Method for Pseudo-probabilistic Block Encryption

Moldovyan Nikolay Andreevich[1],
Moldovyan Alexander Andreevich[1], Tam Nguyen Duc[2],
Hai Nguyen Nam[2], and Minh Nguyen Hieu[2(✉)]

[1] St. Petersburg Institute for Informatics and Automation
of Russian Academy of Sciences, St. Petersburg 199178, Russia
{nmold, ma}@mail.ru
[2] Academy of Cryptography Techniques, Hanoi, Vietnam
nguenductamkma@gmail.com, nnhaivn61@gmail.com,
hieuminhmta@gmail.com

Abstract. There is considered implementation of the plan-ahead share-key deniable encryption algorithms that produce the cryptogram that satisfy criterion of the computational indistinguishability from probabilistic encryption of the fake message. This paper introduces a general design of the pseudoprobabilistic block ciphers. The proposed method includes encryption of the secret message block and the fake message block followed by a transformation procedure mapping the pair of intermediate ciphertext blocks into a single block of the output ciphertext. The transformation procedure represents solving the system of two linear congruencies.

Keywords: Block cipher · Plan-ahead Shared-key
Pseudo-probabilistic cipher · Symmetric Deniable Encryption

1 Introduction

The notions of *public-key deniable encryption* and of *shared-key deniable encryption* were introduced by Canetti et al. in 1997 [1]. These important cryptographic primitives are applied in cryptographic protocols to resist coercive attacks. In the concept of deniable encryption there are considered sender-deniable, receiver-deniable, and sender-and-receiver-deniable (bi-deniable) schemes in which coercive adversary attacks the party sending message, the party receiving message, and the both parties, correspondingly. In the model of the coercive attack it is supposed that coercive adversary has power to force a party or the both parties simultaneously to open the cryptogram (ciphertext) after it has been sent.

Paper [1] initiated a lot of investigations on developing secure and efficient methods for public-key deniable encryption [2] in which no pre-shared information is used. Some of papers propose public-key deniable encryption combined with sharing secrete key (the sender and the receiver initially share a common secret key) and plan-ahead encryption (the fake message is selected at the stage of encryption) [3, 9, 10]. Detailed attention of the researchers to this direction in the area of deniable encryption is explained by the

Y. Chen and T. Q. Duong (Eds.): INISCOM 2017, LNICST 221, pp. 321–331, 2018.
https://doi.org/10.1007/978-3-319-74176-5_28

applicability of the public key deniable encryption to prevent vote buying in the internet-voting systems [4] and to provide secure multiparty computations [5].

Practical applications of the plan-ahead shared-key deniable encryption can be attributed to the case of the information protection against unauthorized access in computer and communication systems in the case of coercive attacks. As it is set in [1] for some models of such attacks "plan-ahead shared-key deniability is trivially solved: use different keys, and construct the ciphertext as concatenation of encryption of all messages, where the *i*th message is encrypted using the *i*th key".

The present paper considers the coercive-attack model against which this trivial construction is not applicable. To resist the proposed coercive attack, the paper proposes the plan-ahead shared-key deniable encryption methods producing cryptogram that is computationally indistinguishable from the ciphertext produced by some probabilistic cipher. The paper introduces design of pseudo-probabilistic block ciphers that satisfy the last criterion.

The organization of the paper is as follows. Section 2 describes the model of the coercive attack and criteria used for designing pseudo-probabilistic block ciphers. Section 3 proposes a simple method for pseudo-probabilistic block encryption. Section 4 introduces pseudo-probabilistic encryption algorithms satisfying an additional criterion of using the same decryption algorithm for disclosing both the secret and the fake message. The probabilistic encryption algorithms associated with the pseudo-probabilistic ones are presented in Sect. 5 followed by concluding Sect. 6.

2 Model of Adversary and Design Criteria

It is assumed that after ciphertext has been sent the adversary has possibility to force both the sender and the receiver to open the following:

- The plaintext corresponding to the ciphertext;
- Encryption and decryption algorithms;
- The encryption key with which encryption of the opened message yields all bits of the ciphertext.

Thus, in the considered model of the coercive attack the sender and the receiver are coerced to open parameters and algorithm of the ciphering procedure with which each bit of the sent ciphertext has been produced depending on the opened message (plaintext).

Security against the described attack can be provided using the symmetric deniable encryption (SDE) algorithm that produces the ciphertext like cryptogram produced as result of probabilistic encryption of the fake message with fake key. The ciphers satisfying the last criterion are called pseudo-probabilistic ciphers (PPC). Construction of the symmetric PPC can be implemented using the following design criteria:

- Symmetric deniable encryption should be performed as simultaneous encryption of two messages, secret one and fake one, using secret and fake keys (which are shared by sender and receiver);
- A probabilistic encryption algorithm should be associated with the SDE algorithm;

– The associated probabilistic encryption algorithm should transform the fake message with the fake key into the same ciphertext that is produced by the SDE algorithm;
– Using the fixed-size shared keys should provide performing secure SDE of messages having arbitrary length.

The parties of secure communication protocol can chart plausible that they used the probabilistic encryption to get higher resistance to potential attacks. Indeed, mixing the encrypted data with random data makes cryptanalysis more difficult. Next section describes a method for implementing pseudo-probabilistic block ciphering on the base of deterministic block encryption functions.

3 Simple Method for Pseudo-probabilistic Block Encryption

3.1 Method for Probabilistic Block Encryption

Suppose E be a b-bit encryption function, T be a π-bit block of plaintext, and R be a u-bit random block, where $u = b - \pi$. The b-bit input data block B can be formed as follows $B = R\|T$, where the sign $\|$ denotes the concatenation operation of two binary vectors, R and P:

$$P \rightarrow B = R||T \rightarrow C = E_K(B).$$

where K is the encryption key. Since the size of the plaintext block T is smaller than the ciphertext block, the last formula maps the given block of the plaintext block T on a large set of ciphertext blocks $\{C_1, C_2, \ldots, C_n\}$, where $n = 2^u$.

The general scheme of a probabilistic block cipher with such simple mechanism of concatenating a random bit string to plaintext block is illustrated in Fig. 1. The random number generator (RNG), like the encryption algorithm implementing the encryption function E, represents an internal part of the encryption device. It is assumed that the RNG is located in a protected part of the encryption device, and that a potential adversary cannot replace it. Thus, it is supposed that adversary has no possibility to manipulate the R value. Such assumption is acceptable, since encryption devices are designed so as to provide protection against encryption algorithm substitution, as well as against reading and copying the key. When decrypting a ciphertext block, the valid user (who knows the secret key) computes the data block $B = R\|T$ block and then the R value is discarded and the data block T of the original message is obtained.

When choosing different values of the b/π ratio, one can control performance and security of the such probabilistic encryption. Roughly speaking, the greater this ratio, the greater the security level. If the function E provides encryption rate s_0, then the speed of the probabilistic encryption is $s = s_0(b - r)/b$. Using such probabilistic encryption mechanism it is possible to provide resistance to potential future attacks based on unforeseen weaknesses of the encryption function E. Besides, such probabilistic mechanism can be also used to protect against possible attacks using trapdoors in block ciphers. However the probabilistic encryption procedure outputs ciphertext blocks that have size larger than the size of input data blocks. To compensate for the expansion effect, one can compress the source message before performing encryption.

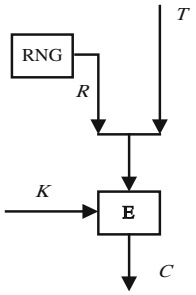

Fig. 1. The basic scheme of probabilistic block encryption.

One can also note that the compression of data before performing data encryption increases additionally resistance against ciphertext attacks.

3.2 Probabilistic Mixing Data Bits with Random Bits

The described above simple probabilistic encryption mechanism, based on generating a ciphered data block by combining random and data bits, can be used to increase encryption security when using many of the known block cryptoalgorithms. In regard to many types of attacks and well known block ciphers it is sufficient to define a relatively small ratio of random bits to data bits. However, for ciphers vulnerable to differential and linear cryptanalysis, strengthening on the basis of this probabilistic encryption method can require a significant increase the portion of random bits, which will result in noticeable reducing the data encryption performance, and significant increase of the ciphertext size.

Let us consider some variants of making the probabilistic encryption method more effective for a small portion of random bits when using encryption procedures with good diffusion properties, but possibly, with unexpected vulnerabilities to differential and linear analysis.

The probabilistic block encryption scheme shown in Fig. 1 can be complemented with a non-deterministic mix of random and data bits. To implement this idea, a random binary vector is divided into two parts with a pre-specified length: $R = R_1 \| R_2$. Then, prior to carrying out encryption transformations over the $R_2 \| T$ binary vector, a bit permutation is done, which depends on the R_1 random value that specifies randomly mixing the bits of the message T and those of the R_2 random value. For bit mixing, it is possible to use controlled operational permutation boxes **P**, used earlier as a basic cryptographic primitive to design secure fast ciphers [8]. The permutation performed by a **P** box depends on the value of the control vector V that is generated depending on R_1. The sequence of transformations in a variant with a random combination of data and random bits (see Fig. 2) is as follows:

$$T \rightarrow R_2||T \rightarrow \mathbf{P}_V(R_2||T) \rightarrow R_1||\mathbf{P}_V(R_2||T) \rightarrow \mathbf{E}_K(R_1||\mathbf{P}_V(R_2||T)).$$

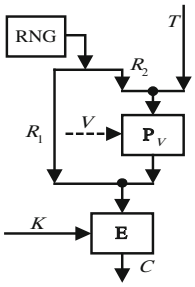

Fig. 2. A scheme with a probabilistic mixing random and data bits.

In typical **P** boxes, the length v of the V control vector is at least twice the length of the $R_2\|T$ $(r_2 + t)$ vector being transformed. In this case, it is assumed that the $r_1 < r_2 + t < v$ condition is true, so the control vector can be created, for example, by repeatedly replicating the R_1 vector ($V = R_1\|\ldots\|R_1\|R_1$), or by alternating R_1 and the K_1 fragment of the private key ($V = R_1\|K_1\|R_1\|K_1$). In the latter case, mixing the bits of R_2 and T is done probabilistically, depending on the private key. Increasing the security against differential and liner cryptanalysis is connected with the probabilistic distribution of the data bits over the bit positions of the data block being encrypted. For example, when performing a chosen-plaintext differential analysis, the probability of getting two data blocks with a given difference is significantly small for r_1, $r_2 = 8$. When $b = 64$ and 128, this corresponds to a rather small portion of random bits (25% and 12%, respectively).

The second way of making a simple probabilistic encryption scheme more secure is related to the idea of pre-encrypting an original text T using a randomly generated value R as a one-time pre-encryption key (see Fig. 3). The transformation sequence is the following:

$$T \rightarrow E'_R(T) \rightarrow E''_K(R||E'_R(T)).$$

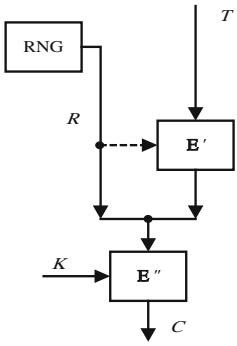

Fig. 3. Probabilistic block encryption scheme including pre-encryption of the source data block.

Strengthening is done using additional transformations with some single-use key whose repetition probability is about 2^{-r} during attacks based on the chosen values T and C (due to sufficiently good diffusion properties of E'' encryption procedures). When doing the pre-encryption, the basic scheme of probabilistic encryption can be used, which will lead to the following transformation sequence (see Fig. 4):

$$T \to R_2||T \to E'_{R1}(R_2||T) \to E''_K(R_1||E'_{R1}(R_2||T)).$$

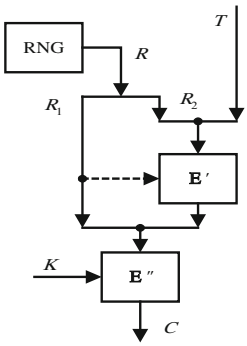

Fig. 4. Two-stage probabilistic encryption.

This case relates to the third variant of increasing security, and it is a generalization of the first variant, in which mixing up random and data bits can be considered a special case of encrypting transformation.

For a hardware implementation, the first variant is the most cost-effective,whilethe second and third variants are the most cost-effective for a software implementation. From the standpoint of increasing security, the third variant is best. In general, the increase in security in the variants discussed is related to the fact that the ratios that connect the T and C pairs of values also include a random (pseudo-random) value R during chosen-plaintext T attacks (chosen-plaintext C attacks).

These probabilistic encryption methods seem to be quite effective for insuring against unexpected weaknesses of the encryption algorithm used, and against built-in trapdoors. Expanding the ciphertext block puts significant limitations on using probabilistic ciphers in computer systems. To compensate for the expansion effect, it is possible to compress the original message beforehand. In some cases, this method makes it possible to design probabilistic ciphers in which the ciphertext length is equal to the length of the original message. Besides which, compressing data before they are encrypted significantly increases the security of the encryption. For many applications in telecommunication systems, this variant of probabilistic encryption can be used without significant limitations.

3.3 A Simple Method for Pseudo-probabilistic Block Encryption

The described variants of implementing the probabilistic block encryption can be easily transformed into methods for pseudo-probabilistic block encryption. Indeed, one can replace the PRNG by the block encryption algorithm E^* performing encryption of the secret message T^* with the key K^*. If the algorithm E^* provides sufficiently high security then the output ciphertext blocks can be used as random bit string R in the probabilistic encryption schemes shown in Figs. 1, 2, 3 and 4. At such modification the data blocks are considered as blocks of some fake message that is encrypted simultaneously with the secret message blocks T^*.

At time of the coercive attack the sender and the receiver of the secret message have possibility to cheat plausible they used probabilistic block encryption algorithm. They will open the fake message and the encryption key K with which the fake message was encrypted. The coercer can decrypt the intercepted ciphertext with key K and obtain the fake message. He is also able to get pseudo-random bit strings R^*, but for him is computationally infeasible to distinguish the pseudo-random values R^* from random ones and to demonstrate that the ciphertext contains one message more.

In the next section we consider pseudo-probabilistic block encryption methods that satisfy an important additional requirement to deniable encryption schemes, which provides security to coercive attacks with measuring the decryption time. The additional requirement is formulated as performing decryption of both the secret message and the fake message with the same decryption algorithm.

4 Method for Pseudo-probabilistic Block Encryption

It is proposed to implement pseudo-probabilistic encryption as simultaneous ciphering two messages Mess1 (fake) and Text2 (secret) the equal size using the shared keys (K_1, m_1) and (K_2, m_2), where K_1 and K_2 are keys of some block cipher E with v-bit input; m_1 and m_2 are two mutually prime numbers. The messages are divided into v-bit data blocks: Mess1 = $(M_1, M_1, \ldots, M_z)$ and Text2 = $(T_1, T_1, \ldots, T_z)$ and then pairs of the respective blocks M_i and T_i are consecutively encrypted as follows:

1. Using the block cipher E and key K_1, it is encrypted the block M of the first message: $C_M = E_{K1}(M)$.
2. Using the block cipher E and key K_2, it is encrypted the block T of the second message: $C_T = E_{K2}(T)$.
3. Using additional secret values m_1 and m_2 compute the block C of output ciphertext as solution of the following system if two congruencies

$$\begin{cases} C \equiv C_M \bmod m_1 \\ C \equiv C_T \bmod m_2 \end{cases}, \tag{1}$$

where blocks C_T and C_M of the intermediate ciphertexts are interpreted as binary numbers; m_1 and m_2 are mutually prime numbers having size $v + 1$ bits. The size of the output ciphertext block C is equal to $2v + 2$ bits (i.e. the size of the block C is two bits

larger than the sum of sizes of the blocks C_T and C_M). Solution of the system (1) is described as follows:

$$C = \left[C_M m_2 \left(m_2^{-1} \bmod m_1\right) + C_T m_1 \left(m_1^{-1} \bmod m_2\right)\right] \bmod m_1 m_2.$$

The values $m_2(m_2^{-1} \bmod m_1)$ and $m_1(m_1^{-1} \bmod m_2)$ can be pre-computed at moment of generating the secret keys, therefore the main contribution in computational difficulty of calculating the value C is defined by the operation of dividing the value in square brackets by the modulus $m_1 m_2$.

From practical point of view it is preferable to use the pseudo-probabilistic block encryption method that outputs the ciphertext block that have size equal exactly to $2v$ bits. This requirement can be met using the procedure of combining two blocks C_T and C_M into one block C which consists in solving the following system of two congruences defined over binary polynomials:

$$\begin{cases} C \equiv E_{K_1}(M) \bmod \mu(x) \\ C \equiv E_{K_2}(T) \bmod \lambda(x) \end{cases}, \tag{2}$$

where $\mu(x)$ and $\lambda(x)$ are mutually irreducible binary polynomials of the degree v (these two polynomials are secret elements); the v-bit blocks C_T and C_M of the intermediate ciphertexts are interpreted as binary polynomials of the degree $v - 1$. Solution of system (2) represents the binary polynomial of the degree $2v$ which is given by the following formula:

$$C = \left[E_{K1}(M)\lambda(x)\left(\lambda^{-1}(x) \bmod \mu(x)\right) + E_{K2}(T)\mu(x)\left(\mu^{-1}(x) \bmod \lambda(x)\right)\right] \bmod \mu(x)\lambda(x).$$

Like in the first block encryption method, the polynomials $\lambda^{-1}(x) \bmod \mu(x)$ and $\mu^{-1}(x) \bmod \lambda(x)$ can be pre-computed to increase the encryption rate.

The related decryption algorithms are evident for the described two variants of the proposed pseudo-probabilistic block encryption method.

Decryption algorithms connected with the pseudo-probabilistic encryption algorithms described in this section coincide with the decryption algorithms connected with the associated probabilistic encryption algorithms (see Sect. 5).

5 Associated Probabilistic Block Encryption Algorithms

Let us show that the block encryption method described in Sect. 3 met criterion of computational indistinguishability from probabilistic block encryption. For this purpose one should propose a probabilistic block encryption algorithm such that, when being applied to encrypt the fake message, it can potentially produce the ciphertext coinciding with the ciphertext produced by the pseudo-probabilistic block encryption algorithm.

Probabilistic block encryption algorithm associated with the PPC including procedure of solving the system of congruences (1) is described as follows. The fake key

represents the pair of secret values (K_1, m_1). To encrypt the fake message data block M the following steps are performed:

1. The data block M is encrypted with the block cipher algorithm E: $C_M = E_{K1}(M)$.
2. It is generated a random value $R < 2^v$ and a random prime number r such that $2^v < r < 2^{v+1}$.
3. It is computed the output ciphertext block C as solution of the following system of congruences

$$\begin{cases} C \equiv C_M \bmod m_1 \\ C \equiv R \bmod r \end{cases}, \tag{3}$$

It is easy to see that the arbitrary value C^* such that $C^* \equiv C_M \bmod m_1$ can be obtained as solution of system (3) at different pairs of the values $R < 2^v$ and $r < 2^{v+1}$. Indeed, let us select a random number r^* such that $2^v < r^* < 2^{v+1}$. The respective value R^* is computed as $R^* \equiv C^* \bmod r$.

Decryption of the ciphertext block is performed as follows.

Algorithm for disclosing the fake message.

1. Compute the intermediate ciphertext block C_M:$C_M = C \bmod m_1$.
2. Compute the data block M:$M = E_{K1}^{-1}(C_M)$.

Algorithm for disclosing the secret message.

1. Compute the intermediate ciphertext block C_T:$C_T = C \bmod m_2$.
2. Compute the data block T:$T = E_{K1}^{-1}(C_T)$.

Probabilistic block encryption algorithm associated with the PPC including procedure of solving the system of congruences (2) is described as follows. The fake key represents the pair of secret values (K_1, $\mu(x)$). Encryption of the data block M of the fake message is performed as follows:

1. The data block M is encrypted with the block cipher algorithm E:$C_M = E_{K1}(M)$.
2. It is generated a random binary polynomials $\lambda(x)$ (of the degree equal to v or less) and $\rho(x)$ (of the degree $v + 1$).
3. It is computed the output ciphertext block C as solution of the following system of congruences (the ciphertext block C_M is considered as binary polynomial):

$$\begin{cases} C \equiv C_M \bmod \mu(x) \\ C \equiv \lambda(x) \bmod \rho(x) \end{cases}, \tag{4}$$

Evidently, a bit string C^* such that $C^* = C_M \bmod \mu(x)$ can be obtained as solution of system (4) at different pairs of the polynomials $\lambda(x)$ and $\rho(x)$. Indeed, for arbitrary polynomial $\rho(x)$ of the degree $v + 1$ the related polynomial is $\lambda(x) = C^* \bmod \mu(x)$.

Decryption of the ciphertext block is performed as follows.

Algorithm for disclosing the fake message.

1. Compute the intermediate ciphertext block C_M:$C_M = C \bmod \mu(x)$.
2. Compute the data block M:$M = E_{K1}^{-1}(C_M)$.

Algorithm for disclosing the secret message.

1. Compute the intermediate ciphertext block C_T:$C_T = C \bmod \lambda(x)$.
2. Compute the data block T: $T = E_{K2}^{-1}(C_T)$.

6 Conclusion

It has been proposed to construct the block deniable encryption as process of pseudo-probabilistic block encryption of secret and fake messages. At time of the coercive attack the sender or/and receiver open the fake message and fake encryption key and declare their using the probabilistic block encryption of the opened message. The coercer is computationally unable to distinguish the intercepted ciphertext from ciphertext produced by probabilistic encryption. The proposed deniable encryption method is fast and provides bi-deniability (resistance to simultaneous attack on both the sender and the receiver of the message).

The main result of the paper is its contribution to the class of pseudo-probabilistic ciphers to which one can attribute the proposed block crypto schemes and introduced earlier pseudo-probabilistic stream ciphers [6, 7].

References

1. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052229
2. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_30
3. Moldovyan, N.A., Shcherbacov, A.V., Eremeev, M.A.: Deniable-encryption protocols based on commutative ciphers. Quasigroups Relat. Syst. 95–108 (2017)
4. Meng, B.: A secure internet voting protocol based on non-interactive deniable authentication protocol and proof protocol that two ciphertexts are encryption of the same plaintext. J. Netw. **4**, 370–377 (2009)
5. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 406–425. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_23
6. Moldovyan, A.A., Moldovyan, D.N., Shcherbacov, V.A.: Stream deniable-encryption algorithm satisfying criterion of the computational indistinguishability from probabilistic ciphering. In: Workshop on Foundations of Informatics, Chisinau, pp. 318–330 (2015)

7. Moldovyan, N.A., Moldovyan, A.A., Moldovyan, D.N., Shcherbacov, V.A.: Stream deniable-encryption algorithms. Comput. Sci. J. Mold. **24**, 68–82 (2016)
8. Moldovyan, N.A., Moldovyan, A.A.: Data-driven Block Ciphers for Fast Telecommunication Systems, 1st edn. Auerbach Publications, Boston (2007)
9. Dürmuth, M., Freeman, D.M.: Deniable encryption with negligible detection probability: an interactive construction. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 610–626. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_33
10. Barakat, T.M.: A new sender-side public-key deniable encryption scheme with fast decryption. KSII Trans. Internet Inf. Syst. (TIIS) **8**, 3231–3249 (2014)