Quasigroups and Related Systems **25** (2017), 95 − 108

Deniable-encryption protocols based on commutative ciphers

Nicolai A. Moldovyan, Alexei V. Shcherbacov and Mikhail A. Eremeev

Abstract. There are considered three new deniable encryption protocols representing practical interest. The sender-deniable and sender&receiver-deniable ones have been designed on the base of combining commutative encryption function (Vernam cipher) with probabilistic public key encryption (RSA algorithm), subexponential resistance to coercive attack being obtained. To get exponential deniability it is proposed to use the ElGamal-like probabilistic algorithm based on computational difficulty of discrete logarithm on elliptic curves instead of the RSA one. The third DE protocol is based on the Pohlig−Hellman exponentiation cipher and represents a plan-ahead shared-key bi-deniable scheme satisfying criterion of computational indistinguishability from probabilistic encryption protocol. Each of the proposed deniable encryption schemes is a three-pass protocol.

1. Introduction

1.1. Deniable encryption

Encryption is usually used to provide confidentiality of the messages sent via insecure public channels, when a potential adversary can intercepts the send messages. In the case of intercepting the sent ciphertext he is unable to read the message until disclosing the decryption key. The widely used private-key (AES, IDEA, RC5, Serpent et. al.) and public-key (RSA, ElGamal et. al.) encryption algorithms [22] provide computational infeasibility of disclosing the key while performing cryptanalysis of the ciphertext. In some particular applications of the cryptographic protocols it is required to provide security against potential coercive attacks. The main feature of the model of the coercive adversary (coercer) consists in his having power to force sender or/and receiver to open both the source message and the decryption key [2]. After he gets the private key he can check that with the opened key the intercepted ciphertext is decrypted into the opened message.

The notion of *deniable encryption* (DE) relates to cryptoschemes that are resistant to coercive attacks. Deniability is provided with possibility to decrypt the

2010 Mathematics Subject Classification: 94A60, 11S05.
Keywords: cryptography, public encryption, commutative encryption, deniable encryption, shared key, probabilistic encryption
The work was supported by Russian Science Foundation, research project 17-11-01252

ciphertext intercepted by the coercer in different ways. The sender or/and the receiver open a fake message instead of the secret one and the coercive adversary is not able to disclose their lie. Practical application of the DE algorithms and protocols is connected with providing data secrecy, secure communications via public channels. They are also applicable for preventing vote buying in the internet-voting systems [1, 12] and for providing secure multiparty computations [9]. There are distinguished sender-deniable [4, 8, 19], receiver-deniable [13, 25], and bi-deniable [20, 21], schemes in which coercer attacks the sender of secret message, the receiver, and the both parties of the communication protocol, respectively.

One should also mention the issue about time at which the attacked parties have to decide on the fake message. In the *plan-ahead* DE protocols the fake message is selected at time of encryption. There are known practical public-key DE schemes [16, 17] and shared-key DE ones [18] in which the fake message is fixed and selected before or during the encryption process. From theoretic point of view the *flexible* DE protocols represent significant interest, in which the fake massage can be selected arbitrary at time of the coercive attack.

Significant part of the papers devoted to the design and analysis of flexible DE protocols consider the case of the sender-deniable public-key encryption protocols [1, 4, 8]. A possible general scheme of such protocols is as follows. The secret message M is encrypted with public-encryption algorithm E and public key P using a random value $r : C = E_P(M, r)$, where C is the produced cryptogram (ciphertext). While being coerced the sender (receiver) opens to adversary the fake message M' and another random value r' (fake opening) such that $E_P(M', r') = C$, where $r' \neq r$. The value r contains some trapdoor information unavailable to coercer, which is used by receiver to decide on the pair (M', r') containing real message. Papers [3, 11, 21] considered some problems connected with construction of the flexible public key DE protocols having super-polynomial security. Recent paper [24] gave the first construction of sender-deniable encryption schemes with super-polynomial security, where a coercive adversary has negligible advantage in distinguishing real and fake messages.

Present paper proposes a novel design of the DE protocols based on combining the probabilistic public encryption with the commutative encryption function implemented with Vernam algorithm. The paper introduces a computationally efficient sender-deniable encryption protocol as well as sender&receiver deniable one in which using respective fake key the ciphertext can be decrypted in arbitrary fake message selected after performing the protocol, namely, at time of coercive attack. The both protocols have super-polynomial security that is defined by sub-exponential security of the RSA public encryption algorithm put into the base of the protocols. The proposed protocols are based on combining the probabilistic public encryption with commutative encryption implemented with the Vernam cipher. The proposed design can be implemented with using the ElGamal-like public encryption on elliptic curves, providing exponential resistance to coercive attack. As compared with the known flexible public key DE protocols the proposed ones have the following merits: i) simplicity of the design, ii) sufficiently high performance, iii) comparatively low overhead in terms of the ciphertext size, and iv) using only one XOR operation to generate a fake opening at time of attack.

1.2. Commutative encryption

Encryption function E is called commutative if it satisfies the following condition

$$E_K[E_Q(M)] = E_Q[E_K(M)],$$

where K and Q are encryption keys and M is some plaintext, for arbitrary keys K and $Q \neq K$. The property of commutativity of some encryption function is exploited in Shamir's no key protocol (also called Shamir's three-pass protocol [14]) described as follows. Suppose Alice wishes to send the secret message M to Bob, using a public channel and no shared key. For this purpose they can use the following protocol that provides privacy, but not authentication:

1. Alice chooses a random key K and encrypts the message M using a commutative encryption function $E : C_1 = E_K(M)$, where C_1 is the produced ciphertext. Then she sends the ciphertext C_1 to Bob.

2. Bob chooses a random key Q and encrypts the message the ciphertext C_1 using the function E as follows: $C_2 = E_Q(C_1)$, where C_2 is the produced ciphertext. Then he sends the ciphertext C_2 to Alice.

3. Alice decrypts the ciphertext C_2 obtaining the ciphertext $C_3 : C_3 = E_K^{-1}(C_2)$. Then she sends the ciphertext C_3 to Bob.

Having received the ciphertext C_3 Bob computes the value $M' = E_Q^{-1}(C_3)$. Due to commutativity of the encryption function the values M' and M are equal, i.e., the protocol works correctly. Indeed, one has the following:

$$M' = E_Q^{-1}(C_3) = E_Q^{-1}[E_K^{-1}(C_2)] = E_Q^{-1}[E_K^{-1}[E_Q(C_1)]] =$$
$$E_Q^{-1}[E_K^{-1}[E_Q(E_K(M))]] = E_Q^{-1}[E_K^{-1}[E_K(E_Q(M))]] = E_Q^{-1}[E_Q(M)] = M.$$

The described three-pass protocol provides security to passive attacks (potential adversary only intercepts the values sent via public channel), if the used commutative encryption function E is secure to the know-input-text attack.

Indeed, if the function E is not secure to such attack, then the passive adversary (after his intercepting the ciphertexts C_1, C_2, and C_3) is able to compute Bob's local key Q from the equation $C_2 = E_Q(C_1)$ and then the secret message $M = E_Q^{-1}(C_3)$.

The Vernam cipher represents the simplest commutative cipher. It consists in simple adding the key to the message M in accordance with the formula

$$C = M \oplus K,$$

where K is the single-use random chosen key such that $|K| = |M|$ (the bit-length of some value x is denoted as $|x|$) and $\oplus$ is the bit-wise modulo 2 addition operation (the XOR operation). Unfortunately it cannot be used in frame of the Shamir's three-pass protocol, since it is not secure to the known-plaintext attack.

The appropriate commutative encryption function is provided by the exponentiation-encryption method proposed by Pohlig and Hellman in [7].

The last method is described as follows. Suppose p is a 2464-bit prime such that number $(p-1)$ contains a large prime divisor q, for example, $p = 2q + 1$.

To select an encryption/decryption key (e, d) one needs to generate a random 256-bit number e that is mutually prime with $(p-1)$ and then to compute $d = e^{-1} \bmod p-1$. The encryption procedure is described with the formula

$$C = M^{-e} \bmod p.$$

Decryption of the ciphertext C is performed as computing the value

$$M = C^{-d} \bmod p.$$

The Pohlig-Hellman algorithm is secure against the known plaintext (ciphertext) attack and can be used in Shamir's no-key protocol.

In the present paper it is also proposed bi-deniable shared-key protocol based on commutative encryption implemented with the Pohlig-Hellman exponentiation cipher. Justifying the bi-deniability of the proposed protocol is performed on the base of the criterion of computational indistinguishability [18] from the probabilistic three-pass protocol applied for encrypting a fake message.

2. Sender&reciever-deniable three pass protocol

In frame of the protocol described below the RSA cryptoscheme [23] is used for performing the public encryption with receiver's (Bob's) public key (n, e) that is generated simultaneously with his private key d as follows. Bob selects two strong [6] primes p and q having large size (for example, 1232 bits). The value n is computed as product of the primes: $n = pq$. Then it is selected a random number e that is relatively prime to Euler phi function $\varphi(n) = (p-1)(q-1)$ and has comparatively small size (for example, 32 bits) to provide faster encryption. The private key d is computed as follows $d = e^{-1} \bmod \varphi(n)$. Probabilistic encryption of some message $M < (n \text{ div } 2^{257})$ is performed with the public key as computing the ciphertext $C = (M||\rho)^e \bmod n$, where $||$ is the concatenation operation; ρ is a random chosen bit string having size exacly equal to 256 bits. Decryption of the ciphertext C is performed using the private key as follows $M = \left(C^d \bmod n\right) \text{ div } 2^{256}$. The random value ρ is an internal randomization parameter actual in frame of the operation of probabilistic public encryption. The protocols described below do not use any information contained in the value ρ destination of which consist only in randomizing the ciphertext. The parameter ρ takes on different values at each step of the probabilistic RSA encryption and they are not to be saved in computer or hardware memory.

The proposed sender-deniable public encryption protocol is described as follows.

1. To send the secret message M ($|M| < |n_B \text{ div } 2^{257}|$, where (n_B, e_B) is Bob's public key) Alice generates a random bit string K such that $|K| = |M|$ and computes the value $C = M \oplus K$ and the ciphertext

$$C_1 = (C||\rho)^{e_B} \bmod n_B = ((M \oplus K)||\rho)^{e_B} \bmod n_B.$$

Then she sends the value C_1 to Bob.

2. Using his private key d_B Bob decrypts the ciphertext C_1: $C||\rho = C_1^{d_B} \bmod n_B$, generates a random bit string Q such that $|Q| = |C|$ and computes the ciphertext

$$C_2 = C \oplus Q = M \oplus K \oplus Q.$$

Then he sends the value C_2 to Alice.

3. Alice computes the ciphertext

$$C_3 = ((C_2 \oplus K)||\rho)^{e_B} \bmod n_B = ((M \oplus Q)||\rho)^{e_B} \bmod n_B$$

and sends the value C_3 to Bob.

Bob decrypts the ciphertext $C_3 : (M \oplus Q)||\rho = (C_3)^{d_B} \bmod n_B$ and discloses the secret message M as follows: $M = (M \oplus Q) \oplus Q$.

If some coercive adversary intercepts the ciphertexts C_1, C_2, and C_3 and then forces Alice to open the secret message and her local key, then she chooses some fake message M' such that $|M'| = |M|$, computes the fake local key $K' = M \oplus K \oplus M'$, and opens the values M' and K' as the values had been used at step 1 of the protocol. From the ciphertext C_2 coercer can compute the value $Q' = C_2 \oplus M' \oplus K'$ for which the following inequality holds $M' \oplus Q' \neq M \oplus Q$. However the coercer has no computational possibility to disclose Alice's lie due to the probabilistic encryption performed at step 3 which gives different pseudo-random ciphertexts while encrypting the same input value arbitrary number of times.

Thus, the coercer is unable to demonstrate inequality $M' \oplus Q' \neq M \oplus Q$ performing public encryption of its left part, using Bob's public key, therefore the described protocol is sender-deniable one. However the protocol is not a receiver-deniable one, since while being coerced Bob should open both his local key Q and his private key d_B. Using the value d_B the coercer is able to disclose Bob's lie, if Bob will open fake key $Q' \neq Q$.

The described protocol can be modified into sender- and receiver-deniable one with using Alice's public key (n_A, e_A) at step 2 of the protocol. The modified protocol looks as follows:

1. To send the secret message M ($|M| < |n \text{ div } 2^{257}|$, where $n = \min\{n_A, n_B\}$,) Alice generates a random bit string K such that $|K| = |M|$ and computes the value $C = M \oplus K$ and the ciphertext

$$C_1 = (C||\rho)^{e_B} \bmod n_B = ((M \oplus K)||\rho)^{e_B} \bmod n_B.$$

Then she sends the ciphertext C_1 to Bob.

2. Using his private key d_B Bob decrypts the ciphertext $C_1 : C||\rho = C_1^{d_B} \bmod n_B$, generates a random bit string Q such that $|Q| = |C|$ and computes the value $C_2' = C \oplus Q = M \oplus K \oplus Q$ and the ciphertext

$$C_2 = (C_2'||\rho)^{e_A} \bmod n_A = ((M \oplus K \oplus Q))||\rho)^{e_A} \bmod n_A.$$

Then he sends the ciphertext C_2 to Alice.

3. Alice computes the values $C_2'||\rho = C_2^{d_A} \bmod n_A$ and

$$C_3 = (C''||\rho)^{e_B} \bmod n_B = ((M \oplus Q))||\rho)^{e_B} \bmod n_B$$

and sends the value C_3 to Bob.

Bob decrypts the ciphertext $C_3 : (M \oplus Q))||\rho = (C_3)^{d_B} \bmod n_B$ and discloses the secret message M as follows: $M = (M \oplus Q) \oplus Q$.

Like the initial version, the modified version of the protocol resists the sender-side coercive attack. Besides, it is also a receiver-deniable protocol. Indeed, if some coercive adversary intercepts the ciphertexts C_1, C_2, and C_3 and then forces Bob to open the secret message and his local key, then Bob chooses some fake message M' such that $|M'| = |M|$, computes the fake local key $Q' = M \oplus Q \oplus M'$, and opens the values M' and Q' as the real values used during execution of the protocol. The coercer can compute the value $C = M \oplus K = M' \oplus K'$, where K' is fake Alice's local key, from the ciphertext C_1 and the value $C'' = M' \oplus Q'$ from the ciphertext C_2. For two different messages M' and M the following inequality holds $M' \oplus K' \oplus Q' \neq M \oplus K \oplus Q$. However, due to using probabilistic public encryption, the coercer has no computational possibility to disclose Bob's lie performing many times the encryption of the left part of the inequality with Aice's public key. The coercer is also unable to compute the value $M \oplus K \oplus Q$ performing decryption of the ciphertext C_2, since he does not know the Alice's private key.

It should be noted that the last protocol is not fully bi-deniable, since it does not resist simultaneous coercive attack on both the sender and the receiver. Indeed, while being simultaneously coerced Alice and Bob should open both their local keys K and Q and their private keys d_A and d_B. Using the values d_A and d_B the coercer is able to disclose Alice's and/or Bob's lie, if Alice and/or Bob will open fake keys $K' \neq K$ and/or $Q' \neq Q$.

The described three-pass protocols are sufficiently practical since only four and six modulo exponentiation operations are performed during the first and second described protocols, respectively. The both protocols provide security defined by computational difficulty of the factoring n problem (about 2^{128} modulo multiplications in the case of 2464-bit modulus n). The second protocol provides authentication due to using both the Alice's public key and Bob's public key. The first protocol provides authentication of one party of the protocol only, namely, authentication of the receiver of the message.

3. Bi-deniable three-pass protocol

For constructing a practical bi-deniable encryption protocol the following design criteria have been used:

1) the protocol should use a key (128 to 2048 bits) shared by sender and receiver of secrete message;

2) the base encryption procedure should be implemented as the modulo exponentiation operation in the finite field $GF(2^s)$, where $s = 128$ to 2048;

3) the protocol should provide bi-deniability, i.e., it should resist simultaneous coercive attacks on the sender and on the receiver;

4) under coercive attack the parties of the protocol disclose a fake shared key and their local fake keys as secret values; when using the fake keys, decryption of the ciphertexts (sent during the deniable-encryption protocol) should recover a fake message;

5) ciphertexts produced at all steps of the protocol should be computationally indistinguishable from the ciphertexts produced by some probabilistic-encryption protocol in the case when the last protocol is used for encrypting some fake message using the disclosed keys.

Construction of the shared-key bi-deniable encryption protocols is connected with the design of respective probabilistic three-pass protocol, which is associated with the first one. The next subsection introduces appropriate probabilistic-encryption protocol.

3.1. Associated probabilistic-encryption protocol

Suppose Alice and Bob share a secret key representing an irreducible binary polynomial $\mu(x)$ of the degree $s = 128$ to 1024. To encrypt some secret message M Alice represents the message as sequence of the s-bit data blocks $M_i : M = (M_1, M_2, , M_i, , M_z)$. To send securely the message M to Bob she can use the following probabilistic-encryption protocol.

1. Alice generates her local key as pair of values (e_A, d_A), where random value e_A is mutually prime with the value $2^s - 1$ and $d_A = e_A^{-1} \bmod 2^s - 1$. Then for each value $i = 1, 2, \ldots, z$ she generates random binary polynomials $\rho_A(x)$ of the degree $s - 1$ and $\eta_A(x)$ of the degree s such that $\eta_A(x) \neq \mu(x)$ and, considering each data block as binary polynomial, encrypts the message M in accordance with the formula

$$\begin{aligned} C_{Ai} = \{&\eta_A(x)[\eta_A^{-1}(x) M_i^{e_A} \bmod \mu(x)] + \mu(x)[\mu^{-1}(x)\rho_A(x) \bmod \eta_A(x)]\} \\ &\bmod \mu(x)\eta_A(x). \end{aligned} \tag{1}$$

Then Alice sends the ciphertext

$$C_A = (C_{A1}, C_{A2}, \ldots, C_{Ai}, \ldots, C_{Az})$$

to Bob.

2. Bob generates his local key (e_B, d_B), where random value e_B is mutually prime with the value $2^s - 1$ and $d_B = e_B^{-1} \bmod 2^s - 1$. Then for each value $i = 1, 2, \ldots, z$ he computes the value $C'_{Bi} = C_{Ai}^{e_B} \bmod \mu(x) = M_i^{e_A e_B} \bmod \mu(x)$, generates random binary polynomials $\rho_B(x)$ of the degree $s-1$ and $\eta_B(x)$ of the degree s such that $\eta_B(x) \neq \mu(x)$ and encrypts each data block $C_{Ai} \bmod \mu(x)$ in accordance with the formula

$$C_{Bi} = \{\eta_B(x)[\eta_B^{-1}(x)C'_{Bi} \bmod \mu(x)] + \mu(x)[\mu^{-1}(x)\rho_B(x) \bmod \eta_B(x)]\} \\ \bmod \mu(x)\eta_B(x), \tag{2}$$

where $i = 1, 2, \ldots, z$. Then Bob sends the ciphertext

$$C_B = (C_{B1}, C_{B2}, \ldots, C_{Bi}, \ldots, C_{Bz})$$

to Alice.

3. For each value $i = 1, 2, \ldots, z$ Alice computes the value $C_{Bi} \bmod \mu(x) = M_i^{e_A e_B} \bmod \mu(x)$, generates random binary polynomials $\rho'_A(x)$ of the degree $s-1$ and $\eta'_A(x)$ of the degree s such that $\eta'_A(x) \neq \mu(x)$ and encrypts each data block $C_{Bi} \bmod \mu(x)$ in accordance with the formula

$$C'_{Ai} = \{\eta'_A(x)[\eta'^{-1}_A(x)C_{Bi}^{d_A} \bmod \mu(x)] + \mu(x)[\mu(x)^{-1}\rho'_A(x) \bmod \eta'_A(x)]\} \\ \bmod \mu(x)\eta'_A(x). \tag{3}$$

Then Alice sends the ciphertext

$$C'_A = (C'_{A1}, C'_{A2}, \ldots, C'_{Ai}, \ldots, C'_{Az})$$

to Bob. Bob discloses the secret message $M = (M_1, M_2, \ldots, M_i, \ldots, M_z)$ computing the values $C'_i = C'_{Ai} \bmod \mu(x) = M_i^{e_B} \bmod \mu(x)$ and $M_i = C'^{d_B}_i \bmod \mu(x)$ for $i = 1, 2, \ldots, z$.

In the described protocol for probabilistic encryption the ciphertexts C_A, C_B, and C'_A sent via public channel have size that is exactly two times larger than the size of the input data blocks M_i. Security of the protocol is provided due to good confusion and diffusion properties of the exponentiation operation and due to using the modulus $\mu(x)$ as secret key. It is worth to mention that in the case of sufficiently large size of the key $\mu(x)$ ($|\mu(x)| \geqslant 1024$ bits) the protocol resists attacks based on the known shared key, i.e., if the adversary gets the key $\mu(x)$ after the protocol have been performed, then he also will not be able to compute the secret message M. However after the key $\mu(x)$ becomes known for potential adversary the protocol will provide secrecy (in possible further use of the protocol) but not authentication.

3.2. Bi-deniable encryption scheme

Suppose Alice and Bob share a secret key representing the pair of mutually irreducible binary polynomials $\mu(x)$ and $\eta(x)$ of the degree $s = 128$ to 1024. To encrypt some secret message T Alice represents the message as sequence of the s-bit data blocks $T_i : T = (T_1, T_2, \ldots, T_i, \ldots, T_z)$. To provide bi-deniability of encrypting the secret message T they can use the following three-pass protocol.

1. Alice generates some fake message $M = (M_1, M_2, \ldots, M_i, \ldots, M_z)$ represented as sequence of s-bit data blocks and two local keys (e_A, d_A) and $(\varepsilon_A, \delta_A)$ such that $d_A = e_A^{-1} \bmod 2^s - 1$ and $\delta_A = \varepsilon_A^{-1} \bmod 2^s - 1$.

Then for each value $i = 1, 2, \ldots, z$ she computes the ciphertext block C_{Ai} as follows:

1.1. Compute the intermediate ciphertext blocks $C_{Ai}^{(M)}$ and $C_{Ai}^{(T)}$:

$$C_{Ai}^{(M)} = M_i^{e_A} \bmod \mu(x) \quad \text{and} \quad C_{Ai}^{(T)} = T_i^{\varepsilon_A} \bmod \eta(x).$$

1.2. Compute the $(2s)$-bit ciphertext block C_{Ai} as solution of the system of congruences

$$\begin{cases} C_{Ai} \equiv C_{Ai}^{(M)} \bmod \mu(x) \\ C_{Ai} \equiv C_{Ai}^{(T)} \bmod \eta(x) \end{cases} \tag{4}$$

Then Alice sends the ciphertext $C_A = (C_{A1}, C_{A2}, \ldots, C_{Ai}, \ldots, C_{Az})$ to Bob.

2. Bob generates two local keys (e_B, d_B) and $(\varepsilon_B, \delta_B)$ such that $d_B = e_B^{-1} \bmod 2^s - 1$ and $\delta_B = \varepsilon_B^{-1} \bmod 2^s - 1$. Then for each value $i = 1, 2, \ldots, z$ he computes the ciphertext block C_{Bi} as follows:

2.1. Compute the intermediate ciphertext blocks $C_{Ai}^{(M)}$ and $C_{Ai}^{(T)}$:

$$C_{Ai}^{(M)} = C_{Ai} \bmod \mu(x) \quad \text{and} \quad C_{Ai}^{(T)} = C_{Ai} \bmod \eta(x).$$

2.2. Compute the intermediate ciphertext blocks $C_{Bi}^{(M)}$ and $C_{Bi}^{(T)}$:

$$C_{Bi}^{(M)} = \left(C_{Ai}^{(M)}\right)^{e_B} \bmod \mu(x) = M_i^{e_A e_B} \bmod \mu(x) \text{ and}$$
$$C_{Bi}^{(T)} = \left(C_{Ai}^{(T)}\right)^{\varepsilon_B} \bmod \eta(x) = T_i^{\varepsilon_A \varepsilon_B} \bmod \eta(x).$$

2.3. Compute the $(2s)$-bit ciphertext block C_{Bi} as solution of the system of congruences

$$\begin{cases} C_{Bi} \equiv C_{Bi}^{(M)} \bmod \mu(x) \\ C_{Bi} \equiv C_{Bi}^{(T)} \bmod \eta(x). \end{cases} \tag{5}$$

Then Bob sends the ciphertext $C_B = (C_{B1}, C_{B2}, \ldots, C_{Bi}, \ldots, C_{Bz})$ to Alice.

3. Then for each value $i = 1, 2, \ldots, z$ Alice computes the ciphertext block C'_{Ai} as follows:

3.1. Compute the intermediate ciphertext blocks $C_{Bi}^{(M)}$ and $C_{Bi}^{(T)}$: $C_{Bi}^{(M)} = C_{Bi} \bmod \mu(x)$ and $C_{Bi}^{(T)} = C_{Bi} \bmod \eta(x)$.

3.2. Compute the intermediate ciphertext blocks$C'^{(M)}_{Ai}$ and $C'^{(T)}_{Ai}$: $C'^{(M)}_{Ai} = \left(C_{Bi}^{(M)}\right)^{d_A} \bmod \mu(x) = M_i^{e_B} \bmod \mu(x)$ and $C'^{(T)}_{Ai} = \left(C_{Bi}^{(T)}\right)^{\delta_A} \bmod \eta(x) = T_i^{\varepsilon_B} \bmod \eta(x)$.

3.3. Compute the $(2s)$-bit ciphertext block C'_{Ai} as solution of the system of congruences

$$\begin{cases} C'_{Ai} \equiv C'^{(M)}_{Ai} \bmod \mu(x) \\ C'_{Ai} \equiv C'^{(T)}_{Ai} \bmod \eta(x). \end{cases} \tag{6}$$

Then Alice sends the ciphertext $C'_A = (C'_{A1}, C'_{A2}, \ldots, C'_{Ai}, \ldots, C'_{Az})$ to Bob.

Bob discloses the secret message $T = (T_1, T_2, \ldots, T_i, \ldots, T_z)$ computing the values $C'^{(T)}_{Ai} = C'_{Ai} \bmod \eta(x) = T_i^{\varepsilon_B} \bmod \eta(x)$ and $T_i = \left(C'^{(T)}_{Ai}\right)^{\delta_B} \bmod \eta(x)$ for $i = 1, 2, \ldots, z$.

Respectively, Bob discloses the fake message M computing the values $C'^{(M)}_{Ai} = C'_{Ai} \bmod \mu(x) = M_i^{e_B} \bmod \mu(x)$ and $M_i = \left(C'^{(M)}_{Ai}\right)^{d_B} \bmod \mu(x)$.

When being coerced simultaneously, Alice and Bob open the fake message $M = (M_1, M_2, \ldots, M_i, \ldots, M_z)$, the shared key $\mu(x)$, and their local keys (e_A, d_A) and (e_B, d_B).

They also declare about using the three-pass probabilistic-encryption protocol described in Subsection 3.1. Distinguishing the bi-deniable encryption protocol from the probabilistic encryption protocol is a computationally difficult problem, therefore the protocol described in Subsection 3.2 provides bi-deniability.

4. Disscusion

Different variants of the protocols described in Section 2 can be constructed using different variants of the commutative cipherE_K with the single-use key K, and/or different public encryption algorithms.

For example, the encryption procedure E_K can be defined with formula

$$C = M * K,$$

where $*$ is one of the following operations: modulo $2^{|M|}$ addition (subtraction), modulo n addition (subtraction), modulo n multiplication. Instead of the RSA public encryption algorithm one can use the ElGamal algorithm [5]. The last modification is interesting from practical point of view since it gives possibility to provide more secure encryption in the case of implementing the ElGamal public-encryption algorithm with using elliptic curves [10]. Indeed, in the last case one can provide exponential security of the deniable encryption and higher performance. Besides the ElGamal algorithm is probabilistic in its nature. Using the Rabin public-encryption algorithm [14] is also possible, but not so attractive.

One can note that the second flexible public key DE protocol from Section 2 resists the coercive attack on the sender or on the receiver, but it does not resist coercive attack performed simultaneously on the both parties. Indeed, resistance to last attack means that the sender and the receiver select the same fake message, however to have such possibility they need some pre-agreed information that indicates what fake message should be selected. Appropriate modification of the source protocol is possible, however it becomes a plan-ahead DE protocol that has no evident advantages as compared with protocols of such type introduced in [16, 17].

As compared with the flexible sender-side DE protocols [1, 8] in which the message is encrypted consecutively bit by bit (each bit is sent in form of the $|n|$-bit pseudorandom number, $|n| > 1024$) in the proposed protocols the message is transformed as a single data block that provides significantly higher performance. Besides, the proposed protocols provide simple and very fast procedure (performing one XOR operation) for computing the fake random input (sender's local key) connected with the fake message.

The bi-deniable encryption scheme presented in Section 3 uses the Pohlig-Hellman modulo-exponentiation cipher represented in a specific form in which the modulus that is the binary polynomial $\mu(x)$ serves as shared key. Therefore such implementation provides sufficiently high security even in the case when binary polynomial $\mu(x)$ has sufficiently small degree ($128 \leqslant s \leqslant 768$). If the modulus $\mu(x)$ has high degree ($s \geqslant 1024$), the protocol becomes resistant to the known-key attacks. However, if the shared key is compromised the protocol will not provide authentication, like in the case of the probabilistic-encryption algorithm from Subsection 3.1.

Resistance to the simultaneous coercive attacks on Alice and Bob is provided due to fact that Bob using the fake key is able to disclose correctly the fake message M generated by Alice at the first step of the protocol. Besides, the ciphertexts C_A, C_B, and C'_A computed at steps 1, 2, and 3, correspondingly, look like the ciphertexts produced during performing the probabilistic-encryption protocol from subsection 3.1, when M serves as input message. In other words the proposed bi-deniable encryption protocol is computationally indistinguishable from the proposed probabilistic encryption protocol for the coercer intercepting the ciphertexts C_A, C_B, and C'_A sent via communication channel. Indeed, computation of each block C_{Ai} of the ciphertex C_A in accordance with formula (1) gives solution of the following system relatively unknown C_{Ai}

$$\begin{cases} C_{Ai} \equiv M_i^{e_A} \bmod \mu(x) \\ C_{Ai} \equiv \rho_A(x) \bmod \eta_A(x). \end{cases}$$

The first congruence coincide with the first congruence in system (4) and for given value C_{Ai} and arbitrary $\eta_A(x)$ of the degree s such that $\eta_A(x) \neq \mu(x)$ we have one value $\rho_A(x)$ that satisfies the second congruence of the last system (namely, $\rho_A(x) = C_{Ai} \bmod \eta_A(x)$).

Computation of each block C_{Bi} of the ciphertex C_B in accordance with formula (2) gives solution of the following system relatively unknown C_{Bi}

$$\begin{cases} C_{Bi} \equiv M_i^{e_A e_B} \bmod \mu(x) \\ C_{Bi} \equiv \rho_B(x) \bmod \eta_B(x). \end{cases}$$

The first congruence coincide with the first congruence in system (5) and for given value C_{Bi} and arbitrary $\eta_B(x)$ of the degree s such that $\eta_B(x) \neq \mu(x)$ we have one value $\rho_B(x)$ that satisfies the second congruence of the last system (namely, $\rho_B(x) = C_{Bi} \bmod \eta_B(x)$).

Computation of each block C'_{Ai} of the ciphertex C_A in accordance with formula (3) gives solution of the following system relatively unknown C'_{Ai}

$$\begin{cases} C'_{Ai} \equiv (M_i)^{e_B} \bmod \mu(x) \\ C'_{Ai} \equiv \rho'_A(x) \bmod \eta'_A(x). \end{cases}$$

The first congruence coincide with the first congruence in system (6) and for given value C'_{Ai} and arbitrary $\eta'_A(x)$ of the degree s such that $\eta'_A(x) \neq \mu(x)$ we have one value $\rho'_A(x)$ that satisfies the second congruence of the last system (namely, $\rho'_A(x) = C'_{Ai} \bmod \eta'_A(x)$).

Thus, the ciphertexts C_A, C_B, and C'_A produced during performing the bi-deniable encryption protocol could be produced while performing the probabilistic-encryption protocol. To prove the ciphertexts were produced with the three-pass protocol for simultaneous encryption of two messages M and T, the coercer has to disclose the secret message from the ciphertexts, however this seems to be a computationally infeasible problem.

A possible modification of the protocols from Section 3 can be get with using the binary polynomials $\eta_A(x)$, $\eta_B(x)$, $\eta'_A(x)$, and $\eta(x)$ having degree $s' < s$ (the message T is to be divided into s'-bit data blocks T_i). In the modified protocols the ciphertext blocks have size $s + s' < 2s$ and for smaller values s' applying the bi-deniable encryption protocol looks more believably as applying the probabilistic-encryption protocol. In the case of probabilistic-encryption protocol one can use sufficiently small values $s' = 4$ to 64. In the case of the bi-deniable encryption protocol one has some restriction: $64 < s' < s$, where $s = 128$ to 1024. This restriction is connected with using the value $\eta(x)$ as shared secret key.

Indeed, to provide deniability the value $\eta(x)$ and the local key ε_A ($\varepsilon_A < 2^{s'}$) should be sufficiently large, for example, $|\eta(x)| + |\varepsilon_A| \geqslant 128$ bits. For small values s' (for example, for $s' = 4$ to 16) the coercer using the values $\mu(x)$ and e_A (that are to be opened in the case of coercive attack) can find easily the secret values $\eta(x)$ and ε_A with help of the exhaustive-search method.

5. Conclusion

There have been proposed sender-deniable, sender&receiver-deniable, probabilistic, and bi-deniable encryption schemes representing three-pass protocols based on using commutative ciphers. The probabilistic-encryption protocol has been designed as protocol associated with the bi-deniable encryption protocol, however it has independent practical interest. To get higher performance of the bi-deniable encryption protocol one can design its modification on the base of commutative encryption operation implemented as multiplying points of elliptic curves defined over finite fields $GF(p)$ and $GF(2^s)$ [15].

The last remark can be attributed also to the design of two flexible public key DE protocols from section 2 in the case of using the ElGamal public encryption algorithm (that is a probabilistic one) in frame of the protocols. For such protocols, besides higher performance, such variants of the flexible sender-deniable and sender&receiver-deniable public encryption DE protocols will provide exponential resistance to coercive attacks in the case of implementing the ElGamal-like algorithm on the base of elliptic curves.

Another interesting research problem is connected with using the commutative encryption functions to design no-key DE protocols.

References

[1] **M.T. Barakat**, *A new sender-side public-key deniable encryption scheme with fast decryption.* KSII Transactions on Internet and Information Systems **8** (2014), 3231−3249.

[2] **R. Canetti, C. Dwork, M. Naor, R. Ostrovsky**, *Deniable encryption*, Lecture Notes Comp. Sci. **1294** (1997), 394−104.

[3] **D. Dachman-Soled**, *On the impossibility of sender-deniable public key encryption*, IACR Cryptology ePrint Archive (2012), 727.

[4] **D.Dachman-Soled**, *On minimal assumptions for sender-deniable public key encryption*, Lecture Notes Comp. Sci. **8383** (2014), 574−591.

[5] **T. ElGamal**, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Information Theory **IT−31** (1985), 469−472.

[6] **J. Gordon** *Strong primes are easy to find*, Lecture Notes Comp. Sci. **209** (1985), 216−223.

[7] **M.E. Hellman, S.C. Pohlig**, *Exponentiation cryptographic apparatus and method*, U.S. Patent # 4,424,414. 3 Jan. 1984.

[8] **M.H. Ibrahim**, *A method for obtaining deniable Public-Key Encryption*, International J. Network Security **8** (2009), 1−9.

[9] **Yu. Ishai, E. Kushilevits, R. Ostrovsky**, *Efficient non-interactive secure computation*, Lecture Notes Comp. Sci. **6632** (2011), 406−425.

[10] **N. Koblitz**, *Elliptic curve cryptosystems*, Math. Computat. Advances **48** (1987), 203−209.

[11] **D. Markus, D.M. Freeman**, *Deniable encryption with negligible detection probability; An interactive construction* Lecture Notes Comp. Sci. **6632** (2011), 610 − 626.

[12] **B. Meng**, *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. Networks **4** (2009), 370−377.

[13] **B. Meng, W.J. Qing**, *A receiver deniable encryption scheme*, Proc. Internat. Symposium on Information (2009), 254−257.

[14] **A.J. Menezes, P.C. Oorschot, S.A. Vanstone**, Applied cryptography. CRC Press, New York, London, 1996.

[15] **A.J. Menezes, S.A. Vanstone**, *Elliptic curve cryptosystems and their implementation*, J. Cryptology **6** (1993), 209–224.

[16] **A.A. Moldovyan, N.A. Moldovyan**, *Practical method for bi-deniable public-key encryption*, Quasigroups and Related Systems **22** (2014), 277–282.

[17] **A.A. Moldovyan, N.A. Moldovyan, V.A. Shcherbacov**, *Bi-Deniable public-key encryption protocol secure against active coercive adversary*, Bul. Acad. Stii. Republ. Moldova, Matematica **3(76)** (2014), 23 – 29.

[18] **A.A. Moldovyan, D.N. Moldovyan, V.A. Shcherbacov**, *Stream deiable-encryption algorithm satisfying criterion of the computational indistinguishability from probabilistic ciphering*, Computer Sci. J. Moldova **24** (2016), 68–82.

[19] **N.A. Moldovyan, A.A. Moldovyan, V.A. Shcherbacov**, *Provably sender-deniable encryption scheme*, Computer Sci. J. Moldova **23** (2015), 62–71.

[20] **N.A. Moldovyan, A.A. Moldovyan, V.A. Shcherbacov**, *Generating cubic equations as a method for public encryption*, Bul. Acad. Stii. Republ. Moldova, Matematica **3(79)** (2015), 60 – 71.

[21] **A. O'Neil, C. Peikert, B. Waters**, *Bi-deniable public-key encryption*, Lecture Notes Comp. Sci. **6841** (2011), 525–542.

[22] **J. Pieprzyk, T. Hardjono, J. Seberry**, *Fundamentals of Computer Security.* Springer-Verlag Berlin Heidelberg 2003.

[23] **R.L. Rivest, A. Shamir, L.M. Adleman**, *A method for obtaining digital signatures and public key cryptosystems*, Commun. ACM **21** (1978), 120–126.

[24] **A. Sahai, B. Waters**, *How to use indistinguishability obfuscation: Deniable encryption, and more*, IACR Cryptology ePrint Archive (2013), 454.

[25] **C. Wang, J. Wang**, *A shared-key and receiver-deniable encryption scheme over lattice*, J. Computat. Inform. Systems **8** (2012), 747–753.

Received September 23, 2016

N.A. Moldovyan
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
14 Liniya 39, St.Petersburg 199178, Russia
Email: nmold@mail.ru

A.V. Shcherbacov
Theoretical Lyceum C. Sibirschi, Lech Kaczynski str. 4, MD-2028 Chişinău, Moldova
Email: admin@sibirsky.org

M.A. Eremeev
Mozhaisky Military Space Academy, Zhdanovskaya st. 13, St.Petersburg 197198, Russia
Email: mae1@rambler.ru