first computing device determines the master private keys, which is used for the encrypting the messages into cipher-text by the encryption module 608.

[00113] The first computing device 602 transmits the cipher-text and the trap-door information to the second computing device 604. The decryption module 610 of the second computing device 604 determines the pair of private keys from the trap-door information using the secret key. Based on the pair of private keys the decryption module 610 determines the master private key in manner described for the encryption module 608. The decryption module 610 decrypts the cipher-text into the message sent by the first computing device 602.

[00114] Fig. 7 illustrates computing devices in a computing network 700, in accordance with an embodiment of the present subject matter. The computing network 702 includes a plurality of computing devices (704-1, ..., 704-N) performing encryption and decryption during communication of messages in the computing network 702. In one embodiment, the computing network 702 may be a public network including a large number of computing devices 704. In another embodiment, the computing network 702 may be a private network with a limited number of computing devices.

[00115] The computing devices 704 may include one or more of servers, a desktop personal computer, a portable computer, a workstation, a mainframe computer, a laptop, communication devices, such as mobile phones and smart

phones, etc. The computing device 704 may refer to the first computing device 704-1 and the second computing device 704-2. The computing device 704 includes the processor 706 coupled to the memory 708, the modules 710, data 712, and the interface 714. The processor 706 obtains the pair of private keys (a_0, d_0), the one or more predetermined numbers, the predetermined key (e) and the message to be transmitted to the second computing device 704-2. In one embodiment, the processor 706 determines the pair of private keys (a_0, d_0) based on the public key and the one or more predetermined numbers received from the Key Management System (not shown in figure).

[00116] Further, the message to be transmitted to the second computing device may already be stored in the memory 708 of the computing device 704. The memory 708 may include any device known in the art including, for example, volatile memory, such as static random access memory (SRAM) and dynamic random access memory (DRAM), and/or non-volatile memory, such as read only memory (ROM), erasable programmable ROM, hard disks, optical disks, and magnetic tapes.

[00117] The memory 708 further includes module 710 and data 712. The module 710 includes programs, routines, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types. The data 712 serves as a repository for storing data processed, received, and generated by one or more of the module 712 and the processor 704. In one embodiment, the module 710 may be an encryption module and/or a decryption module, which further includes a key generation module 710-1, the parameter generation module 710-2, and a key refreshing module 710-3. In one embodiment, the key generation module 710-1 may be used for generating the pair of private keys by determining new leading term and common difference based on random substrings.

[00118] Further, the message may be provided by an external memory connected to the computing device or may be provided by a user through an interface 714. In one embodiment, the interface 714 may include hardware interface, for

example, interface for peripheral device(s), such as a keyboard, a mouse, an external memory, a printer, etc. Additionally, the interface 714 may facilitate multiple communications within a wide variety of protocols and networks, such as a network, including wired networks, e.g., LAN, cable, etc., and wireless networks, e.g., WLAN, cellular, satellite, etc.

[00119] Yet further, the processor 706 generates one or more sequences of arithmetic progressions based on the first set of properties and the pair of private keys (a_0, d_0). The processor 706 retrieves the first set of properties from the memory 708 and generates the one or more sequences of arithmetic progressions. The number of sequences generated is equivalent to the greatest common divisor of the pair of private keys. The processor then determines the master sequence from the one or more sequences based on the predetermined key (e).

[00120] Further, the processor 706 determines sub-collections of arithmetic progressions in the master sequence and one or more master arithmetic progression from each sub-collection. The determination of sub-collections of arithmetic progressions in the sequence are based on the second set of properties stored in the memory 708 and the one or more master arithmetic progressions are determined based on the one or more predetermined numbers.

[00121] The parameter generation module 710-2 generates the leading terms and the common difference of each arithmetic progression. The processor 706 selects the leading terms and the common difference of the master arithmetic progressions from each of the sub-collection. Based on at least the leading terms and the common difference of the master arithmetic progression, the processor 706 determines the master private keys from each master arithmetic progression.

[00122] Further, the processor encrypts the messages into cipher-text based on the master private key. The cipher-text is transmitted to the second computing device and the processor of the second computing device determines the master private keys

as explained earlier. Based on the master private key, the processor decrypts the cipher-text into messages.

[00123] In one embodiment, the pair of private keys (a_0, d_0) may be refreshed to a new pair of numbers by a key refreshing module. The key refreshing module may use the key refreshing model to generate a new pair of numbers. The key refreshing module may refresh the value of the leading term using equation (1), (2), or (4), which may be stored in the memory 708. The new pair of co-prime numbers may be generated for encrypting messages of larger sizes.

[00124] The present subject matter can be used in symmetric and asymmetric encryption to provide secure communication of messages in mobile communications, wireless communications, and the like. The cryptographic methods of the present subject matter may be used for securely accessing data in a cloud environment. Further, the user authentication offers client-independent access, i.e., users need not change their password and only master key can be changed periodically to prevent attacks like recovery of passwords from the stored leading terms in system's memory. Further, the present subject matter provides a simple encryption and decryption method that can be customizable as it provides a wide choice of values for private key pair, predetermined numbers, and sequences. Additionally, the cryptographic methods can also be implemented on a wide variety of hardware, such as servers, a desktop personal computer, a portable computer, a workstation, a mainframe computer, a laptop, communication devices, such as mobile phones and smart phones, and the like.

[00125] Although embodiments for encryption and decryption of messages have been described in language specific to structural features and/or methods, it is to be understood that the above subject matter is not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as exemplary implementations for encryption and decryption of messages.

I/We claim:

1. A method for encryption of messages transmitted over a computing network (702), the method comprising:

obtaining, by a processor (706), a pair of private keys (a_0, d_0), one or more predetermined numbers, a predetermined key (e), and a message;

generating, by the processor (706), one or more sequences of arithmetic progressions based on a first set of properties and the pair of private keys (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determining, by the processor (706), sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and one or more master arithmetic progressions in each of the sub-collections based on the one or more predetermined numbers;

selecting, by the processor (706), leading terms and common differences of the one or more master arithmetic progressions from each of the sub-collections;

determining, by the processor (706), master private keys based on at least the leading terms and the common differences of each master arithmetic progression; and

encrypting, by the processor (706), the message into a cipher-text using the master private keys.

2. The method as claimed in claim 1 further comprising transmitting the cipher-text to a computing device in the computing network.

3. The method as claimed in claim 1, wherein the message is encoded to a number (M) and split into smaller numbers (m_1, m_2, …, m_i) before encryption.

4. A method for decryption of cipher-text received over a computing network (702), the method comprising:

obtaining, by a processor (706), a cipher-text, a pair of private keys (a_0, d_0), a predetermined key (e), and one or more predetermined numbers;

generating, by the processor (706), one or more sequences of arithmetic progressions based on a first set of properties and the pair of private keys (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determining, by the processor (706), sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and master arithmetic progressions in each of the sub-collections based on the one or more predetermined numbers;

selecting, by the processor (706), leading terms and common differences of the master arithmetic progressions from each of the sub-collections;

determining, by the processor (706), master private keys based on the leading terms and the common differences of each master arithmetic progression; and

decrypting, by the processor (706), the cipher-text into a message using the master private keys.

5. The method as claimed in claim 4, wherein the cipher text (C) is split into smaller numbers (c_1, c_2, …, c_i) before decryption.

6. The method as claimed in claim 1 or 4, wherein the master private keys are determined based on at least the leading terms, the common differences, and a second common difference.

7. The method as claimed in claim 1 or 4, wherein the pair of private keys (a_0, d_0) is a pair of numbers, and wherein the numbers are the leading term and the common difference of the first arithmetic progression in the sequence.

8. The method as claimed in claim 7, wherein the pair of numbers are co-prime numbers.

9. The method as claimed in claim 1 or 4, wherein the first set of properties includes:

i) the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor of the pair of private keys;

ii) the common difference of the arithmetic progressions are in non-increasing order;

iii) the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression; and

iv) the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function

10. The method as claimed in claim 1 or 4, wherein the second set of properties include:

i) the difference between the common differences of consecutive arithmetic progressions in the sub-collection is equal; and

ii) the sub-collection is maximal.

11. The method as claimed in claim 1 or 4, wherein the one or more predetermined numbers and the pair of private keys (a_0, d_0) are agreed between a first computing device (602) and a second computing device (604) of the computing network (702).

12. A method for encryption of messages transmitted over a computing network (702), the method comprising:

receiving, by the processor (706), one or more predetermined numbers, a predetermined key (e), and a public key (△), wherein the public key (△) is based on a secret key pair (p, q);

selecting, by the processor (706), a trap-door parameter (s), wherein the trap-door parameter (s) is co-prime to the public key (△);

determining, by the processor (706), a pair of private keys (a_0, d_0) and a trap-door information (v) based on one or more of the public key (△), the one or more predetermined numbers, and the trap-door parameter (s);

generating, by the processor (706), one or more sequences of arithmetic progressions based on a first set of properties and the private key (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determining, by the processor (706), sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and one or more master arithmetic progressions in each of the sub-collections based on the one or more predetermined numbers;

selecting, by the processor (706), leading terms and common differences of the one or more master arithmetic progressions from each of the sub-collections;

determining, by the processor (706), master private keys based on at least the leading terms and the common differences of each master arithmetic progression; and

encrypting, by the processor (706), the message into a cipher-text using the master private keys.

13. The method as claimed in claim 12, wherein the trap-door information (v) and the cipher-text are transmitted to a second computing device (604).

14. A method for decryption of cipher-text received over a computing network (702), the method comprising:

receiving, by the processor (706), one or more predetermined numbers, predetermined key (e), and a secret key pair (p, q);

receiving, by the processor (706), a trap-door information (v) from a computing device;

determining, by the processor (706), a pair of private keys (a_0, d_0) from the trap-door information (v) based on the secret key pair (p, q);

generating, by the processor (706), a sequence of arithmetic progressions based on a first set of properties and the pair of private keys (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determining, by the processor (706), sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and master arithmetic progressions in each of the sub-collections based on the one or more predetermined numbers;

selecting, by the processor (706), leading terms and common differences of the master arithmetic progressions from each of the sub-collections;

determining, by the processor (706), master private keys based on the leading terms and the common differences of each master arithmetic progression; and

decrypting, by the processor, the cipher-text into a message using the master private keys.

15. The method as claimed in claim 12 or 14, wherein the first set of properties includes:

i) the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor of the pair of private keys;

ii) the common difference of the arithmetic progressions are in non-increasing order;

iii) the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression; and

iv) the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function.

16. The method as claimed in claim 12 or 14, wherein the second set of properties include:

i) the difference between the common differences of consecutive arithmetic progressions in the sub-collection is equal; and

ii) the sub-collection is maximal.

17. The method as claimed in claim 12 or 14, wherein the one or more predetermined numbers, the public key (△), predetermined key (e), and the secret key pair (p, q) are generated by a key management system.

18. The method as claimed in claim 12 or 14, wherein the secret key pair (p, q) is a pair of prime numbers, and wherein a product of the prime numbers is the public key (△).

19. The method as claimed in claim 12 or 14, wherein the trap-door parameter (s) is lesser than the public key (△) and greater than half of the public key (△).

20. The method as claimed in any of the preceding claims, wherein the pair of private keys (a_0, d_0) is refreshed to a new pair of numbers based on a key refreshing model.

21. A method for user authentication, the method comprising:

generating, by a processor of a computing device, a first sequence $\Phi(u, w)$ of arithmetic progressions based on a first set of properties and an authenticating key (u) and master key (w);

selecting and storing, by the processor, leading terms of predetermined arithmetic progressions in the first sequence $\Phi(u, w)$;

receiving, by the processor, an input key (x) provided by a user for authentication;

generating, by the processor, a second sequence $\Phi(x, w)$ of arithmetic progressions based on the first set of properties and the input key (x) and master key (w);

determining, by the processor, whether the leading terms of the predetermined arithmetic progressions of the second sequence $\Phi(x, w)$ are equal to the stored leading terms of the predetermined arithmetic progressions of the first sequence $\Phi(u, w)$; and

authenticating, by the processor, the user based on the determination.

22. The method as claimed in claim 21, wherein the first set of properties includes:

i) the product of j^{th} terms of i^{th} and $(i+1)^{th}$ arithmetic progressions is congruent to G modulo $(j+1)^{th}$ term of the i^{th} arithmetic progression, where G is the greatest common divisor of the pair of private keys;

ii) the common difference of the arithmetic progressions are in non-increasing order;

iii) the product of common difference of $(i+1)^{th}$ arithmetic progression and the leading term of i^{th} arithmetic progression is congruent to G modulo the common difference of the i^{th} arithmetic progression; and

iv) the leading term of the $(i+1)^{th}$ arithmetic progression is equal to the addition of the common difference of the $(i+1)^{th}$ arithmetic progression and a predetermined function.

23. A computing device (704) for encrypting messages, the computing device (704) comprising:

a memory (708);

a processor (706) coupled to the memory (708);

an encryption module (608) executable by the processor, wherein the encryption module (608) is configured to:

receive a pair of private keys (a_0, d_0), a predetermined key (e), one or more predetermined numbers, and a message to be transmitted to a second computing device;

generate a sequence of arithmetic progressions based on a first set of properties and the pair of private keys (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determine sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and one or more master arithmetic progressions in each of the sub-collection based on the one or more predetermined numbers;

select leading terms and common differences of the master arithmetic progressions from each of the sub-collections;

determine master private keys based on the leading terms and the common differences of each master arithmetic progressions; and

encrypt the message into a cipher-text using the master private keys.

24. The computing device (704) as claimed in claim 23, wherein the encryption module (608) is further configured to:

receive the one or more predetermined numbers, and a public key (Δ), wherein the public key (Δ) is based on a secret key (p, q);

select a trap-door parameter (s), wherein the trap-door parameter (s) is co-prime to the public key (Δ); and

determine the pair of private keys (a_0, d_0) and a trap-door information (v) based on one or more of the public key (Δ), the one or more predetermined numbers, and the trap-door parameter (s).

25. A computing device (704) for decrypting messages, the computing device (704) comprising:

a memory (708);

a processor (706) coupled to the memory (708);

a decryption module (610) executable by the processor, wherein the decryption module (610) is configured to:

receive a cipher-text, a pair of private keys (a_0, d_0), a predetermined key (e), and one or more predetermined numbers;

generate a sequence of arithmetic progressions based on a first set of properties and the pair of private keys (a_0, d_0), wherein the number of sequences generated is the greatest common divisor of the pair of private keys (a_0, d_0);

determining, by the processor (706), a master sequence from the one or more sequences based on the predetermined key (e);

determine sub-collections of arithmetic progressions in the master sequence based on a second set of properties, and a master arithmetic progression in each of the sub-collection based on the one or more predetermined numbers;

select leading terms and common differences of the master arithmetic progression from each of the sub-collection;

determine master private keys based on at least the leading term and the common difference of each master arithmetic progression; and

decrypt the cipher-text into a message using the master private keys.

26. The computing device (704) as claimed in claim 25, wherein the decryption module (610) is further configured to:

receive the one or more predetermined numbers, and a secret key (p, q);

receive the cipher-text and a trap-door information (v) from a first computing device; and

determine the private key (a_0, d_0) from the trap-door information (v) based on the secret key (p, q).

27. The computing device (704) as claimed in claim 23 or 25, wherein the encryption and decryption module (610) comprises:

a key generation module (710-1) for generating a pair of private keys (a_0, d_0);

a parameter generation module (710-2) for generating leading terms and common differences based on the first set of properties; and

a key refreshing module (710-3) for refreshing the private keys based on a key refreshing model.

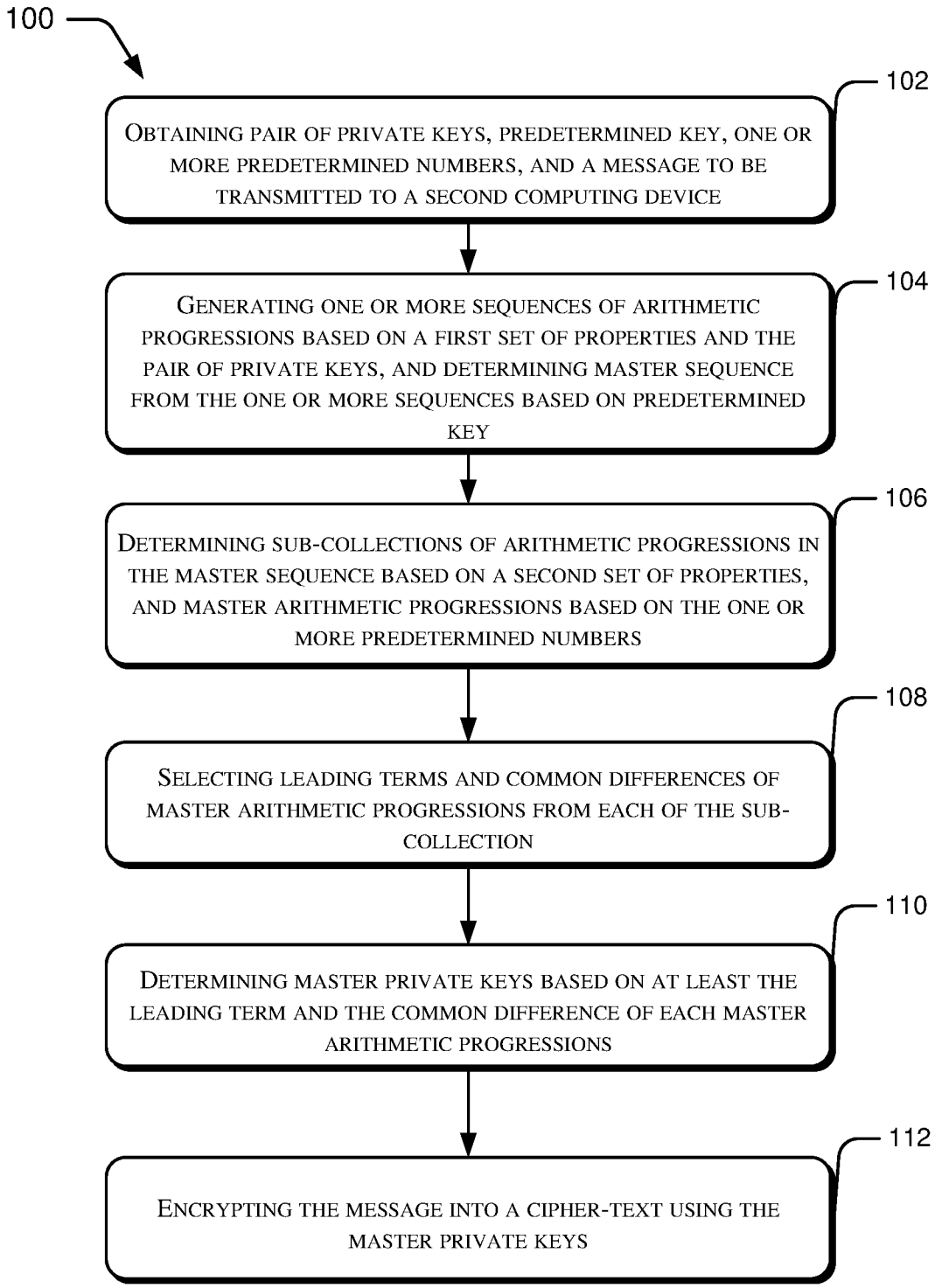
100
102
OBTAINING PAIR OF PRIVATE KEYS, PREDETERMINED KEY, ONE OR MORE PREDETERMINED NUMBERS, AND A MESSAGE TO BE TRANSMITTED TO A SECOND COMPUTING DEVICE
104
GENERATING ONE OR MORE SEQUENCES OF ARITHMETIC PROGRESSIONS BASED ON A FIRST SET OF PROPERTIES AND THE PAIR OF PRIVATE KEYS, AND DETERMINING MASTER SEQUENCE FROM THE ONE OR MORE SEQUENCES BASED ON PREDETERMINED KEY
106
DETERMINING SUB-COLLECTIONS OF ARITHMETIC PROGRESSIONS IN THE MASTER SEQUENCE BASED ON A SECOND SET OF PROPERTIES, AND MASTER ARITHMETIC PROGRESSIONS BASED ON THE ONE OR MORE PREDETERMINED NUMBERS
108
SELECTING LEADING TERMS AND COMMON DIFFERENCES OF MASTER ARITHMETIC PROGRESSIONS FROM EACH OF THE SUB-COLLECTION
110
DETERMINING MASTER PRIVATE KEYS BASED ON AT LEAST THE LEADING TERM AND THE COMMON DIFFERENCE OF EACH MASTER ARITHMETIC PROGRESSIONS
112
ENCRYPTING THE MESSAGE INTO A CIPHER-TEXT USING THE MASTER PRIVATE KEYS

Fig. 1

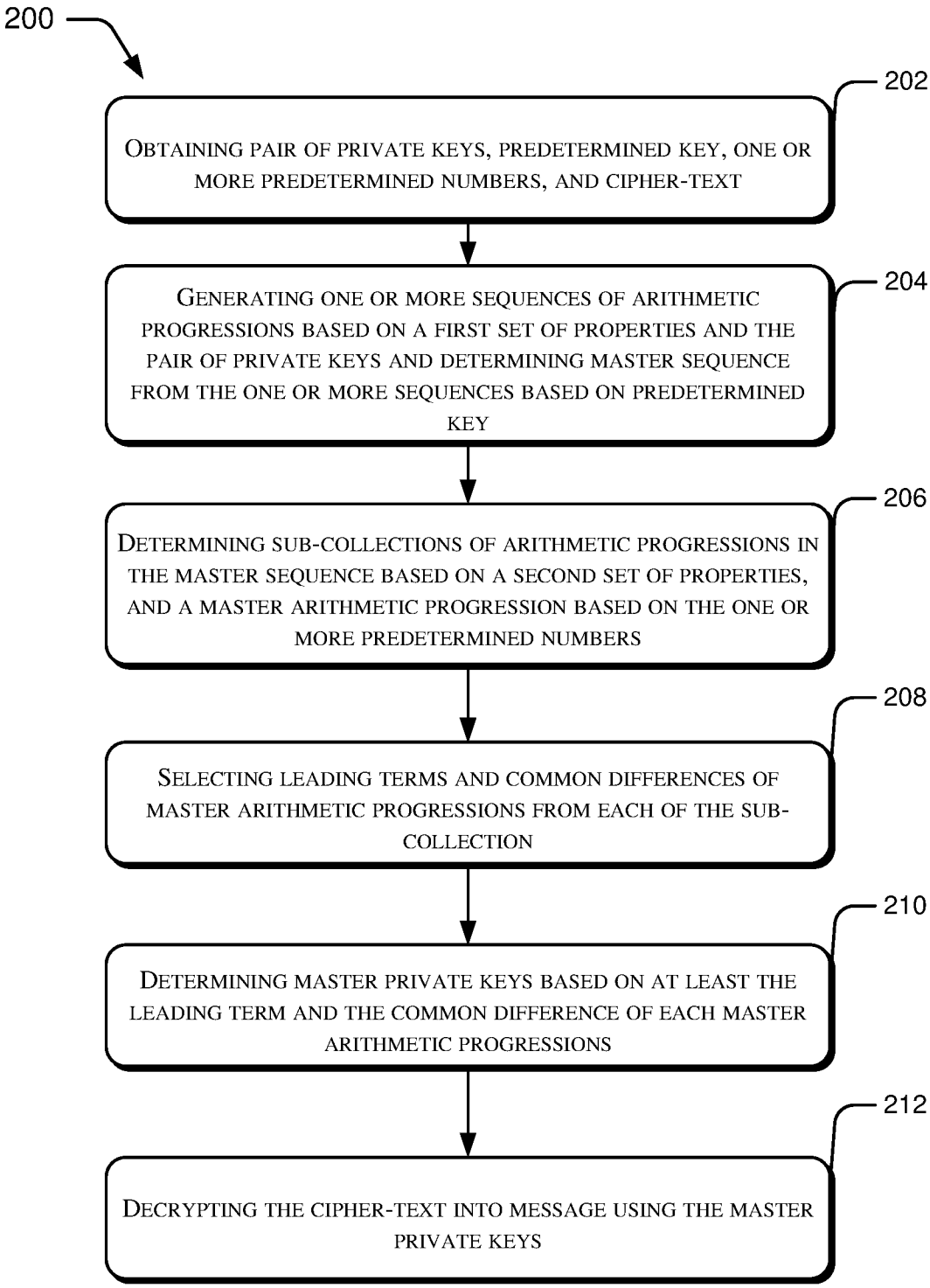
200
202
OBTAINING PAIR OF PRIVATE KEYS, PREDETERMINED KEY, ONE OR MORE PREDETERMINED NUMBERS, AND CIPHER-TEXT
204
GENERATING ONE OR MORE SEQUENCES OF ARITHMETIC PROGRESSIONS BASED ON A FIRST SET OF PROPERTIES AND THE PAIR OF PRIVATE KEYS AND DETERMINING MASTER SEQUENCE FROM THE ONE OR MORE SEQUENCES BASED ON PREDETERMINED KEY
206
DETERMINING SUB-COLLECTIONS OF ARITHMETIC PROGRESSIONS IN THE MASTER SEQUENCE BASED ON A SECOND SET OF PROPERTIES, AND A MASTER ARITHMETIC PROGRESSION BASED ON THE ONE OR MORE PREDETERMINED NUMBERS
208
SELECTING LEADING TERMS AND COMMON DIFFERENCES OF MASTER ARITHMETIC PROGRESSIONS FROM EACH OF THE SUB-COLLECTION
210
DETERMINING MASTER PRIVATE KEYS BASED ON AT LEAST THE LEADING TERM AND THE COMMON DIFFERENCE OF EACH MASTER ARITHMETIC PROGRESSIONS
212
DECRYPTING THE CIPHER-TEXT INTO MESSAGE USING THE MASTER PRIVATE KEYS

Fig. 2

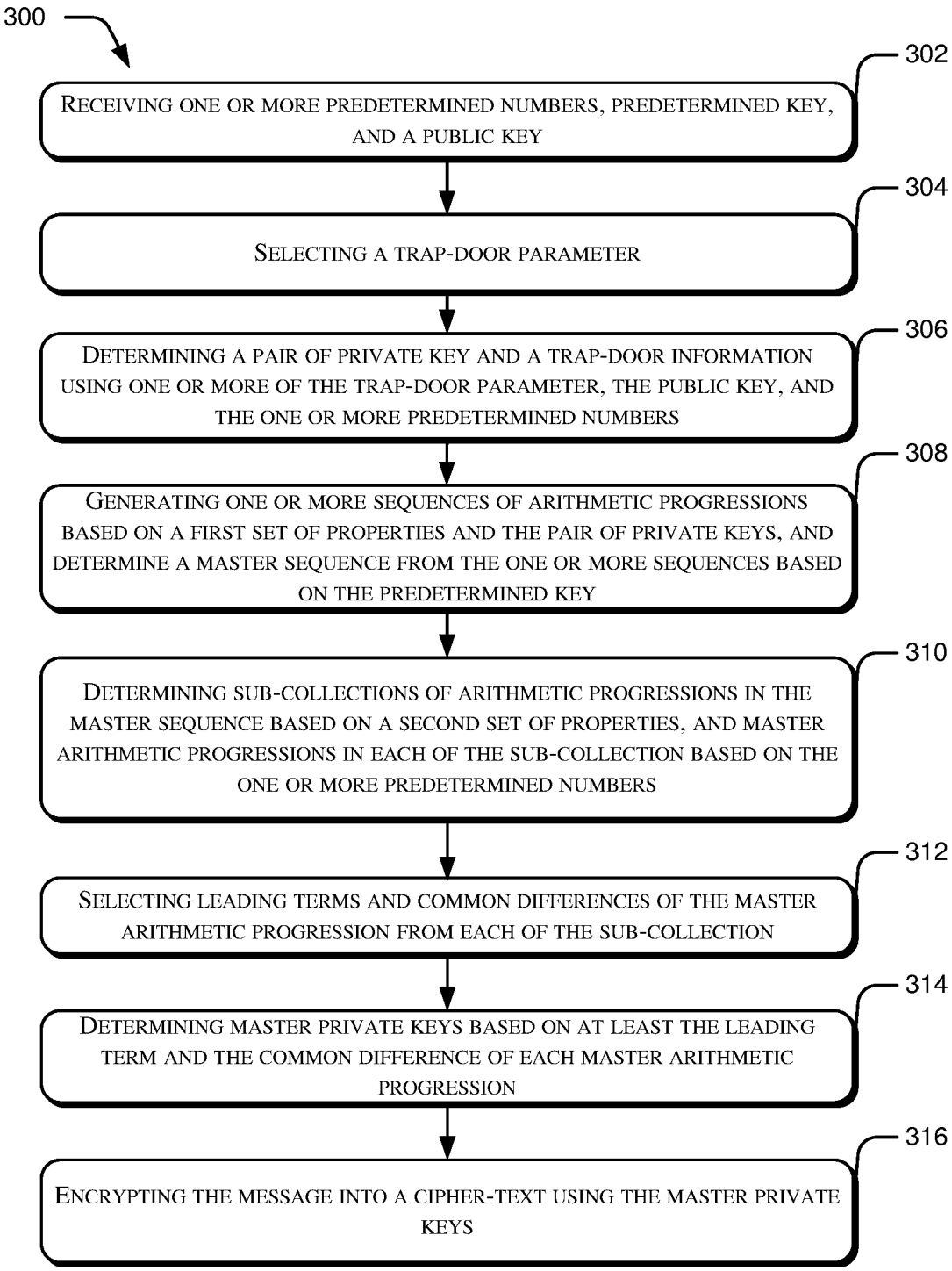
300
302
RECEIVING ONE OR MORE PREDETERMINED NUMBERS, PREDETERMINED KEY, AND A PUBLIC KEY
304
SELECTING A TRAP-DOOR PARAMETER
306
DETERMINING A PAIR OF PRIVATE KEY AND A TRAP-DOOR INFORMATION USING ONE OR MORE OF THE TRAP-DOOR PARAMETER, THE PUBLIC KEY, AND THE ONE OR MORE PREDETERMINED NUMBERS
308
GENERATING ONE OR MORE SEQUENCES OF ARITHMETIC PROGRESSIONS BASED ON A FIRST SET OF PROPERTIES AND THE PAIR OF PRIVATE KEYS, AND DETERMINE A MASTER SEQUENCE FROM THE ONE OR MORE SEQUENCES BASED ON THE PREDETERMINED KEY
310
DETERMINING SUB-COLLECTIONS OF ARITHMETIC PROGRESSIONS IN THE MASTER SEQUENCE BASED ON A SECOND SET OF PROPERTIES, AND MASTER ARITHMETIC PROGRESSIONS IN EACH OF THE SUB-COLLECTION BASED ON THE ONE OR MORE PREDETERMINED NUMBERS
312
SELECTING LEADING TERMS AND COMMON DIFFERENCES OF THE MASTER ARITHMETIC PROGRESSION FROM EACH OF THE SUB-COLLECTION
314
DETERMINING MASTER PRIVATE KEYS BASED ON AT LEAST THE LEADING TERM AND THE COMMON DIFFERENCE OF EACH MASTER ARITHMETIC PROGRESSION
316
ENCRYPTING THE MESSAGE INTO A CIPHER-TEXT USING THE MASTER PRIVATE KEYS

Fig. 3

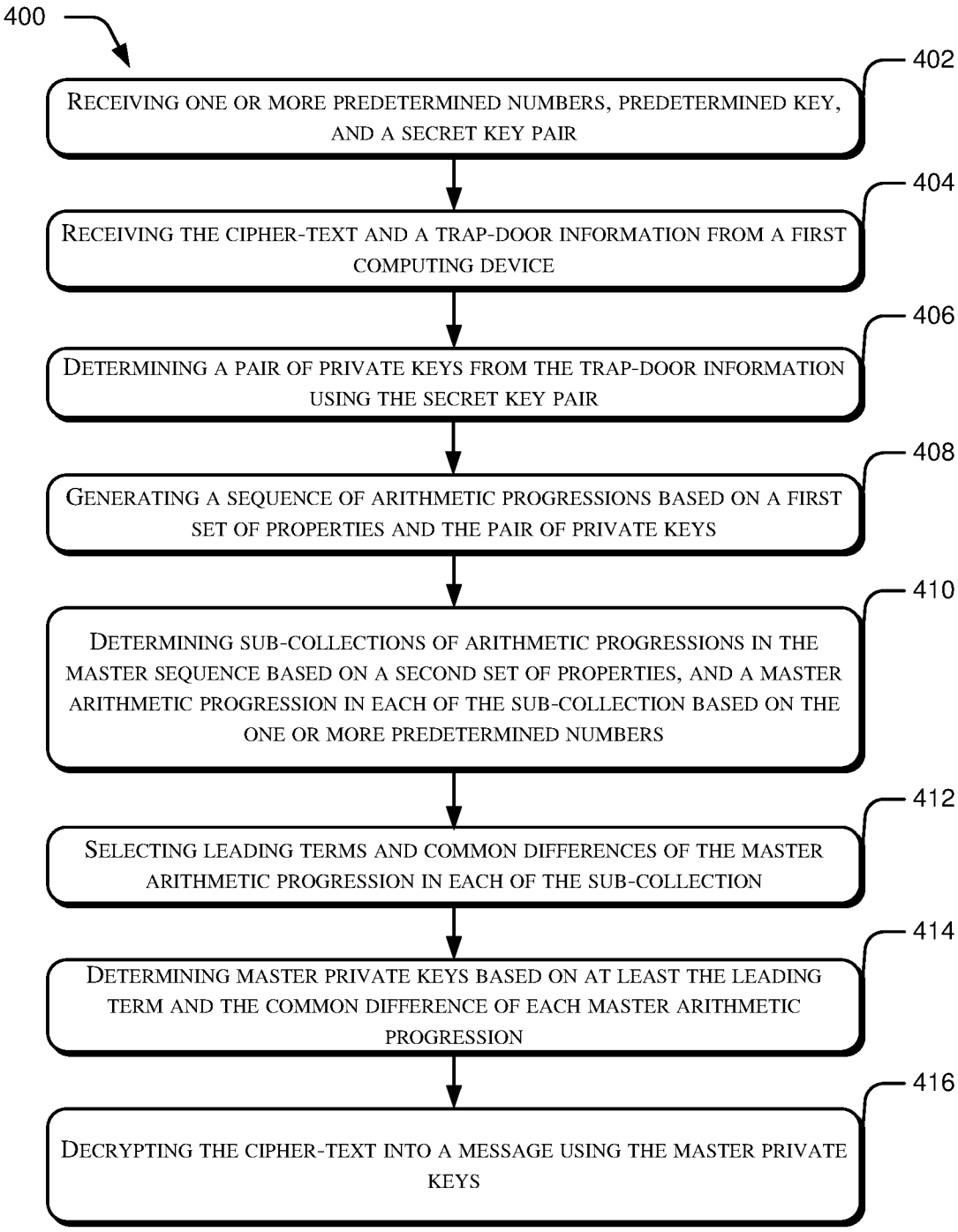
400
402
RECEIVING ONE OR MORE PREDETERMINED NUMBERS, PREDETERMINED KEY, AND A SECRET KEY PAIR
404
RECEIVING THE CIPHER-TEXT AND A TRAP-DOOR INFORMATION FROM A FIRST COMPUTING DEVICE
406
DETERMINING A PAIR OF PRIVATE KEYS FROM THE TRAP-DOOR INFORMATION USING THE SECRET KEY PAIR
408
GENERATING A SEQUENCE OF ARITHMETIC PROGRESSIONS BASED ON A FIRST SET OF PROPERTIES AND THE PAIR OF PRIVATE KEYS
410
DETERMINING SUB-COLLECTIONS OF ARITHMETIC PROGRESSIONS IN THE MASTER SEQUENCE BASED ON A SECOND SET OF PROPERTIES, AND A MASTER ARITHMETIC PROGRESSION IN EACH OF THE SUB-COLLECTION BASED ON THE ONE OR MORE PREDETERMINED NUMBERS
412
SELECTING LEADING TERMS AND COMMON DIFFERENCES OF THE MASTER ARITHMETIC PROGRESSION IN EACH OF THE SUB-COLLECTION
414
DETERMINING MASTER PRIVATE KEYS BASED ON AT LEAST THE LEADING TERM AND THE COMMON DIFFERENCE OF EACH MASTER ARITHMETIC PROGRESSION
416
DECRYPTING THE CIPHER-TEXT INTO A MESSAGE USING THE MASTER PRIVATE KEYS

Fig. 4

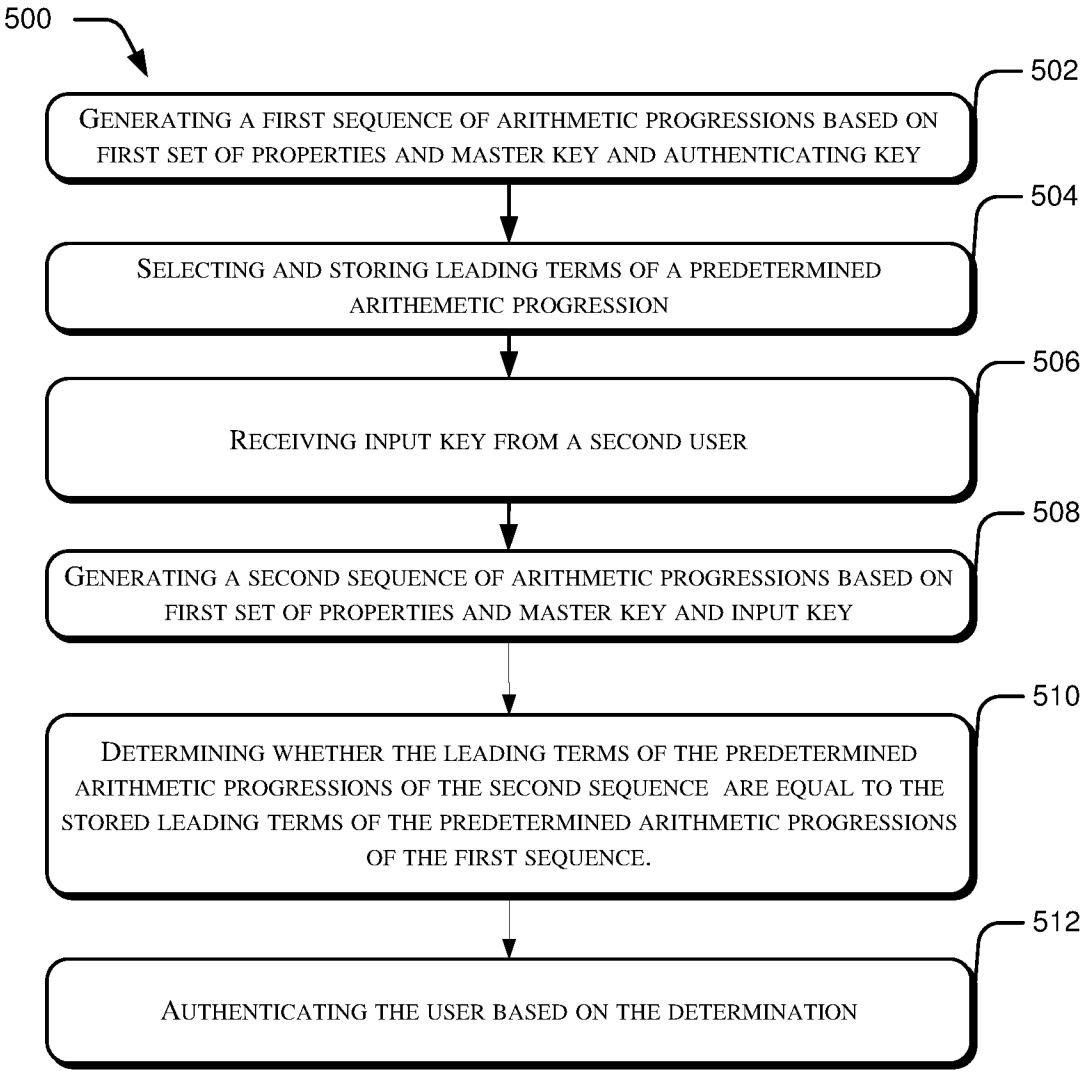
500
GENERATING A FIRST SEQUENCE OF ARITHMETIC PROGRESSIONS BASED ON FIRST SET OF PROPERTIES AND MASTER KEY AND AUTHENTICATING KEY
502
SELECTING AND STORING LEADING TERMS OF A PREDETERMINED ARITHEMETIC PROGRESSION
504
RECEIVING INPUT KEY FROM A SECOND USER
506
GENERATING A SECOND SEQUENCE OF ARITHMETIC PROGRESSIONS BASED ON FIRST SET OF PROPERTIES AND MASTER KEY AND INPUT KEY
508
DETERMINING WHETHER THE LEADING TERMS OF THE PREDETERMINED ARITHMETIC PROGRESSIONS OF THE SECOND SEQUENCE ARE EQUAL TO THE STORED LEADING TERMS OF THE PREDETERMINED ARITHMETIC PROGRESSIONS OF THE FIRST SEQUENCE.
510
AUTHENTICATING THE USER BASED ON THE DETERMINATION
512

Fig. 5

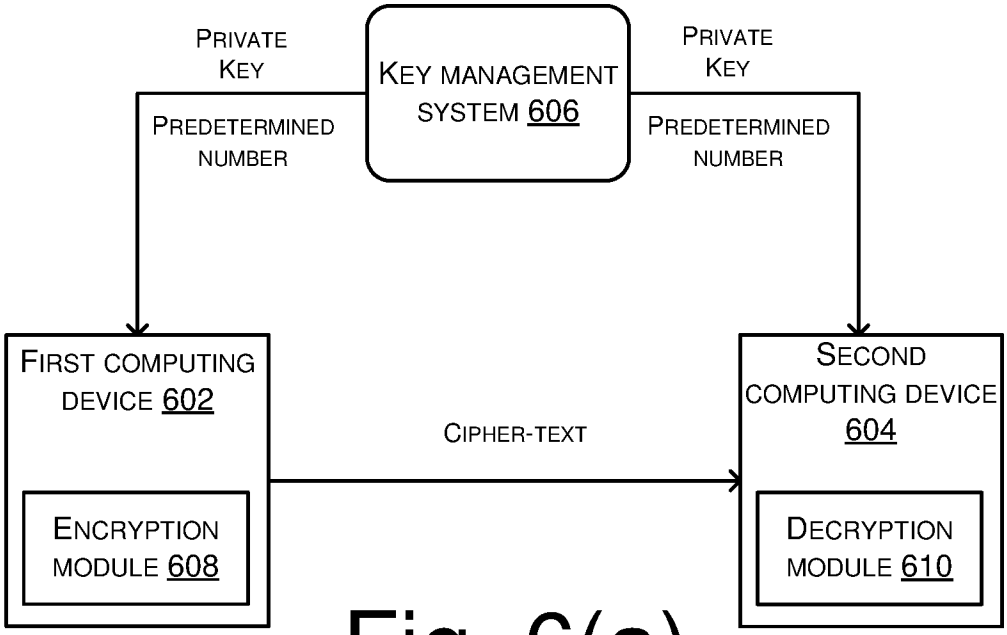
PRIVATE KEY
PREDETERMINED NUMBER
KEY MANAGEMENT SYSTEM 606
PRIVATE KEY
PREDETERMINED NUMBER
FIRST COMPUTING DEVICE 602
ENCRYPTION MODULE 608
CIPHER-TEXT
SECOND COMPUTING DEVICE 604
DECRYPTION MODULE 610

Fig. 6(a)

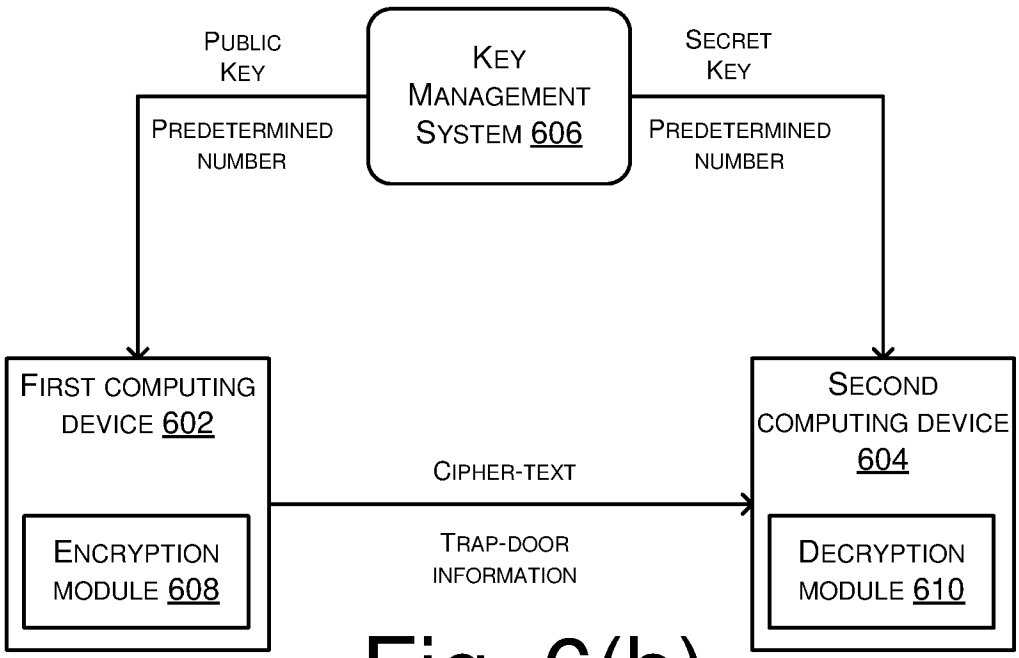
PUBLIC KEY
PREDETERMINED NUMBER
KEY MANAGEMENT SYSTEM 606
SECRET KEY
PREDETERMINED NUMBER
FIRST COMPUTING DEVICE 602
ENCRYPTION MODULE 608
CIPHER-TEXT
TRAP-DOOR INFORMATION
SECOND COMPUTING DEVICE 604
DECRYPTION MODULE 610

Fig. 6(b)

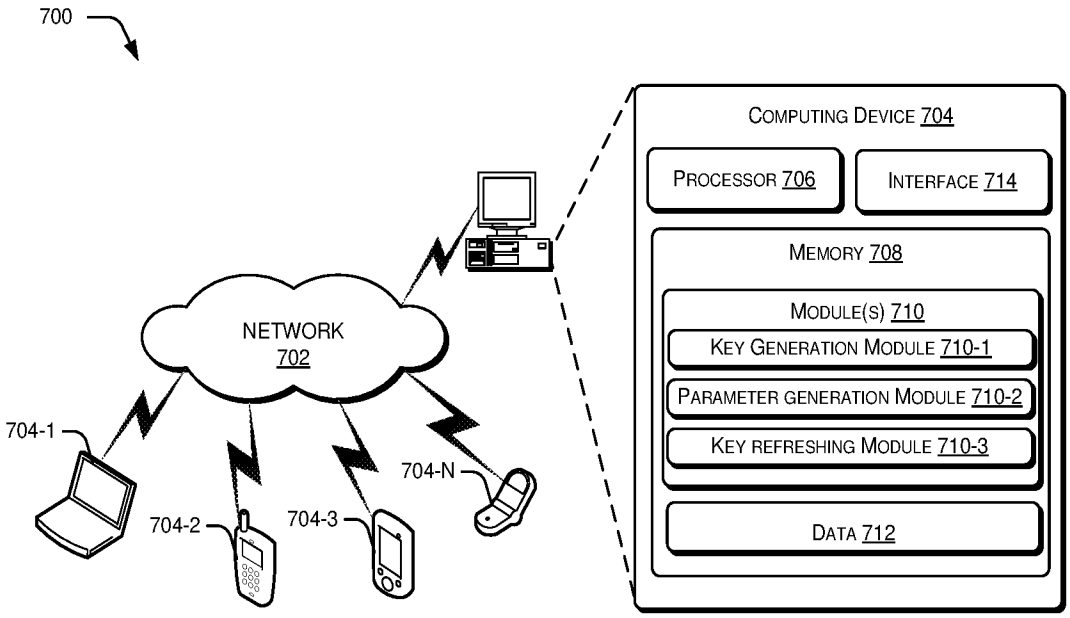
700
NETWORK
702
704-1
704-2
704-3
704-N
COMPUTING DEVICE 704
PROCESSOR 706
INTERFACE 714
MEMORY 708
MODULE(S) 710
KEY GENERATION MODULE 710-1
PARAMETER GENERATION MODULE 710-2
KEY REFRESHING MODULE 710-3
DATA 712

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2017/050289

A. CLASSIFICATION OF SUBJECT MATTER
G06F21/60,H04L9/00,H04L9/28 Version=2017.01

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)

H04L, G06F

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

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

Patseer, IPO Internal Database
Keywords: encryption, decryption, message, private key, arithmetic

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2005002532 A1 (YONGXIN ZHOU et al.), 6 January 2005 (06-01-2005) Abstract; Paragraphs [0025]-[0030], [0037]-[0094]; Claims 1, 30, 42, 50, 52	1-27
Y	US 2004223616 A1 (KOCAREV LJUPCO et al.), 11 November 2004 (11-11-2004) Abstract; Paragraphs [0025]-[0032], [0035]-[0062], [0179]-[0249]; Claims 31, 59	1-27

☐ 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
18-09-2017	18-09-2017
Name and mailing address of the ISA/ Indian Patent Office Plot No.32, Sector 14,Dwarka,New Delhi-110075 Facsimile No.	Authorized officer Vishal Shukla Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IN2017/050289

Citation	Pub.Date	Family	Pub.Date
US 2005002532 A1	06-01-2005	WO 2003065639 A2	07-08-2003
		CA 2369304 A1	30-07-2003
		AU 2003203207 A1	02-09-2003
US 2004223616 A1	11-11-2004	DE 602004001158 D1	27-07-2006
		EP 1467512 A1	13-10-2004
		DE 60322338 D1	04-09-2008