

METHOD AND SYSTEM FOR ENCRYPTING DATA

5 The present invention relates to a method for encrypting data based on an all-or-nothing encryption.

10 The present invention further relates to a system for encrypting data based on an all-or-nothing encryption, preferably in form of a computer, one or more processors and/or security modules, preferably for performing with a method according to one of the claims 1-12.

Although applicable to security in general the present invention will be described with regard to cloud security.

15 Cloud security is gaining more and more importance in many applications and services nowadays. One of the important techniques that can be used to strengthen confidentiality of data stored in the cloud is the so-called all-or-nothing encryption. All-or-nothing encryption provides semantic security of data while
20 guaranteeing that the data can only be recovered if and only if all blocks of a ciphertext are available for download by or known to a given client. Therefore all-or-nothing encryption does not solely rely on the secrecy of the encryption key for the data: In order to acquire any meaningful information of the input plaintext it is required that any adversary has access to all the data or blocks of the ciphertext respectively. Therefore all-or-nothing encryption ensures a transparent key
25 management process and naturally complement information dispersal techniques that can be used to efficiently store the data in a distributed storage like cloud storage.

30 Conventional all-or-nothing encryptions are for example disclosed in the non-patent literature of R. Rivest, "All-or-Nothing Encryption and The Package Transform", In Proceedings of Fast Software Encryption, pages 210-218, 1997, or in the non-patent literature of Victor Boyko, "On the Security Properties of OAEP as an All-or-Nothing Transform", CRYPTO 1999: 503-518. 1998.

Another conventional all-or-nothing-encryption is disclosed in the non-patent literature of D. R. Stinson, "Something About All or Nothing (Transforms)", In Designs, Codes and Cryptography, pages 133-138, 2001.

5 Most of these conventional techniques add an encryption layer after an all-or-nothing transformation of the all-or-nothing encryption. Therefore conventional all-or-nothing encryptions require at least two rounds of encryption, wherein the all-or-nothing transformation is achieved using encryption, which makes these conventional all-or-nothing encryptions inefficient in particular when dealing with
10 large files.

It is therefore an objective of the present invention to provide a method and a system for encrypting data based on an all-or-nothing encryption being more efficient, in particular improving a performance compared to conventional all-or-
15 nothing encryptions.

It is a further objective of the present invention to provide a method and a system for encrypting data based on an all-or-nothing encryption which are semantically secure and do not allow a recovery of plaintext even if the encryption key is known
20 to an adversary.

The aforementioned objectives are accomplished by a method of claim 1 and a system of claim 12.

25 In claim 1 a method for encrypting data based on an all-or-nothing encryption is defined.

According to claim 1 the method is characterized by the steps of

- 30 a) Providing data to be encrypted, preferably in form of a plaintext bitstream, and an encryption key, preferably wherein the encryption key may be secret or publicly known,
b) Dividing said data into an odd number of blocks, such that each of said blocks has the same size,

- 3 -

- c) Encrypting said blocks, preferably using symmetric encryption, with said encryption key obtaining an intermediate ciphertext comprising the encrypted blocks, and
- d) Linear transforming the intermediate ciphertext based on additive contravalence operations to obtain a final ciphertext.

In claim 13 a system for encrypting data based on an all-or-nothing encryption is defined, preferably in form of a computer, one or more processes and/or security modules, preferably for performing with a method according to one of the claims 1-12.

According to claim 13 the system is characterized by an input module operable to provide data to be encrypted, preferably in form of a plaintext bitstream, and an encryption key,

an encryption module operable to divide said data provided by the input module into an odd number of blocks, such that each of said blocks has the same size and to encrypt said blocks, preferably using symmetric encryption, with said encryption key obtaining an intermediate ciphertext comprising the encrypted blocks, and a transformation module operable to transform linearly the intermediate ciphertext based on additive contravalence operations to obtain a final ciphertext.

According to the invention it has been recognized that only one round of encryption and one round of additive contravalence operations are necessary thus providing enhanced efficiency.

According to the invention it has been further recognized that an efficient semantic security scheme, preferably a symmetric encryption scheme, with an efficient linear transformation is provided.

According to the invention it has been further recognized that the provided method and system do not allow a recovery of a plaintext even if the encryption key is known to an adversary. In particular even if the encryption key is known to the public the provided method and system still enable semantic security.

In other words on input of a plaintext bitstream and an encryption key K a plaintext is divided into blocks $p_1, \dots, p_N$, where N is odd such that each block size has size l , wherein l is preferably the block size of particular block cipher used. The set of input blocks $p_1, \dots, p_N$ is then encrypted under the encryption key K resulting in the intermediate ciphertext $c' = \{S, c_1', \dots, c_N'\}$, preferably S is a random seed and set as c_0' . When the linear transformation is applied on the intermediate ciphertext then the final ciphertext is obtained by additive contravalence operations, preferably addition and multiplications are implemented by means of XOR and AND operations, respectively.

Given the encryption key K the intermediate ciphertext c' can be computed by the final ciphertext c using the corresponding inverse operation of the linear transformation and decrypting the intermediate ciphertext c' with the encryption key K .

In the following it is shown that the method and system according to the invention is based on an all-or-nothing encryption: When the intermediate ciphertext is $c' = c_1', \dots, c_N'$ with N being an odd number and when the ciphertext c is an all-or-nothing encryption with the parameters $c_0', \dots, c_N'$, then when given any N blocks of the final ciphertext c an adversary can only recover one block of the intermediate ciphertext c' :

Without loss of generality, it may be assumed that the adversary is given blocks $c_0 \dots c_{(N-1)}$. Since N is odd, it is easy to compute $c_N' = c_0 \text{ XOR } c_1 \text{ XOR } \dots c_{(N-1)}$.

The set of remaining unknown blocks, i.e., $c_1' \dots c_{(N-1)'}$ can be computed as $[c_0 \dots c_N] * M$ where M is an N -by- N matrix with diagonal elements set to the value 0 and other elements set to the value 1. The Matrix M has the rank $N-1$, since, e.g., the last column can be computed XOR-ing the first $N-1$ columns. As the matrix does not have full rank and because the output of steps a)-d) of the provided method is pseudo-random, it is computationally infeasible for an adversary to recover all unknown $c_1' \dots c_{(N-1)'}$.

However it further can be shown that given the final ciphertext blocks $c_0 \dots c_{(N-1)}$ and once recovered the intermediate ciphertext block c_N' , it is computationally infeasible to recover any other block.

5 It can be assumed that if an adversary could correctly compute one intermediate ciphertext block c_i' , then the adversary could correctly compute all other unknown intermediate ciphertext blocks $c_1' \dots c_{(N-1)}'$. Nevertheless, this contradicts the aforementioned observation that, given N elements of the final ciphertext c , it is computationally infeasible to recover all elements in the intermediate ciphertext c' .

10 Therefore it can be concluded that given N blocks of the final ciphertext c , all but one block in the intermediate ciphertext c' are undetermined.

15 In the following it is shown that when a block of the plaintext $p_i = F(c_i, c_j)$ with F being the block cipher with $i \neq j$ for a given $F()$, the method and system according to the invention ensure that an adversary who has all final ciphertext blocks but one and also the encryption key K cannot require any plaintext block.

20 Therefore based on the above that only one block of the intermediate ciphertext c_j' can be recovered given any N blocks of the final ciphertext c , it can be assumed that an adversary when given the encryption key K and one block of the intermediate ciphertext c_j' the following situations may occur:

- 25 • It is computationally infeasible to compute a plaintext block p_j , since $p_i = F(c_j, c_i')$ and c_i' is a pseudo-random bitstring of size l that is unknown to the adversary.
- It is computationally infeasible to compute a plaintext block p_i , with $i < j$, since p_i is (computationally) independent from c_j' ,
- It is computationally infeasible to compute a plaintext block p_{j+1} , since $p_{j+1} = F(c_{j+1}', c_k')$, with $k < j+1$ and c_k' and c_{j+1}' are a pseudo-random bitstring of
- 30 size l that are unknown to the adversary.
- It is computationally infeasible to compute p_j with $j > i+1$ since $p_j = F(c_j, c_m')$ and both c_m, c_j' are two pseudo-random bitstrings of size l that are unknown to the adversary.

Therefore if the adversary has access only to the secret encryption key K and to any N blocks in the final ciphertext c the adversary cannot acquire any meaningful information about the plaintext p . This ensures one-way security of the plaintext. It can be even further shown that given the encryption key K and all but two of the
5 final ciphertext blocks, then the semantic security of the plaintext can be guaranteed.

Further features, advantages and preferred embodiments are described in the following subclaims.

10 According to a preferred embodiment the encryption key is publicly known. This enables for example to provide the encryption key to key management servers or systems enhancing the flexibility of the method allowing the use of key management and lowering the security restrictions on disclosure of the encryption
15 key.

According to a further preferred embodiment linear transforming the intermediate ciphertext is performed using a single matrix multiplication with the vector representing the intermediate ciphertext. This enables to easily compute the linear
20 transformation of the intermediate ciphertext to obtain the final ciphertext. A further advantage is that the linear transformation can be performed parallel.

According to a further preferred embodiment the matrix is provided invertible and identical to its invertible, preferably wherein the matrix is provided in form of a
25 diagonal matrix having the value 0 on the diagonal and the value 1 otherwise. This further enhances the efficiency when linear transforming the intermediate ciphertext and saves storage for the matrix since only a bitwise storage for each entry of the matrix has to be provided.

30 According to a further preferred embodiment the vector-matrix multiplication is performed using XOR- and AND-operations. This enables an efficient computation of the vector matrix multiplication.

According to a further preferred embodiment after step d) the final ciphertext, preferably represented in form a vector, comprises blocks, wherein each block is represented by the sum of all encrypted blocks of the intermediate ciphertext prior to step c) without the corresponding encrypted block corresponding to the same position in the intermediate ciphertext prior to step d). This provides an all-or-nothing encryption based on a block cipher encryption in the cipher block chaining mode.

According to a further preferred embodiment the steps c)-d) are performed by

7a) recursively encrypting each plaintext block of the actual round using the encryption key combined, preferably via an XOR-operation, with an encrypted plaintext block of the round before resulting in corresponding blocks of the intermediate ciphertext

7b) directly summing up all intermediate ciphertext blocks according to step 7a) and

7c) generating the blocks of the final ciphertext by directly summing up the result of step 7b) and the corresponding block of the intermediate ciphertext.

This provides an efficient implementation of cipher block chaining encryption mode for performing step c)-d).

According to a further preferred embodiment the blocks of the final ciphertext are distributed among at least two different storage servers, preferably providing at least two different storage services. This further enhances the security since the final ciphertext is distributed among different storage servers. The security can be even further enhanced when the different storage servers run on different operating systems so that a security leak of one storage service can not be used to access the other storage service of the other storage server(s).

According to a further preferred embodiment each storage server stores at least two blocks of the final ciphertext. This enables to efficiently use a limited number of servers for storing the blocks of the final ciphertext without losing the high level of security for the encrypted data.

According to a further preferred embodiment random data is added as a block into the intermediate ciphertext, preferably added at the beginning of the intermediate ciphertext. This provides an easy implementation of a random seed for encryption as well as an initial block of the intermediate ciphertext.

5

According to a further preferred embodiment even intermediate ciphertext blocks are generated based on an XOR-operation with the corresponding plaintext block and the result of an encryption of the random seed and the respective even number and odd ciphertext blocks are a result of an encryption of the corresponding plaintext block and an XOR-operation with the previously generated even intermediate ciphertext block. This enables parallelizing the generation of the intermediate ciphertext blocks resulting in a further enhanced efficiency.

10

According to a further preferred embodiment for performing encryption according to step b) a semantically secure encryption and symmetric encryption scheme is used, preferably counter mode encryption, cipher block chaining mode encryption, output feedback mode encryption and/or cipher feedback mode encryption. This enables to select the most appropriate encryption. For example using the output feedback mode encryption allows to generate the bit sequence with which a plaintext is bitwise XOR-combined in advance, whereas the block chaining mode encryption provides different ciphertext blocks in case of identical plaintext blocks.

15

20

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end it is to be referred to the patent claims subordinate to patent claim 1 on the one hand and to the following explanation of preferred embodiments of the invention by way of example, illustrated by the figure on the other hand. In connection with the explanation of the preferred embodiments of the invention by the aid of the figure, generally preferred embodiments and further developments of the teaching will be explained. In the drawings

25

30

Fig. 1 shows steps of a method base on an all-or-nothing encryption of data according to a first embodiment of the present invention and

Fig. 2 shows principal steps of a method according to a second embodiment of the present invention.

5

Fig. 1 shows that an all-or-nothing encryption of data according to a first embodiment of the present invention.

10

In Fig. 1 a possible instantiation for an encoding procedure guaranteeing an all-or-nothing encryption is shown.

15

In Fig. 1 the encoding procedure corresponds to block cipher encryption in the CBC mode. On input of an initialization vector IV which is set as first intermediate ciphertext block the other blocks of the intermediate cipher $\underline{c}_i$ are represented by the result of an encryption with encryption key K of the direct sum of the i-th plaintext block p_i and the previous intermediate cipher block c_{i-1}' , cf line 4 in Fig. 1. Then – cf. line 6 - the direct sum of all intermediate cipher blocks $c_0', \dots, c_N'$ is computed and then the final ciphertext blocks c_i are computed by directly adding t and the corresponding intermediate cipher block $\underline{c}_i$, cf. line 8 in Fig. 1.

20

In another example the intermediate ciphertext $c' = S, c_1, \dots, c_N$ wherein S is a random seed maybe constructed as follows:

25

- $c_i' = p_i \text{ XOR Enc_k}(S||i)$, when $i \text{ MOD } 2 = 0$.
- $c_i' = \text{Enc_k}(p_i \text{ XOR } c_{(i-1)'})$, when $i \text{ MOD } 2 = 1$.

Thus this may be seen as a combination of both the counter mode and the cipher block chaining encryption modes.

30

Fig. 2 shows principal steps of a method according to a second embodiment of the present invention.

In Fig. 2 steps of a method according to a further embodiment are shown:

In a first step the plaintext p is divided into N plaintext blocks $p_1, \dots, p_N$, wherein N is odd. Then each of the plaintext blocks $p_1, \dots, p_N$, is encoded by using for example AES resulting in the intermediate ciphertext blocks $c_1', \dots, c_N'$. After that an intermediate overall ciphertext t is computed by using the direct sum of all intermediate ciphertext blocks. In a further step each intermediate ciphertext block c_i and the intermediate overall ciphertext t are directly added resulting in the corresponding final ciphertext block $c_1, \dots, c_N$. The resulting final ciphertext blocks $c_1, \dots, c_N$ are then combined to the ciphertext c .

If the encryption key k is secret then the provided encryption method is semantically secure. If the encryption key is public and all but one ciphertext block is known then the encryption method is one way and if the encryption key is given and all but two ciphertext blocks are known then the encryption method is also semantically secure.

To further enhance security the different obtained final ciphertext block $c_1, \dots, c_N$ can be distributed on different servers of a distributed storage providing different storage services. Preferably each server stores a plurality of final ciphertext blocks.

In summary the present invention provides an efficient semantic security scheme with an efficient linear transformation. The present invention first encrypts and then applies a linear transform in contrast to conventional all-or-nothing encryption methods.

The present invention provides a fast all-or-nothing encryption and requires only one round of encryption and one round of bitwise XOR operations. The present invention enables semantic security since the underlying encryption is semantically secure and does not allow the recovery of plaintext even if the encryption key is known to an adversary. In particular even when the encryption key is known to the public the present invention still provides semantic security even when all but two of the final ciphertext blocks can be accessed.

One of the advantages of the present invention is that the all-or-nothing encryption according to the invention is efficient, in particular improving the performance of conventional all-or-nothing methods.

5 Many modifications and other embodiments of the invention set forth herein will come to mind the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to
10 the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

15

Claims

1. A method for encrypting data based on all-or-nothing encryption,
5 characterized by the steps of
 - a) Providing data to be encrypted, preferably in form of a plaintext
bitstream, and an encryption key, preferably wherein the encryption key
may be secret or publicly known,
 - 10 b) Dividing said data into an odd number of blocks, such that each of said
blocks has the same size
 - c) Encrypting said blocks, preferably using symmetric encryption, with said
encryption key obtaining an intermediate ciphertext comprising the
encrypted blocks, and
 - 15 d) Linear transforming the intermediate ciphertext based on additive
contravalence operations to obtain a final ciphertext.
2. The method according to claim 1, characterized in that the encryption key is
publicly known.
- 20 3. The method according to one of the claims 1-2, characterized in that linear
transforming the intermediate ciphertext is performed using a single matrix
multiplication with a vector representing the intermediate ciphertext.
- 25 4. The method according to claim 3, characterized in that the matrix is
provided invertible and identical to its invertible, preferably wherein the
matrix is provided in form of a diagonal matrix having the value 1 on the
diagonal and the value 0 otherwise.
- 30 5. The method according to one of the claims 3-4, characterized in that the
vector-matrix multiplication is performed using XOR- and AND-operations.

6. The method according to one of the claims 1-5, characterized in that after step d) the final ciphertext, preferably represented in form of a vector, comprises blocks, wherein each block is represented by the sum of all encrypted blocks of the intermediate ciphertext prior to step d) without the corresponding encrypted block corresponding to the same position in the intermediate ciphertext prior to step d).
7. The method according to one of the claims 1-6, characterized in that steps c)-d) are performed by
- 7a) recursively encrypting each plaintext block of the actual round using the encryption key combined, preferably via an XOR-operation, with an encrypted plaintext block of the round before resulting in corresponding blocks of the intermediate ciphertext,
- 7b) directly summing up all intermediate ciphertext blocks according to step 7a), and
- 7c) generating the blocks of the final ciphertext by directly summing up the result of step 7b) and the corresponding block of the intermediate ciphertext.
8. The method according to one of the claims 1-7, characterized in that the blocks of the final ciphertext are distributed among at least two different storage servers, preferably providing at least two different storage services.
9. The method according to claim 8, characterized in that each storage server stores at least two blocks of the final ciphertext.
10. The method according to one of the claims 1-9, characterized in that random data is added as a block into the intermediate ciphertext, preferably added at the beginning of the intermediate ciphertext.

11. The method according to one of the claims 1-10, characterized in that for performing encryption according to step b) a semantically secure encryption and symmetric encryption scheme is used, preferably counter mode encryption, cipher block chaining mode encryption, output feedback mode encryption and/or cipher feedback mode encryption.
12. The method according to one of the claims 1-11, characterized in that even intermediate ciphertext blocks are generated based on an XOR-operation with the corresponding plaintext block and the result of an encryption of the random seed and the respective even number and odd ciphertext blocks are a result of an encryption of the corresponding plaintext block and an XOR-operation with the previously generated even intermediate ciphertext block.
13. A system for encrypting data based on an all-or-nothing encryption, preferably in form of a computer, one or more processors and/or security modules, preferably for performing with a method according to one of the claims 1-12, characterized by
- an input module operable to provide data to be encrypted, preferably in form of a plaintext bitstream, and an encryption key,
- an encryption module operable to divide said data provided by the input module into an odd number of blocks, such that each of said blocks has the same size and to encrypt said blocks, preferably using symmetric encryption, with said encryption key obtaining an intermediate ciphertext comprising the encrypted blocks, and
- a transformation module operable to transform linearly the intermediate ciphertext based on additive contravalence operations to obtain a final ciphertext.