Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Facile Solution on Secret Key generation and Secure Distribution

Lifeng Cai (lifengcai@yahoo.com)
Academy of Military Medical Sciences

Physical Sciences - Article

Keywords: cryptosystem, symmetric encryption algorithm, cipher, cryptography, provable security, key distribution, key management, information security

Posted Date: October 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-990969/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Abstract

We designed systems to generate and securely distribute unlimited number of secret keys from a master key with key strength of the system can be equal to the bit-count of the master key. We first set 3 prerequisites for system security from basic principles and design scheme accordingly, use master key as the solely system privacy, combine the master key and arbitrarily unrepeated timestamps to generate medium keys, generate keys therefrom using one-way data conversion algorithm (OWDCA), and distribute keys by corresponding timestamps between legitimate users exclusively sharing the master key. We then defined 4 criterions for qualified OWDCA (q-OWDCA) and proved that systems based on the above scheme using q-OWDCA would fulfill these prerequisites. We designed some q-OWDCAs and constructed typical cryptosystems, demonstrating a universal strategy to realize information security in large complex systems using a single master key. Our work may solve the fundamental problem in symmetric cryptography.

Introduction

The rapid development in information sciences and communication technologies and the resulted emerging areas, such as internet of things, cloud computing, and cloud data storage, pose huge challenges in information security, and they also bring new opportunities. Secure communications and secure data storages are core issues for information security where cryptography plays the central role.

Encryption-decryption using an exclusively shared key between legitimate communicators makes user-friendly ciphers and is widely used in information security. However, repeatedly using the same key raises security problems and various attack strategies have been developed targeting specific encryption algorithm. One-time key strategy can avoid most kinds of attacks, while brings more problems for generation and secure distribution of large number of one-time keys. So far, it is still a formidable task to design a convenient, high-efficient, universal and provably secure cipher. For privacy concerns, communicator pair must use unique shared key, feasible and reliable key management and maintenance is also a problem. The issue becomes more challenging in big organizations with many users, where the number of keys involved is in the order of the square of the user number.

In most ciphers, keys can be used to encrypt plaintexts of much more bits for secure communications, so that dynamic key can be a choice and the shared key can be securely refreshed timely during communications using partial encrypting capacity, realizing practical one-time key cipher to avoid corresponding attacks. Dynamic key cipher is suitable for real-time communications, and is less advantageous in key management and data storage scenarios however.

Therefore, a method that can high-efficiently generate and securely distribute any number of keys needed with well-defined key strength will overcome the limitations of current available ciphers and solve the fundamental problem of cryptography.

In this paper, we designed a key generation and secure distribution system that fulfills the above requirements from basic principles, and constructed robust, universal, and reliable information security systems which is provably secure based on information theory, including encryption systems, key management system, and document management system. Our work may provide a solution on symmetrical cryptography.

Results

System design

Imagine a system which can generate and securely distribute unlimited number of keys from a master key with the key strength of the system can be equal to the bit-count of the master key.

We first set 3 prerequisites for security of the imagined system.

Prerequisite 1: Master key keeps confidential in normal system operations, i.e. the confidentiality of the master key cannot be compromised from all the outputted information from the system, including keys and information for key distribution.

Prerequisite 2: Outputted keys are independent from each other so that new generated keys cannot be deduced from all the outputted keys using known methods and public available information.

Prerequisite 3: Keys can be securely distributed between legitimate communicators exclusively sharing the master key, i.e. keys can be and can only be distributed between legitimate communicators through public information channel, and it is impossible for an adversary to get meaningful information of the key from the distribution processes without the master key.

Obviously, systems that fulfill the 3 prerequisites can generate keys from a master key and securely distribute the keys between legitimate communicators exclusively sharing the master key, and the key strength of the system can be equal to the bit-count of the master key.

Following the 3 prerequisites, we design a key generation and distribution system.

As shown in Fig. 1, system stores master key and at least one type of one-way data conversion algorithm (OWDCA) as system information. The master key is an undisclosed random bit sequence. OWDCA can easily convert input messages into secondary messages deterministically, while the reverse process to convert the secondary messages to the original input messages is impossible or is hard in calculation.

For key generation and distribution, we introduce timestamps which can be arbitrarily unrepeated messages. Messages encoding time information on encryption can be used as timestamps to ensure opening value space of the timestamp.

Scheme 1 shows typical key generation and distribution process:

- Generate unrepeated arbitrary message as timestamp.
- Combine the information from the timestamp and master key to generate medium key.
- Use the medium key as input message, and convert the medium key into secondary message by OWDCA.
- Extract bits from the secondary message to generate key, and tag the key with corresponding timestamp.
- Send the timestamp to the paired key generation and distribution system.
- The paired system recovers the key from the timestamp following the same steps using the same system information.

We then define 4 criterions for a qualified OWDCA (q-OWDCA) that can be used in systems based on Scheme 1 for sustainable key generation and secure distribution.

Criterion 1: It is easy to convert input messages into secondary messages deterministically.

Criterion 2: It is impossible to deduce the original input message from the secondary message generated.

Criterion 3: It is nondegenerate and nonlinear data conversion from input messages to secondary messages, i.e. the value space of the secondary messages is no smaller than that of the input messages to ensure different secondary messages can be generated from different input messages statistically, and the smallest deviation of input message causes global change of the resulted secondary message and the change cannot be predicted from the deviation. In strict nonlinear conversions, single bit deviation in input message causes global change of secondary message generated.

Criterion 4: Any information in required format is valid input message. For N-bit input messages with value space of 2^N, any input message with value between 0 to 2^N can be converted into corresponding secondary message by OWDCA.

We will prove that a system based on Scheme 1 fulfills the 3 prerequisites when q-OWDCA satisfying the 4 criterions is used.

Information outputted from the system which can be used to probe the master key is no more than timestamps and keys. Using q-OWDCA satisfying criterion 2, medium keys cannot be deduced from the secondary messages generated, therefore, even if all the outputted keys are disclosed after being used, partial or even whole set of secondary messages obtained from these disclosed keys cannot be used to deduce any medium keys to compromise confidentiality of the master key. Use arbitrary messages, timestamps can be general information independent from method used thus disclosing no master key and other system information. Therefore, master key can be kept confidential in normal system operations, and prerequisite 1 is fulfilled.

In the system, system information keeps unchanged, and variable outputted keys are determined by variable timestamps. Using arbitrary messages, the value space of the timestamp is opening and the

timestamps can contain independent information. Using nondegenerate combination of master key and timestamp to generate medium key, the medium key can inherit independent information from the timestamp. Using q-OWDCA satisfying criterion 3 and 4, any medium key generated can be valid input message and can be used to generate secondary message that inherits independent information from the medium key, so that independent keys can be extracted from the secondary message. Prerequisite 2 is fulfilled.

Using nondegenerate combination of master key and timestamps to produce medium keys, the medium keys can inherit the key strength of the master key, so that short timestamps can be used to generate independent keys with stronger key strength determined by the master key.

If needed, timestamps with bit-count equal to the master key can be used to produce medium keys, secondary messages, and keys following the steps in Scheme 1. Longer timestamps will not reduce the efficiency of the system significantly when set secondary messages much longer than the master and medium keys.

Obviously, keys generated from the system are readily distributed by corresponding timestamps between legitimate communicators sharing the master key. Use q-OWDCA satisfying criterion 3, single bit modification in medium keys will cause unpredictably global change of the corresponding secondary messages, so that secondary messages cannot be deduced from the deviations of the medium keys, and the resulted keys cannot be probed from unrepeated timestamps without the master key. We can conclude that keys generated from the system can be securely distributed by corresponding timestamps. Prerequisite 3 is fulfilled.

Opening value space of timestamps facilitates generating unlimited number of medium keys from a master key, and accordingly, generating any number of keys needed which can be securely distributed by corresponding timestamps.

Therefore, follow scheme 1 and use q-OWDCA, make sure master key does not disclosed through means other than normal system operations, unlimited number of independent keys can be generated from a single master key, the keys can be securely distributed by corresponding timestamps between legitimate communicators exclusively sharing the master key, and the key strength of the system can be equal to the bit-count of the master key.

Some qualified one-way data conversion algorithms

Qualified one-way data conversion algorithms that satisfy the 4 criterions are common, as will be described in this section.

Sqrt algorithm

Sqrt (square root) algorithm, or other algorithms that can produce irrational numbers, can be used to generate secondary messages of required length from input messages.

Typical scheme for sqrt algorithm:

- Digitalize input message and convert the digital information into nonnegative integer N.
- Calculate $\sqrt{N}$ to certain precision so that the fraction part contains enough digits, discard high-order digits with same length of N and extract digit sequence ‘a’ with required length thereafter.
- Calculate $\sqrt{(N+10)}$ and extract digit sequence ‘b’ in the same way.
- Perform digit-wise modular addition of ‘a’ and ‘b’ to generate secondary message.

The calculation is demonstrated using 1-8-8-9 as input message, using decimal number, N=1889, and 24-digit secondary messages. N will be a very big number in practical encryption scenarios and secondary message will be also much longer.

Calculate $\sqrt{1889}$ = 43.462627624201461756701166968 08..., discard 4 high-order digits, get 26276242014617567011669680 8, extract 24 digits, get 262762420146175670116696 as ‘a’.

Calculate $\sqrt{(1889+10)}$ =43.5775171390018502049862735363 1..., and get 75171390018502 0498627353 as ‘b’ in the same way.

Generate secondary message by digit-wise modular addition of ‘a’ and ‘b’ using modulus 10.

a：262762420146175670116696

b：751713900185020498627353

Modular addition with modulus10

Secondary message: 913475320221195068733949

In modular addition with modulus 10, digit-wise addition of ‘a’ and ‘b’, if the sum is smaller than modulus 10, the result is the sum, if the sum is bigger than 10, subtract 10 or multiple 10s from the sum until reach a number between 0 and 9 as the result. The calculation can expand to modular additions using other value of modulus.

We will prove that sqrt algorithm is a q-OWDCA that satisfies the 4 criterions.

Perform sqrt operation on an input nonnegative number can produce a secondary message deterministically, criterion 1 is satisfied.

Any sequence with the same format can be used as sequence ‘a’ or ‘b’ and perform modular addition with matched sequence to generate the target secondary message. For example, the third digit in the secondary message is 3, and the third digit of ‘a’ and ‘b’ can be any value between 0 and 9, by modular addition with matched pairs, such as 0 and 3, 1 and 2, 2and 1, 3 and 0, 4 and 9, 5 and 8, 6 and 7, 7 and 6, 8 and 5, 9 and 4, to produce target value 3, same as digits from other positions. Therefore, deduce the

value of the original sequence ‘a’ or ‘b’ from the secondary message generated is impossible, same as that of N and the input message. Criterion 2 is satisfied.

The calculation makes use of the fact that N and N+10 cannot be complete square number at the same time so that any length of nonzero secondary message needed can be generated from an input message using sqrt algorithm to ensure a nondegenerate conversion from input messages to secondary messages. By discarding high-order digits with same length of the input number, single bit modification of input message causes global change of ‘a’ and ‘b’ as well as the resulted secondary message, making strict nonlinear data conversion from input messages to secondary messages. Criterion 3 is satisfied.

Any input message can be digitalized and formatted into a defined nonnegative integer falling into the range of the value space of the message and can be valid input. Criterion 4 is satisfied.

Division algorithm

Using public data, high-efficient q-OWDCA can be constructed using division algorithm.

A typical division algorithm scheme:

- Digitalize input message in the form of $x_1x_2x_3…x_{i-2}x_{i-1}x_i$-$y_1y_2y_3…y_{i-2}y_{i-1}y_i$.
- Using public data in the form of $z_1z_2z_3…z_{k-2}z_{k-1}z_k$, where k is much bigger than 2*i.
- Calculate $1z_1z_2z_3…z_{k-2}z_{k-1}z_k1/1x_1x_2x_3…x_{i-2}x_{i-1}x_i1$ to reach enough significant figure, discard high-order digits equal to the length of the input message, and extract digits thereafter to generate sequence ‘a’ with the same length as that of the public data.
- Calculate $1z_1z_2z_3…z_{k-2}z_{k-1}z_k1/1y_1y_2y_3…y_{i-2}y_{i-1}y_i1$ to generate sequence ‘b’ in the same way.
- Perform digit-wise modular addition of ‘a’ and ‘b’ to generate secondary message.

The calculation is demonstrated using input message 3-7-2-8-1-5-9-6, using decimal number, the formatted input message is 3728-1596, and public data is a 24-digit decimal number 367368971209437083569112. Input message, secondary message and public data will be much longer in practical encryption scenarios.

Calculate 1367368971209437083569112 1/137281=9960365754980201 8019.180520246793, discard 8 high-order digits and get 549802018019.180520246793, extract 24-digit sequence 549802018019180520246793 as sequence ‘a’.

Calculate 1367368971209437083569112 1/115961=1179162797155454923 26.65396986918, and get 971554549232665396986918 as sequence ‘b’ in the same way.

Generate secondary message by digit-wise modular addition of ‘a’ and ‘b’ using modulus 10.

a: 549802018019180520246793

b: 971554549232665396986918

Modular addition using modulus10

Secondary message: 4103565572417458161226O1

We can prove that division algorithm is a q-OWDCA by the same way as that in the sqrt algorithm, and the course of the proof is omitted.

Data processing speed in division algorithm is in inverse proportion with the bit-count of the key, same as the fastest encryption algorithm in use.

Message combination and modular addition

We provided a more efficient q-OWDCA based on message combination and modular addition (MCMA).

A public database is added in the system information beside master key and OWDCA. The public database contains N sequences (sequences of number) of the same length, and each sequence is identified with a unique ID number (ID) between 0 to N-1 and serves as seed so that all seeds have the same bit-count. Accordingly, input message is a sequence of M IDs and points to M corresponding seeds.

For MCMA, select M seeds with IDs defined in the input message sequentially, modular addition of these seeds to generate a secondary message with the same format.

We will prove that MCMA algorithm is a q-OWDCA satisfying the 4 criterions.

It is easy to generate a secondary message deterministically by selecting M seeds with IDs defined in input message and performing modular addition, criterion 1 is satisfied.

Using modulation addition to generate secondary messages, any sequence with the same format as that of the secondary message can be combined with matched sequence to produce a target secondary message, therefore, in order to deduce the original input message from secondary message generated, any seeds in the public database cannot be excluded before testing all the seed combinations. The different combinations of IDs in input messages, or possible combinations of the seeds to be tested, are equal to the value space of the input message and are unbiased distributed, therefore, deduce the original input message from the secondary message generated is no easier than random guess the input message itself, which means that input messages cannot be deduced from corresponding secondary messages. Criterion 2 is satisfied.

It is easy to set seeds and secondary messages with length significantly longer than that of input messages to ensure nondegenerate conversion from input messages to secondary messages. Because input message is sequence of seed IDs, single bit modification in input message changes an ID and results in different seeds in modular addition. Since the secondary message has the same length as that

of the seeds, change a seed in modular addition will cause global change of the resulted secondary message, making MCMA a strict nonlinear OWDCA. Criterion 3 is satisfied.

Obviously, any input message can be digitalized and formatted into ID sequence and can be valid input message. Criterion 4 is satisfied.

MCMA algorithm is a strict nonlinear OWDCA when keep seeds confidential. Using public data as seeds, additional OWDCA should be introduced in the combination of the master key and timestamps to make sure single bit modification of timestamps will cause global and unpredictable change of the resulted medium keys, therefore secondary messages and keys cannot be deduced from the deviation of the timestamp to satisfy strict nonlinear one-way data conversion from timestamp to secondary message in MCMA algorithm using public database.

Data processing efficiency can be significantly increased in MCMA algorithm using more seeds and the efficiency can be N times higher than that of division algorithm when 2^N seeds are used. For example, in a 256 bits key generation system, 256 times of addition is needed to produce a key using division algorithm. Using MCMA algorithm with a public database containing $256=2^8$ seeds with 8-bit IDs, 32 times of modular addition are required to produce a key, 8 times more efficient than division algorithm.

In MCMA algorithm, public database can be a ring sequence containing N*M integers which is divided into N units of M-integer, and each unit is identified by a unique ID from 0 to N-1. Seed starts from a unit, expands in the same direction, spans the whole ring sequence, forms a sequence with length of N*M, and uses the ID of the start unit as corresponding seed ID. N seeds with length of N*M can be derived from a ring sequence containing N*M integers.

Fig. 2 depicts a typical scheme for MCMA algorithm. Public database is a 64-bit binary ring sequence which is divided into 16 4-bit units with 4-bit IDs from 0000 to 1111. The ring sequence will be much longer in practical encryption scenarios.

Sixteen seeds are derived from the binary ring sequence, and each seed starts from a unit, expands in the same direction, spans the whole ring sequence, forms a 64-bit binary sequence with the same length as the ring sequence, and uses the ID of the start unit as corresponding seed ID.

For example, seed 0101 starts from unit 0101 which is ‘1001’, expands forwardly to unit 1111 which is ‘1100’, passes unit 1111 and connects with unit 0000, ends at unit 0100 which is ‘0110’, and forms a 64-bit binary sequence.

Input message is a 16-bit binary sequence, containing 4 seed IDs, 1010, 0110, 0111, 0101, sequentially.

Typical MCMA scheme:

Select seeds with IDs defined in the input message sequentially, 1010, 0110, 0111, 0101, tag the selected seeds with number 0, 1^{st}, 2^{nd}, 3^{rd} respectively as shown in Fig. 2.

Perform sequence shift operation on the selected seeds before modular addition. As shown in Fig. 2, selected seeds 0 (ID 1010) is kept unchanged, seeds 1st (ID 0110) is shifted 1 digit forwardly, and the last digit fold back to maintain the length of the seeds. Seed 2nd and 3rdare shifted 2 and 3 digits respectively in the same way, and fold back digits are underlined.

Modular addition of the shifted seed messages to produce secondary message. Using binary sequences, modular addition becomes XOR operation.

The shift ensures input messages pointing to same set of seeds with different orders can generate different secondary messages thus establishing a one to one mapping between input messages and secondary messages statistically.

In MCMA algorithm, input message 0000-0000-0000-0000 with simplest pattern points to 4 mutants of seed 0000, equivalent to 4 different sequences, and has the same complicity in calculation as all other input messages. Therefore, all the input messages have the same complicity in MCMA algorithm as long as the format of the input messages is defined.

Public data conversion under the control of input message

Based on common features of division algorithm and MCMA algorithm, we introduce public data conversion under the control of input message (PDCUCIM) as a class of q-OWDCA. A public database is added to the system information beside master key and OWDCA. As public data, strict randomness is not a necessity, and numbers generated from a random number generator in the tool box of a common computer can be used.

PDCUCIM algorithm is straightforward: convert the public data into secondary message under the control of an input message.

Besides division algorithm and MCMA algorithm, PDCUCIM algorithm can be constructed from many established encryption algorithms. The basic function of a conventional encryption algorithm such as DES, AES, is to convert plaintexts into ciphertexts under the control of a key, and the bit-count of the ciphertexts and plaintexts is usually much bigger than that of the key.

Using key as input message, plaintext as public data, and ciphertext as secondary message, common encryption algorithm are q-OWDCA satisfying the 4 criterions, as will be proved below.

The basic function of an encryption algorithm is easily converting plaintexts into defined ciphertexts under the control of a key, it means that the encryption algorithm can easily convert the public data (plaintext) into corresponding secondary message (ciphertext) deterministically under the control of input message (key), criterion 1 is satisfied.

Encryption algorithms usually use the same key to encrypt different plaintexts, so a valid encryption algorithm should satisfy the requirements that it is impossible to deduce key from corresponding

ciphertexts and plaintexts, and the complexity of possible deduction algorithm defines the key strength of the cipher. Criterion 2 is satisfied.

It is common for most encryption algorithms to encrypt plaintext with much more bit-count than that of the key to generate ciphertext with even bigger bit-count, making a nondegenerate conversion from key to ciphertext. A valid encryption algorithm should satisfy nonlinear conversion from key to ciphertext to make sure minimal deviation in the key can cause global change of the corresponding ciphertext and the change cannot be predicted from the deviation. Criterion 3 is satisfied.

In most encryption algorithms except public key encryption algorithms, any information that can be converted into required format can be used as key and can be valid input message in PDCUCIM. Criterion 4 is satisfied.

Different from application scenarios for most ciphers, medium key is one-timely used to convert public data into secondary message in our system, so that attack strategies against common encryption algorithms will be unsuccessful, and many encryption algorithms can be candidates for PDCUCIM. Additionally, it is not required to convert secondary messages back into original public data in our system, so that many one-way algorithms which are not suitable for common encryption algorithms can be candidates for q-OWDCA.

Applications

Key generation and secure distribution are central issues in cryptography. Methods reported here, including schemes for key generation and distribution as well as q-OWDCAs designed for these schemes, providing a facile solution on the critical problems. Various cryptosystems that are provably secure based on information theory can be constructed from above methods, such as encryption systems, key management systems, and document management systems, which will be discussed in this section.

Encryption system

Based on Shannon's theory, unconditional secure cipher can be realized by encrypting plaintexts with undisclosed one-time keys of the same bit-count, this encourages us to design cipher by using one-time keys generated from the above key generation and secure distribution system *(1)*. In this section, we construct a universal and high-efficient cipher based on Scheme 1 using MCMA as OWDCA. Unlimited encryption-decryption with 1024-bit key strength can be realized by 65 times of modular addition using 1024-bit master key.

A feasible key generation and distribution scheme is shown in Fig. 3. The scheme becomes concise and compact by introducing private data for medium key generation, and the system information include master key, public database, and OWDCA 1, 2, and 3.

Public database contains a binary ring sequence of 512 Kb which is divided into $65536=2^{16}$ 64-bit units with 16-bit unit IDs. Seed starts from a unit, expands and spans the whole ring sequence, forms a 512 Kb

binary sequence and uses ID of the start unit as corresponding seed ID. Totally 65536 512 Kb seeds can be derived from the 512 Kb public database with 16-bit seed IDs.

Master key is a 1024-bit undisclosed binary sequence, including 64 16-bit IDs sequentially and pointing to 64 seeds.

For OWDCA 1, select 64 seeds with IDs defined in the master key sequentially from the public database, and perform shifted modular addition of the selected seeds similar as shown in Fig. 2 to generate 512 Kb private data.

The private data is divided into $4096=2^{12}$ 1024-bit fragments with 12-bit fragment IDs. Private data exist exclusively in encryption-decryption processes and are temporarily stored in RAM (random access memory) or Cache memory of the computer.

48-bit timestamp is used, including 4 12-bit IDs and pointing to 4 fragments in the private database. The 48-bit timestamp can be allocated 8-bit for year, 4-bit for month, 5-bit for day, 5-bit for hour (using 24 hour a day), 6-bit for minute and 6-bit for second, totally 34 bits for encoding the time information, and 14 bits redundancy. The redundancy can be filled with random bits to maintain fixed format of the timestamp, and it can be used for expansion in need, for example, 10-bit can be allocated in time scale bellow second to increase the time resolution of the system to millisecond in high speed cryptosystem, and additional 4-bit can be allocated for year to increase the time span of the system to $2^{12}=4096$ year.

For OWDCA2, generate a 48-bit timestamp according to encryption time, select 4 fragments from the private database with IDs defined in the timestamp sequentially, and perform shifted modular addition to generate a 1024-bit medium key. Medium keys exist exclusively in RAM or Cache memory of the computer.

The 1024-bit medium key is in the same format as that of the master key and OWDCA 3 is the same as OWDCA 1: select 64 seeds with IDs defined in the medium key sequentially and perform shifted modular addition to generate a 512 Kb secondary message.

Extract bits with required length from the secondary message to generate a key, the maximum length can be 512 Kb one-time key including the whole secondary message, also, 1024-bit key with the same length of the master key can be generated.

Private data are calculated only once and are stored in Cache memory temporarily and used in the whole encryption-decryption cycle to produce all the keys needed. In applications that encryption-decryption speed is a matter, such as secure real time communications and encryption-decryption of large data files, the added workloads for private data generation have negligible effect on the performance of the cipher.

The key strength of the system is 1024-bit if the master key is not exposed by means other than necessary processes for key generation and distribution using the timestamp.

In the system, 64 times of modular addition, or XOR operations, are required to produce keys with 1024-bit key strength. Four times of modular addition of 48-bit input message to generate medium key in OWDCA2 should be added in the total workload of a 512 Kb one-time key generation, the ratio added is (4x48)/(512x8x64) =0.073%, can be considered negligible.

Provably secure cipher based on information theory can be constructed by using secondary messages generated from the above scheme as one-time keys to encrypt plaintexts of the same format by modular addition to produce ciphertexts, exchanging the ciphertexts between legitimate communicators exclusively sharing a master key, and using corresponding timestamps as clue for decryption. For the above cipher with 1024-bit key strength, encryption or decryption costs 65 times of modular addition, 64 times for one-time key generation and one time for generating ciphertexts from plaintexts or recovering plaintexts from ciphertexts.

The cipher can be feasibly implemented high-efficiently using routine portable communication devices. For example, data processing speed is 15 Mbps (Mb per second) for ciphertext of 1024-bit key strength using a smart phone with 1 GHz processor, and the 512-Kb public database can be easily stored in regular communication devices.

Parameters for public database and algorithms in the cipher can be adapted for different applications. Public database can be a 4Kb ring sequence including 256 128-bit units to derive 256 4Kb seeds with 8-bit seed IDs. Key of 1024-bit key strength can be generated by 128 times of modular addition, one-time key length is 4Kb, and 129 times of modular addition are required in encryption or decryption. For current mainstream ciphers with 256-bit key strength, 32 times of modular addition are required for key generation, and 33 times for encryption or decryption. Also, 2048-bit or stronger keys can be feasibly realized in above system.

Key management system

Current methods use arbitrary messages (timestamps) to generate keys from a master key, the keys cannot be deduced from the timestamps without the master key, and the confidentiality of the master key is not compromised from all the generated keys and their corresponding timestamps, providing a feasible key management strategy. Using user IDs as timestamps, current reported methods can be used to construct robust and reliable key management systems.

System generates and stores master key and OWDCAs as system information. System generates user ID for each user, and the user ID can be arbitrary message, i.e. user feature information plus random bits. System stores user IDs as part of system information. The solely privacy of the system is a master key and all other information can be set public accessible for easily handling.

System uses user IDs as timestamps to generate corresponding user keys following steps in Scheme 1, and sends user IDs and user keys to corresponding users, realizing key distribution.

System uses safe manner to distribute keys to users, for example, user keys can be stored in mediums such as discs, USB disks or print matters, marking the mediums with corresponding user IDs and delivering the mediums to corresponding users. Users can also obtain user keys and user IDs from system manager directly.

In case user key is lost, user submits a query to the system. System manager may ask the user to submit user ID and verify the user ID based on the stored information, and use the verified user ID as timestamp to generate corresponding user key and return the key to the user, restoring the user key.

System can combine different user IDs to generate a timestamp and uses the timestamp to generate key for secure communications between corresponding users. In this way, system can easily manage keys with number much larger than the number of the users, providing convenient and reliable key management service for big organizations with large number of users.

The cost for system running and maintenance can be greatly reduced by leaving all other system information including user IDs public accessible and keeps the master key as the solely system privacy. Use public accessible user IDs to generate and restore user keys can effectively prevent from losing user key permanently and enhance reliability of the key management system.

In the following key management system, a communicator can securely and conveniently manage all related keys using a master key.

System generates arbitrary message as key IDs which are used as timestamps to generate stem keys, and stores the stem keys in communication device for certain applications in certain period of times. The key IDs are stored safely for retrieving the corresponding stem keys. For example, a stem key used in 2021 can use the year number 2021 as key ID.

Stem key is used as master key of subsystem, or application layer key management system, to manage user keys for different communication partners.

The subsystem generates partner ID for each communication partner. We suggest set permanent or fixed partner ID for each communicator, and partner ID can include feature information of a communicator that is suitable to be made public accessible.

The subsystem uses partner IDs as timestamps to generate corresponding partner keys from the stem key, and passes the partner keys to corresponding communication partners face to face or via other safe manners. The partner stores the partner key in the e-card under the name of the sender and the e-card also contains the sender’s partner ID. Communication partners establish secure connection by exchanging partner keys with each other.

Secure communications can be realized between connected communicator A and B. Communicator A generates communicator B’s partner key A-B, and combines with partner key B-A stored in the e-card of communicator B to generate key AB. Communicator B generates key BA in the same manner. Using

symmetric combination, key AB and key BA are the same, which is used as communication master key to generate one-time keys for encrypting messages in communications, realizing secure communications.

By combining both communicators' partner keys as master key for communications, messages from unintended senders can be effectively blocked. Additionally, in case one communicator lost partner key, he can use partner key generated by his own side as half master key for communications to contact his partner urgently and restores the key securely.

Hierarchy key management system is formed from the above setups where a communicator uses a master key to generate stem keys, and uses the stem keys as master keys of subsystems for certain applications, such as managing partner keys for routine communications. Using hierarchy key management system, stem keys can be generated in secure circumstances and are used for practical applications to avoid master key exposure thereby enhancing the system security. A stem key can be stop to use when it is unintended disclosed to restraint the range and impact of the disclosure. In a hierarchy key management system, keys are generated and managed by direct sub-manager with certain rank and authority, enabling easily tracing the responsibilities of all members in the system under the control of a single master key, forming a concise and compact key management system with highly ordered structure.

It is a challenge for key generation, secure distribution and maintenance in large complex information security systems. High-ordered hierarchy key management system can be constructed based on the above setups with certain modification and expansion by introducing multiple layers between system manager and terminal users, so that it can be used to manage keys for large and complex organizations high-efficiently and reliably using a single system master key.

The hierarchy key management system can be feasibly expanded so that users can include all persons and their connected belongings worldwide. For key management, if needed, the system can manage keys of all users worldwide with a single master key using a personal computer or small server. For user key service and maintenance, several independent and competitive providers can be elected to distribute keys for all users worldwide, and user combines keys distributed from these providers as user master key. In such a manner, the user master keys are confidential unless all these providers are colluded, thus achieving confidentiality and reliability at the same time for users' master keys.

Document management system

Document management system can be constructed by combining the encryption system and key management system, which can be used to securely manage documentations and databases for big organizations with many different types of users using a single system master key.

The system includes administrative terminus (administrator) and user terminus (users).

The administrative terminus can be big data management systems such as cloud storage systems, cloud computing systems, bank systems, and transportation control systems, which manage sensitive information involving different types of users. It can be small systems such as personal information

management system, to manage information for personal routine communications. Administrative terminus includes: key management module, using key management system described in section 4-2, for distribution and maintenance of user master keys and user IDs; system encryption module, using encryption system described in section 4-1, for encrypting/decrypting files and messages communicated between administrator and users; and storage module, for storing documents.

Key management module generates user ID for each user, and uses user IDs as timestamps to generate corresponding user master keys from system master key, and distributes user IDs and user master keys to corresponding users.

User terminus includes each user of the system, and users can be persons or any belongings connected in internet of things. User encryption module is installed in communication devices for each user, using encryption module same as the system encryption module with unique user master key distributed by administrator to each user for encrypting/decrypting files and messages communicated between each user and administrator independently.

User encryption module generates one-time keys from user master key, encrypts files or messages to generate main ciphertexts, uses corresponding timestamp and user ID as title, combines the title and the main ciphertext to generate ciphertext, and sends the ciphertext to administrator through public information channel as proofs for documents submission.

Administrative terminus analyzes the ciphertext using system encryption module to extract main ciphertext as well as timestamp and user ID from the title. Key management module generates user master key according to the user ID from system master key. System encryption module generates decryption key according to the timestamp from the user master key, decrypts the main ciphertext, confirms the submission and stores the document in storage module.

Administrative terminus can also generate user master key for target user using key management module, generate one-time keys from the user master keys to encrypt files using system encryption module, and send resulted ciphertext to the target user. The target user receives and decrypts the ciphertext and recovers document using user encryption module.

Various document management systems can be constructed.

Administrative terminus can be a manager of an organization or company, users can be employees or members, and documents can be work reports from employees or members. Each member or employee is assigned a user ID and corresponding user master key on enrollment. Using the document management system, manager can securely and reliably exchange working information with multiple employees independently using a single system master key.

Administrative terminus can be a publisher or patent office, users can be authors or applicants, and documents can be manuscript for publication or patent application materials. Each user receives user ID and corresponding user master key on registration. Using the document management system,

administrator can securely and reliably handle applications from multiple highly dynamic users independently using a single system master key.

Hierarchy document management system can be constructed by setting nodes of different layers between system administrator and terminal users using hierarchy key management system as key management module, so that administrator can securely and conveniently manage documents for a big and complex organization involving large number of highly dynamic users with complex grouping using a system master key.

The document management system can find applications in the emerging area of internet of things. All connected personal belongings such as entrance guards, air conditioners, cars, monitors, smart home appliances, as well as data and files stored in cloud disc or other internet servers, can be users. Encryption module is installed in each belonging and assigned a user ID and corresponding user master key as soon as it is put in use. Owners can conveniently and reliably connect and control multiple belongings with various properties by secure communication with each belonging independently using a single master key.

Discussion

We designed key generation and secure distribution systems based on basic principles and constructed universal cryptosystems which are provably secure based on information theory.

We first propose a system which can generate and distribute keys from a single master key. We set 3 prerequisites to make sure that the proposed system can sustainably generate and securely distribute keys with key strength of the system equal to the bit-count of the master key. 1) The master key keeps confidentially in normal system operations. 2) New keys cannot be deduced from all the outputted keys and known information and knowledge. 3) Keys can be securely distributed.

Based on the 3 prerequisites, we designed a scheme for key generation and distribution: generate unrepeated arbitrary message as timestamp, combine master key and timestamp to generate medium key, convert medium key into secondary message using one-way data conversion algorithm, extract information from the secondary message to generate key, tag the key with corresponding timestamp, and send the timestamp to paired system. The paired system uses the timestamps to generate corresponding keys from the same master key following the same steps, recovers the keys and realizes secure key distribution.

We further defined 4 criterions for a qualified one-way data conversion algorithm that can be used in the scheme: 1) Convert input messages into corresponding secondary messages is easy and deterministic. 2) Convert secondary messages into the original input messages is impossible or hard in calculation. 3) It is nondegenerate and nonlinear conversion from input message to secondary message. 4) Any message that can be formatted into the required format is valid input message and can be converted into corresponding secondary message.

We proved that a system based on the above scheme and using the qualified one-way data conversion algorithm could fulfill the 3 prerequisites in normal key generation and distribution processes. Therefore, use unrepeated arbitrary messages as timestamps and at least one type of one-way data conversion algorithm, any number of independent keys needed can be generated from a master key and securely distributed by corresponding timestamps between legitimate communicators exclusively sharing the master key, and the key strength of the system can be equal to the bit-count of the master key.

We designed some feasible q-OWDCAs including sqrt algorithm, division algorithm, MCMA algorithm, and PDCUCIM algorithm, and provided preliminarily proofs that these algorithms satisfied the 4 criterions. Among these algorithms, the performance of MCMA algorithm surpasses those of the fastest encryption algorithms in use.

We described several typical cryptosystems which are provably secure based on information theory using the above methods, including encryption system, key management system, as well as document management system, and demonstrated that a single master key can be used to realize various information security purposes, such as high-efficiently sustainable secure communications, reliable key management and system maintenance for big and complex organizations, and secure management of large-scale and complex database or file systems.

It is an advantage that our method is derived from basic principles without involving abstruse mathematical problems, while it also brings some weakness at the same time. The 4 criterions we defined for a qualified one-way data conversion algorithm are mainly in view of cryptographic applications and may lack strict completeness in mathematics and logics. For example, the nonlinear condition in criterion 3 and calculatingly irreversible condition in criterion 2 may be overlapped or be the same problem. We proved preliminarily that the reported algorithms are q-OWDCAs satisfying the 4 criterions, comprehensive mathematical argumentations are required to finally justify our proofs however. Nonetheless, for our main concerns on cryptographic applications with certain key strength, we believe these preliminary proofs can justify the validity of our methods. Furthermore, the scheme is highly modular and adaptive and can be adjusted or expanded to overcome possible problems encountered.

In conclusion, we designed key generation and secure distribution systems from basic principles and constructed robust, universal and reliable cryptosystems which are provably secure based on information theory. Use a single master key as the solely privacy of the system, the reported methods can be used to achieve information security for large-scale complex organizations with different types of users. Our work provided a facile solution on key generation and secure distribution problem in cryptography and may solve the fundamental problem in symmetric encryption algorithm. Future works may be required to adjust and expand the scheme for better applications, and strict and comprehensive mathematical argumentations for specific algorithm may be also needed.

References

1. C. E. Shannon, Communication Theory of Secrecy Systems. *Bell System Technical Journal*, **28**,656 (1949).

Figures

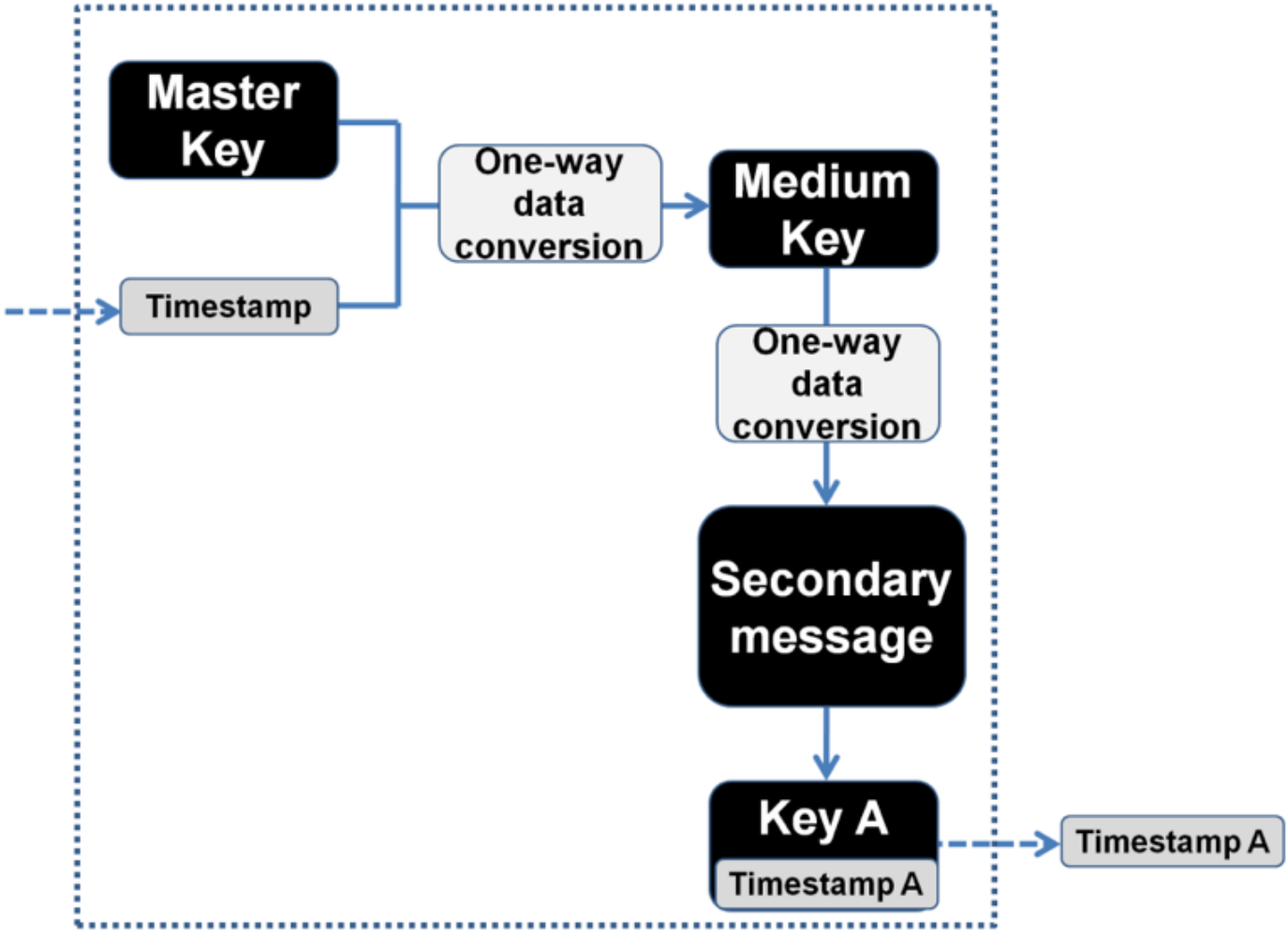

Figure 1

Scheme for key generation and distribution. Solid line: key generation, dash-line: key distribution

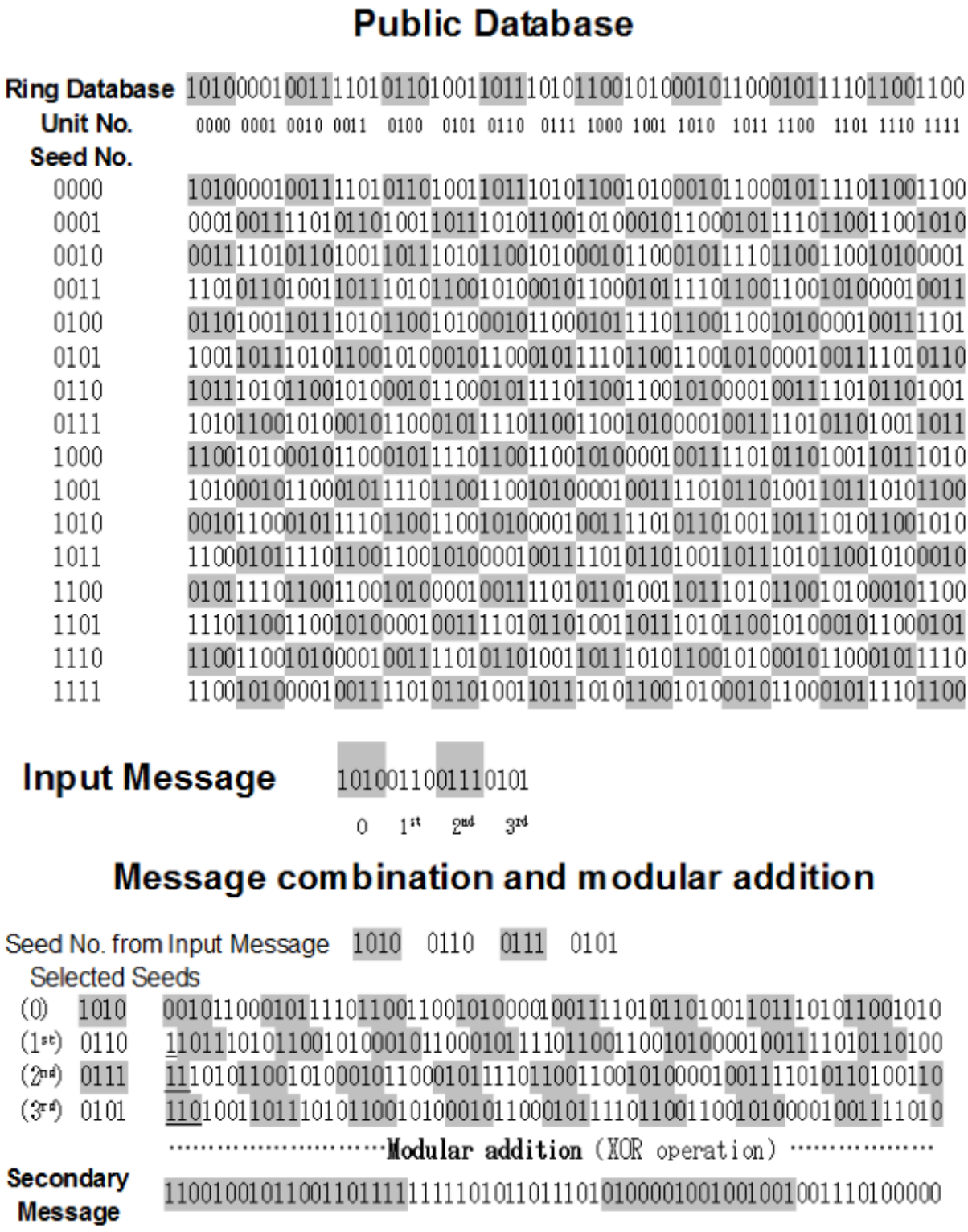

Figure 2

Message combination and modular addition

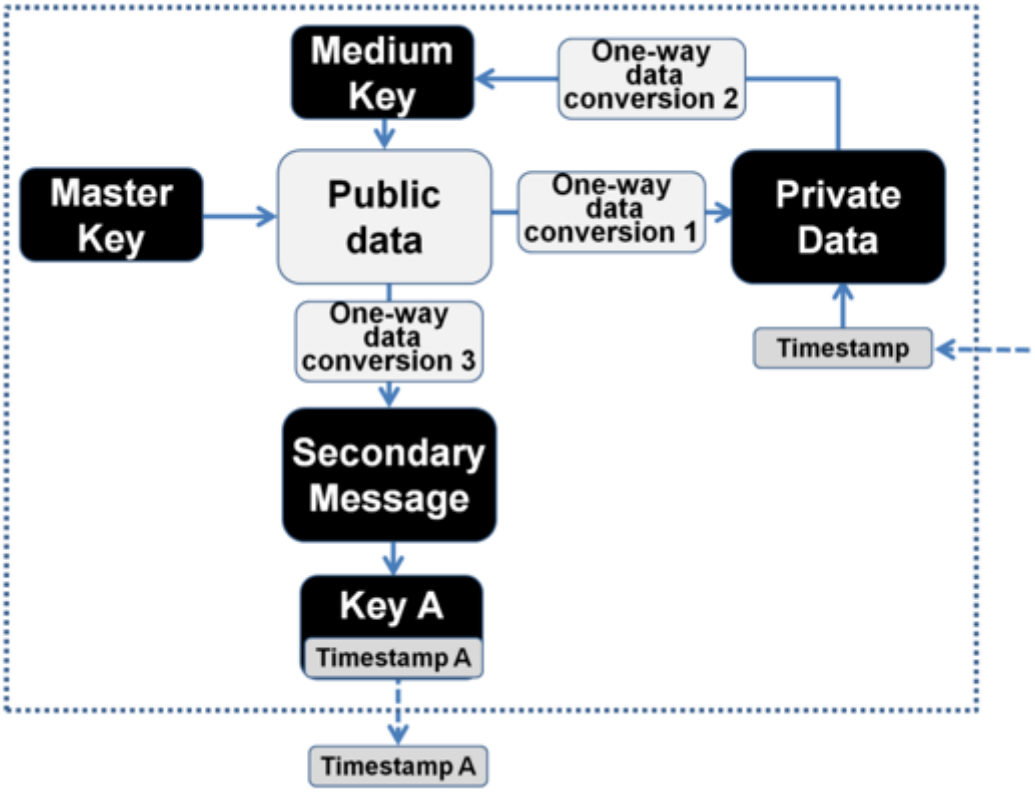

Figure 3

Scheme for key generation and distribution using private data. Solid line: key generation, dash-line: key distribution