Bi-deniable Public-Encryption Protocols Based on Standard PKI

Nikolay Moldovyan[1], Andrey Berezin[2],
Anatoly Kornienko[3], Alexander Moldovyan[4]
[1] Saint Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
[2]Saint Petersburg Electrotechnical University "LETI"
[3]Petersburg State Transport University
[4]ITMO University
St. Petersburg, Russian Federation
nmold@mail.ru, a.n.berezin.ru@gmail.com, {kaa.pgups,maa1305}@yandex.ru

***Abstract*—The paper proposes new deniable encryption protocols providing bi-deniability in the case of both the passive coercive attack and the active one. It is supposed the coercive adversary intercepts all data sent during the protocol. The protocols use no shared secret keys that are pre-agreed by parties of the protocol. Bi-deniability is based on computational indistinguishability between deniable encryption and probabilistic one. Significant merit of the proposed protocols is their using the standard public key infrastructure.**

I. Introduction

The regular encryption schemes provide very high security against known-plaintext and chosen text attacks, therefore they are widely used to protect information sent via telecommunication channels from unauthorized access. However, in real world sometimes an adversary (coercer) has power to force a user to open all secret keys. This can take place due to a criminal action or law enforcement procedure. Such types of attacks are called coercive attacks (see Fig. 1 and Fig. 2). To provide security against such attacks it was proposed a notion of *public-key deniable encryption*, by R. Canetti et. al [1]. The deniable encryption schemes are classified according to which parties of the communication session may be coerced: sender-deniable, receiver-deniable, sender- and receive-deniable (bi-deniable) schemes in which coercive adversary attacks the user sending message, the user receiving message, and the both users, correspondingly. Deniable encryption is a powerful notion for both the practice and the theory. Its practical applications relate to prevention of the vote buying in the internet-voting systems [14], [16], to providing secure multi-party computations [8], and to providing information secrecy with practical methods of the public-key deniable encryption [20], [15], [12].

The common idea of the public-key deniable encryption schemes is potential possibility to decrypt the ciphertext c in different ways, while using the private key corresponding to the public one with which the secret message t has been encrypted. Such possibility is due to using a random value r in the procedure of encrypting the secret message t. The public-key encryption can be represented with the formula $c = E_P(t, r)$, where P is a public key. While being coerced the sender of the message can open a fake message m with another random value $r' \neq r$ such that $c = E_P(m, r')$. The fake random value r' can be computed with some faking algorithm that is a part of the deniable encryption scheme. Input of the faking algorithm F_P, parameterized with the public-key value P, is the pair (c, m), i.e. $r' = F_P(c, m)$. The fake message can be selected arbitrary while the sender or the receiver of the ciphertext are coerced. If both parties are coerced simultaneously, then they are to have possibility to select the same fake message. To decrypt the secret message t the receiver of the cryptogram c is to use the same random value r as that used by the sender. In some deniable encryption schemes the fake message m is planed ahead, i.e. the message m is selected before performing the encryption process. Such schemes are called plan ahead deniable encryption schemes. To provide bi-deniability the last schemes are composed so that the sender and the receiver of the secret message t open to coercer the same fake message m (see Fig. 3) and show that encrypting m results in the ciphertext c and decrypting c outputs m. The schemes that are free from using any shared key and from using many interactive passes of the sending message protocol are attractive for practical application. One of such schemes is proposed in [15]. However that scheme does not provide bi-deniability. In the known literature devoted to design of the deniable encryption schemes usually it is considered the passive coercive attack, consideration of the model of the active coercer is actual though. For example, suppose the coercer impersonates the sender in the deniable encryption protocol and after sending the ciphertext he attacks the receiver. If the receiver opens a fake message, then the attack is successful, since the lie of the receiver is disclosed. It is a common assumption that coercion is performed after the ciphertext has been sent.

In this paper we present a new protocol for plan-ahead public-key bi-deniable encryption that can be practically implemented using international standards ISO/IEC 15946-1:2008, FIBS 186-4, GOST R 34.10-2012 and the exiting public key infrastructure (PKI) without any modification [11], [2], [3]. Besides, bi-deniability is provided against the coercive adversary that knows the ciphertext and all data sent via communication channel during the process of performing the protocol. Active coercive attacks are prevented due to including the entity authentication mechanism in the protocol. Then in the frame of the proposed approach it is designed the bi-deniable protocol based on using the RSA public-encryption

algorithm.

The paper organized as follows. Section 2 describes the model of coercive attack and design criteria. Section 3 describes the proposed bi-deniable encryption protocol. Section 4 presents discussion. Section 5 concludes the paper.

II. Model of coercive attack and design criteria

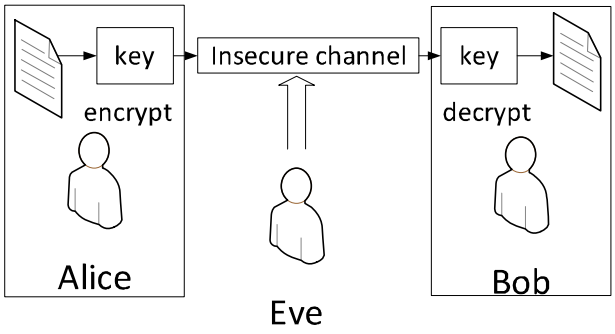

Fig. 1. Model of classic attack

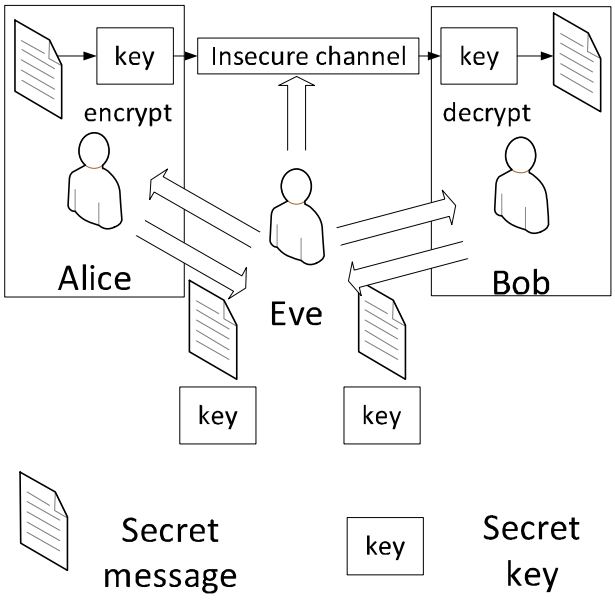

Fig. 2. Model of coercive attack

It is supposed that the coercive adversary can perform both the passive attacks and the active ones. In the case of passive attack he can read all data from communication channel. In the case of active attack the coercive adversary plays a role of sender or receiver of the message. *After ciphertext has been sent*, the coercive adversary has possibility to force both parties to open the following:

1) the private key of the receiver;
2) the private key of the sender;
3) the plaintext corresponding to the ciphertext;
4) the decryption algorithm output of which depends on each bit of the ciphertext;
5) the encryption algorithm.

To resist such attacks we propose the following design criteria for constructing a public-key deniable encryption protocol:

1) the scheme should perform authentication of the sender and receiver as its internal sub procedure;
2) the deniable encryption should be performed with using only receiver's public key and random values;
3) each bit of the fake message should depend on each bit of the ciphertext;
4) a probabilistic public-key encryption algorithm should be associated with the deniable encryption algorithm and the ciphertext generated by the last algorithm should be computationally indistinguishable from the ciphertext generated by the first one.

The last item serves as also as a method for justifying that the size of ciphertext is larger than the size of fake message: the parties of the protocol use probabilistic encryption to provide more secure communication.

III. Proposed method

Let us consider the case in which Alice wants to send a secret message t to Bob and provide resistance to passive and active coercive attacks. The idea of our method is illustrated by the following generalized protocol:

1) Bob generates a random value r_B that serves as his single-use public key and sends the value r_B to Alice.

2) Alice generates a random value r_A that serves as her single-use public key and computes her signature $S_A = Sign_A(r_A||r_B)$ to $r_A||r_B$, where $||$ is the concatenation operation. Then she sends the values r_A and S_A to Bob.

3) Bob verifies validity of the signature S_A to $r_A||r_B$. If the signature is invalid, then he stops the protocol, otherwise he computes his signature $S_B = Sign_B(r_A||r_B)$ to $r_A||r_B$ and sends it to Alice.

4) Alice verifies Bob's signature S_B to $r_A||r_B$, if it is invalid she terminates communication session. Otherwise she generates a fake message m and encrypts simultaneously the messages t and m with using Bob's public key, random value r_B and secret connected with random value r_A. The produced ciphertext c coincides with the cryptogram generated by some probabilistic encryption of the fake message m with using Bob's public key and some random value r'. Then she computes her signature $S_C = Sign_A(c)$ to ciphertext and send S_C and c to Bob.

5) Bob verifies Alice's signature to the received ciphertext. If the signature is invalid he reject the ciphertext. Otherwise he decrypts the cryptogram using the secret connected with the random value r_B and discloses the secret message t.

When being coerced Bob decrypts the ciphertext using his private key and opens the fake message m.

For practical implementation of the proposed deniable encryption scheme we propose to use elliptic-curve digital signature standards (ISO/IEC 15946-1:2008 [11], FIBS 186-4 [2], GOST R 34.10-2012 [3]) for performing signature generation and verification procedures as well as already existing PKI.

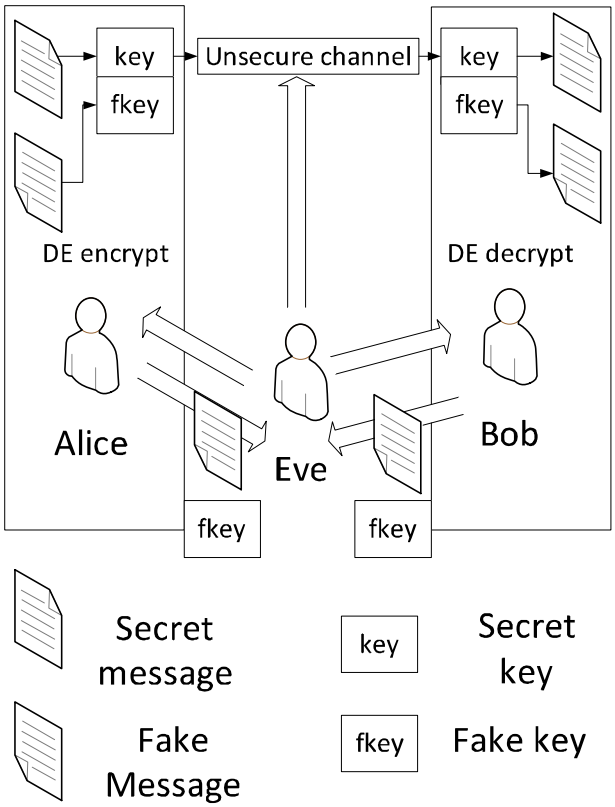

Fig. 3. Coercive attack on deniable encryption protocol

In the algorithm for performing simultaneous encryption of the fake message m and the secret message t we use some cryptographic hash function $h(\cdot)$. Actually it is possible to use any secure hash-function (for example, SHA-3 [10] or hash-function from ISO/IEC 10118-3:2004 [9]).

The size of the secret message t and the fake message m is to be less than size of some prime p that serves as a public domain parameter. If the size of the messages t and m is large, then the messages are to be divided into some data blocks before computing the ciphertext.

a) Signature scheme and public key agreement protocol using computation on an elliptic curve: The digital signature standard GOST R 34.10-2012 [3] specifies a signature scheme based on elliptic curves (ECs) over finite field $GF(p)$, where p is a prime (for details of the application of the EC's in cryptography see [13], [17]. The standard specifies using EC described by the following equation

$$y^2 = x^3 + ax^2 + b \bmod p, \tag{1}$$

where coefficients a and b are selected so that the EC order contains a large prime factor q (having size 256 to 512 bits). Points of the EC are pairs of numbers x and y ($0 < x < p$, $0 < y < p$) called abscissa and ordinate, which satisfy equation (1).

Such EC represents a commutative finite group with the point addition operation as the group operation. The multiplication of some EC point A by number m is defined as $kA = A + A + \cdots + A$ (k times). The neutral element of the group of the EC points is the point in infinity denoted O. On definition it is assumed $A + O = O + A = A$ and $mO = O$.

The addition of the points $A = (x_A, y_A)$ and $B = (x_B, y_B)$ is performed with the following formulas for computing the abscissa x_R and ordinate y_R of the point $R = A + B$:

$$\begin{array}{rcl} x_R & = & \lambda^2 - x_A - x_B \bmod p \\ y_R & = & \lambda(x_A - x_R) - y_A \bmod p \end{array}, \text{ where}$$

$$\lambda = \begin{cases} \frac{y_B - y_A}{x_B - x_A} \bmod p, & \text{if } A \neq B \\ \frac{3x_A^2 + a}{2y_A} \bmod p, & \text{if } A = B. \end{cases}$$

Subtraction of the points B and $A = (x_A, y_A)$ is defined as follows $B - A = B + (-A)$, where $-A = (x_A, -y_A)$.

In GOST R 34.10-2012 [3] the public key is some EC point Q computed as follows $Q = dG \bmod q$, where d is the secret key and G is the EC point having the order q. The signature to some message μ is generated as follows:

1) Generate a random value k, compute the point $R = kG$ and define $r = x_R$. The value r is the first element of the signature.

2) Using the hash function F_h specified by the Russian standard GOST R 34.11-2012 [4] compute the hash value h from the message μ: $h = F_h(\mu)$. Then it is computed value $e = h \bmod q$.

3) Using the secret key compute the value $s = ke + dr \bmod q$, which is the second element of the signature.

Verification of the signature (r, s) to the message μ is performed as follows:

1) Compute the hash value h from the message μ: $h = F_h(M)$. Then compute $e = h \bmod q$.

2) Compute the point $R^* = \left(e^{-1}s \bmod q\right) G - \\ - \left(e^{-1}r \bmod q\right) Q$.

3) Compare the values x_{R^*} and r. If $x_{R^*} = r$, then the signature is valid. Otherwise the signature is rejected.

The public key agreement protocol using computations on EC, in which two users (Bob and Alice) generate a common secret value z, looks as follows (see Fig. 4).

1) Alice generates his private key $d_A < q$ and computes his public key $Q_A = d_A G$ and sends the point Q_A to Bob.

2) Bob generates his private key $d_B < q$ and computes his public key $Q_B = d_B G$ and sends the point Q_B to Alice.

3) Alice computes the common secret point $Z_{AB} = d_A Q_B = d_A d_B G$.

4) Bob computes the same secret point $Z_{AB} = d_B Q_A = d_B d_A G$.

Having computed the same secret point Z_{AB} Alice and Bob can take its abscissa as the common secret value $z = x_{Z_{AB}}$.

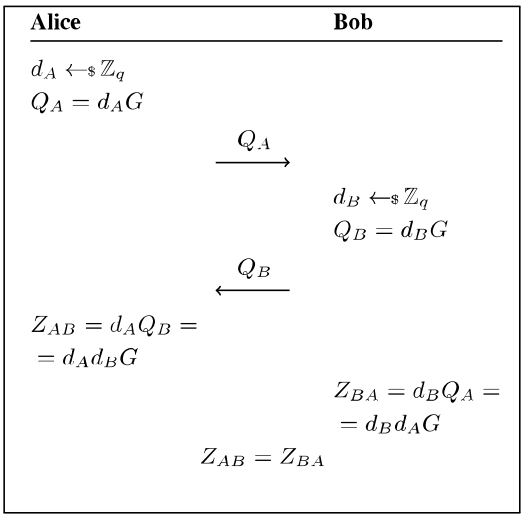

Fig. 4. The public key agreement protocol on EC

b) Mapping the point on elliptic curve into the integer number and vice versa: There are two points (x, y) and $(x, p - y)$, where $y < p$, on elliptic curve with the same abscissa. Therefore it is possible to use the abscissa of the point concatenated with an extra bit as an integer number u. The extra bit with a value equal 1 defines the upper point (the point (x, y), if $y > p-y$, or the point $(x, p-y)$, if $y < p-y$), in another case he defines the lower point. Such method provides a possibility to convert easily the point into the number and vice versa. Thus, to send a point via communication channel it is sufficient to send its abscissa with an extra bit.

c) Associated probabilistic encryption algorithm: Suppose Alice wants to send a secret message $t < q$ to Bob. She can use the following probabilistic encryption algorithm.

1) Bob generates a random integer R_B satisfying condition $1 < k_A < q - 1$ and sends it to Alice.

2) Alice generates a random integer k_A satisfying condition $1 < k_A < q - 1$ and computes a random point $R_A = k_A G$. Then she generates her signature $S_A = Sign_A(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$ and sends the signature and the point R_B to Bob.

3) Bob computes his signature $S_B = Sign_B(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$ and sends his signature to Alice.

4) Alice verifies the signature $S_B = Sign_B(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$. If the signature is invalid she terminates the communication session. Otherwise she generates a random integer r' and ω satisfying condition $1 < k_A < q - 1$.

5) Compute the point $W = \omega G$.

6) Using Bob's public key Q_B Alice computes the point $Z = \omega Q_B$.

7) Map the point Z into the number z.

8) Solve the following system of linear congruences with unknowns c_1 and c_2:

$$\begin{cases} zc_1 + zh(Z)c_2 = m \bmod q \\ c_1 = r'c_2 \bmod q \end{cases}$$

9) Send to Bob the ciphertext C represented by triple (W, c_1, c_2).

The described protocol is illustrated in Fig. 5. For some value r' the associated algorithm generates the ciphertext that coincide with the ciphertext generated by the following deniable encryption protocol (see Fig. 6).

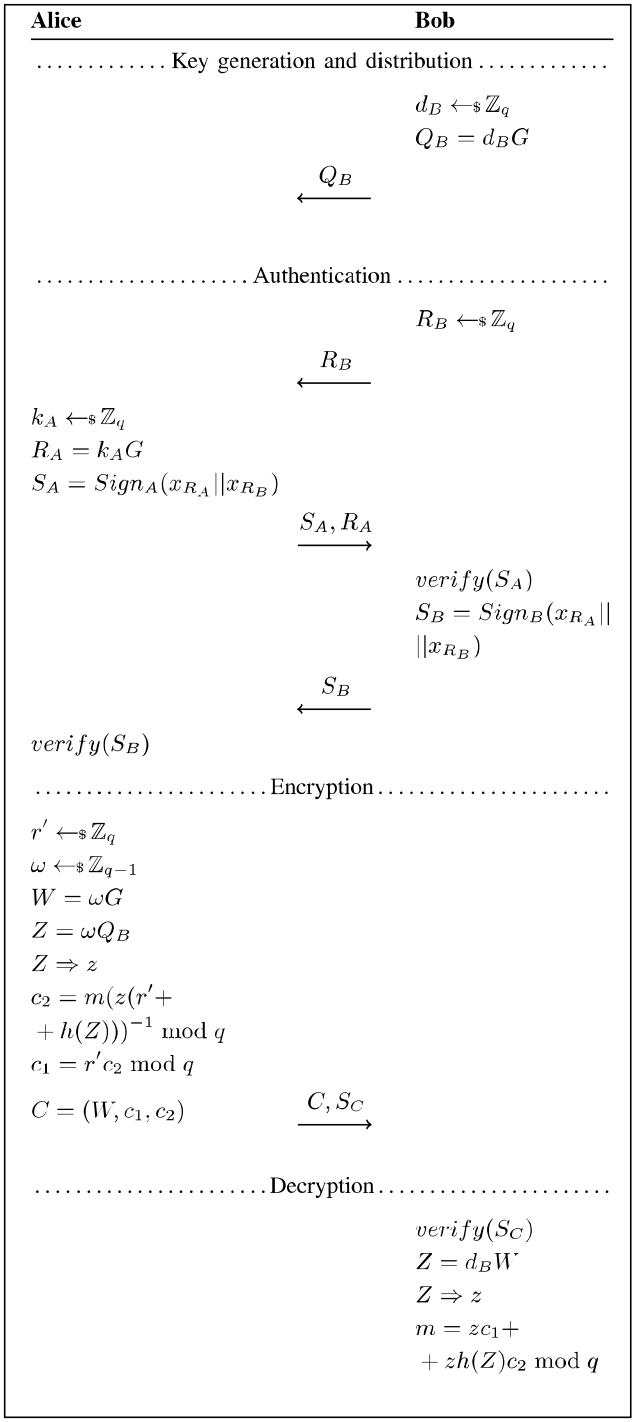

Fig. 5. The probabilistic encryption algorithm associated with deniable encryption protocol

d) Proposed public-key deniable encryption protocol: Suppose Alice wants to send a secret message $t < q$ to Bob. She can use the following bi-deniable encryption protocol.

1) Bob generates a random integer k_B satisfying condition $1 < k_B < q - 1$ and computes a random point $R_B = k_B G$ and sends the point R_B to Alice.

2) Alice generates a random integer k_A satisfying condition $1 < k_A < q - 1$ and computes a random point $R_A = k_A G$. Then she generates her signature $S_A = Sign_A(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$ and sends the signature and the point R_B to Bob.

3) Bob computes his signature $S_B = Sign_B(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$ and sends his signature to Alice.

4) Alice verifies the signature $S_B = Sign_B(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$. If the signature is invalid she terminates the communication session. Otherwise she generates a fake message $m < q$ and a random integer value ω from the interval $[1, q-1]$ and computes the point $W = \omega G$. Using Bob's public key Q_B and the point R, Alice computes the points $Z = \omega Q_B$ and $Z' = k_A R_B$. Then, she maps the points Z, Z' into the integer number z, z' accordingly and solves the following system of linear congruences with unknowns c_1 and c_2:

$$\begin{cases} zc_1 + zh(Z)c_2 = m \bmod q \\ z'c_1 + z'h(Z')c_2 = t \bmod q \end{cases}$$

5) Then Alice computes her signature $S_C = Sign_A(c_1||c_2)$ and sends S_C and the ciphertext $C = (W, c_1, c_2)$ to Bob.

6) Bob verifies Alice's signature $S_C = Sign_A(c_1||c_2)$ to the concatenation $c_1||c_2$. If it is invalid, then he rejects the ciphertext. Otherwise he computes the point $Z' = k_B R_A$, maps the point Z' into the integer z', and opens the secret message t as follows $t = z'c_1 + z'h(Z')c_2 \bmod q$.

When being coerced Bob opens his private key d_B and decrypts the cryptogram into the fake message m. Namely, he uses the received point W to compute the point $Z = d_B W$ and maps the point Z into the integer z. Then he computes the fake message $m = zc_1 + zh(Z)c_2 \bmod q$.

IV. Discussion

The proposed protocol implements all design criteria proposed in section 2:

1) in the protocol it is performed mutual authentication of the sender and receiver; the authentication is based on using digital signatures to random values and to the ciphertext;
2) the encryption procedure is performed using only Bob's public key, Alice's public key is used only to verify her signatures;
3) the probabilistic public-key encryption algorithm associated with the deniable encryption algorithm, using the random value $r' = c_1/c_2 \bmod q$ defines the probabilistic public-key encryption algorithm that generates the ciphertext that coincides with the ciphertext $C = (W, c_1, c_2)$ produced by the public-key deniable encryption procedure.

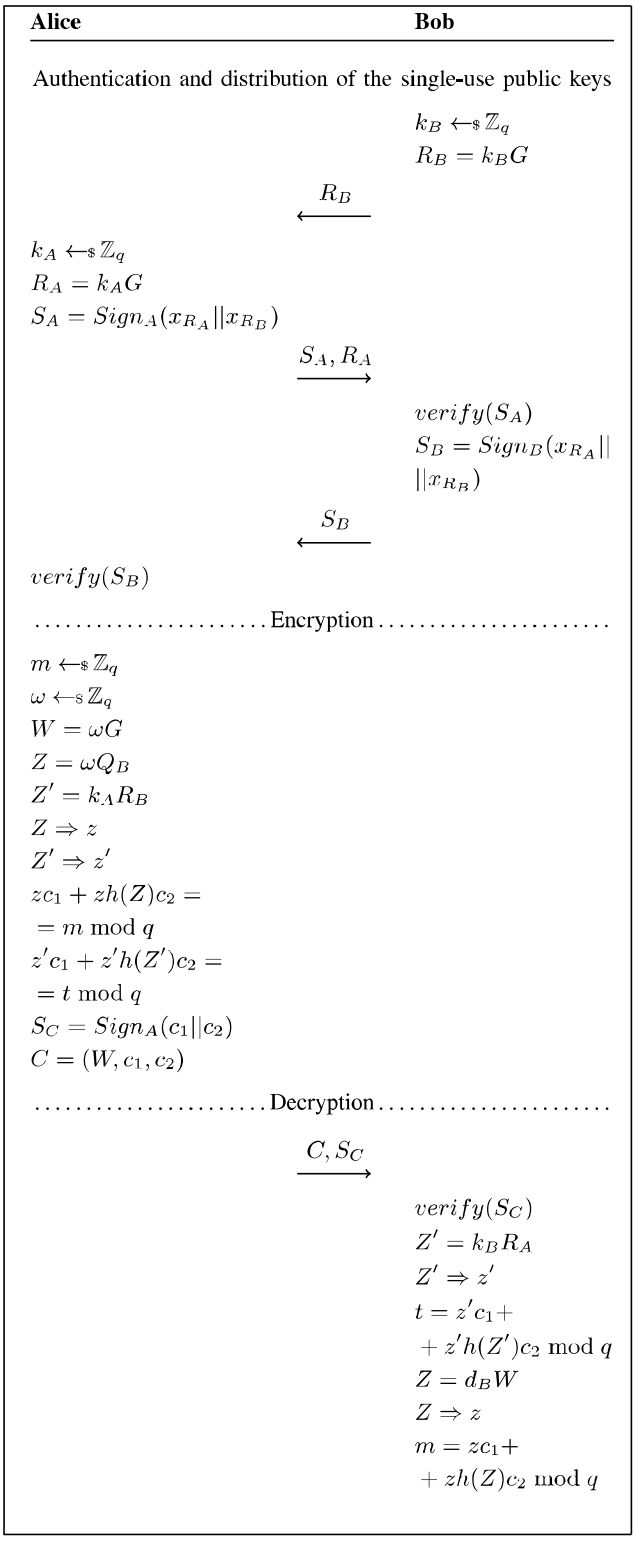

Fig. 6. The deniable encryption and decryption protocol

The sender authentication procedure allows to resist the coercive attacks in which the adversary tries to initiate the deniable encryption protocol as sender and trying to convict receiver is lying. Adversary can send a ciphertext containing both the secret and the fake messages and then to force the receiver to decrypt the ciphertext. If the receiver will not open the secret message known to adversary, then the last attack is considered as successful one, since the adversary is able to prove that the receiver is lying. However the adversary has no practical possibility to sign the random value $x_{R_A}||x_{R_B}$, where the value x_{R_B} depends on Bob's random choice, nor the value $c_1||c_2$ that is part of the ciphertext.

Also the adversary may try to participate as receiver and trying to convict sender is lying. It is obliviously that he can't generate the valid signature $Sign_B(x_{R_A}||x_{R_B})$ to the concatenation $x_{R_A}||x_{R_B}$, where the value x_{R_A} depends on random choice by Alice. If Alice receives invalid signature $Sign_B(x_{R_A}||x_{R_B})$, she terminates the communication session. In this case no ciphertext is sent. Suppose the adversary interferes in the protocol after the point $R_A = k_A P$ and Alice's signature to the value $x_{R_A}||x_{R_B}$ have been sent by Alice. In such case the adversary has possibility to send his own ciphertext, however he is not able to create a valid signature that is to be sent together with the ciphertext. At the last step of the protocol Bob verifies Alice's signature $Sign_A(c_1||c_2)$ to the value $c_1||c_2$ and rejects the ciphertext, if the signature is not valid. If Bob and Alice are attacked at this moment they open their private keys to the adversary and say they not be relevant to the ciphertext. Thus, the active attacks by coercive adversary are prevented due to entity authentication mechanism included in the proposed protocol.

In the case when Bob and Alice are simultaneously attacked by the passive coercive adversary they refer to the use of the probabilistic public-key encryption algorithm to encipher the message m (that is fake). They also refer to the use of the random values R_A and R_B in order to perform the entity authentication procedure. To refute this assertion the adversary should compute the value k_A such that $R_A = k_A G$ or the value k_B such that $R_B = k_B G$, i.e. he should solve the discrete logarithm problem on elliptic curve, which is computationally infeasible. Thus, in the case of two-side coercive attack Alice opens the fake message m and performs the associated public probabilistic encryption with using Bob's public key and random value r' which defines formation of the ciphertext C. (Opening Alice's private key after the ciphertext has been sent is useless for the coercer since it has not been used in computing the ciphertext.) Correspondingly, Bob opens his private key and show that decryption algorithm outputs the fake message m with using only his private key, each bit of the ciphertext C influencing each bit of m.

In comparison with the deniable encryption schemes described in papers [12], [6], [7], [16], [20] the main difference of the protocol proposed in the present paper consists in the following:

i) using the hidden public key agreement procedure inset in the entity authentication stage of the deniable encryption protocol;

ii) using the associated probabilistic public-key encryption algorithm for giving credibility of the fake message, instead of using a faking algorithm;

iii) providing bi-deniability (comparison in Table IV).

TABLE I. COMPARISON OF THE TYPICAL DENIABLE ENCRYPTION SCHEMES WITH OUR PROPOSED SCHEME ('*'- SCHEME HAS DENIABILITY WITH SOME CONDITIONS)

Scheme	Deniability		Efficiency
	Reciever	Sender	
[1]	+	+	low
[12]	+*		high
[6]		+	low
[7]	+*		low
[16]	+		high
[20]	+	+	low
our	+	+	high

From practical point of view it is sufficiently interesting to apply the proposed approach (characterized in embedding the entity authentication mechanism in the designed cryptoscheme) for designing the bi-deniable cryptoscheme using the RSA cryptoscheme to implement both the public encryption and the entity authentication. Next section is devoted to this problem.

V. BI-DENIABLE ENCRYPTION PROTOCOL BASED ON THE RSA PUBLIC ENCRYPTION ALGORITHM

a) Cryptosystem RSA: The RSA public-key cryptoscheme [21] is widely used for signing electronic documents and for public encryption. It is described as follows (see Fig 7). A user selects at random two sufficiently large primes r and q and generates his public key in form two numbers (n, e), where $n = rq$ and e is a random number that is relatively prime with Euler phi function $\phi(n) = (r-1)(q-1)$. Then he computes his private key $d = e^{-1} \bmod \phi(n)$. The values r and q are secret, however they are not used further. The public encryption of some digital message $M < n$ is performed using the public key as the computating the ciphertext $C = M^e \bmod n$. The decryption procedure can be performed using the private key connected with the public key (n, e) as follows $M = C^d \bmod n$. The digital signature S to the message M is computed using the formula $S = H^d \bmod n$, where $H = h(M)$ for some specified hash function $h(\cdot)$. The signature verification is performed using the formula $h(M) = S^e \bmod n$. If the last equation holds, then the signature is accepted as a valid one. Security of the RSA cryptoscheme is based on the difficulty of factoring the composite number n.

b) Deniable public encryption protocol based on RSA: Let Alice and Bob be users of the RSA cryptosystem. Suppose the following: the pair of numbers (n_A, e_A) is Alice's public key; d_A is her private key; (n_B, e_B) is Bob's public key; d_B is his private key.

Besides, Bob public key is such that the number $P = 2n_B + 1$ is prime and order of the number 3 is equal to $2n_B$ or n_B. Earlier primes with such structure were used in papers [18],[19]. The deniable public encryption protocol based on the RSA cryptosystem includes the following steps:

1) Alice selects a random number k_A and computes $R_A = 3^{k_A} \bmod P$ and sends the value R_A to Bob as her random choice.

2) Bob selects a random number k_B, computes the number $R_B = 3^{k_B} \bmod P$ and his signature S_B to the value

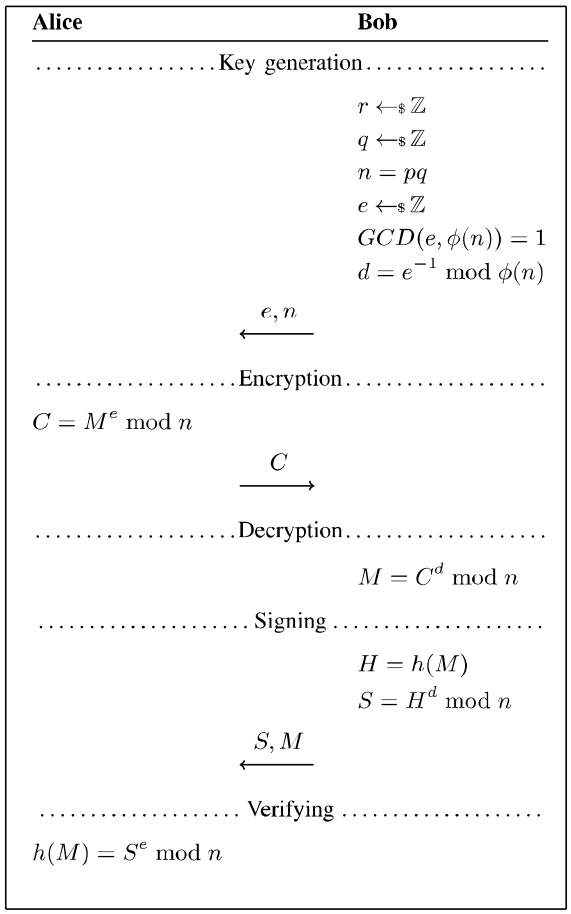

Fig. 7. The RSA cryptoscheme

$(R_A||R_B)$: $S_B = (h(R_A||R_B))^{d_B} \bmod n_B$. Then he sends the numbers R_B and S_B to Alice.

3) Alice verifies Bob's signature to the value $(R_A||R_B)$. If the signature S_B is false she terminates the protocol. If the signature S_B is valid, then she computes her signature S_A to the value $(R_A||R_B)$: $S_A = (h(R_A||R_B))^{d_A} \bmod n_A$. Then Alice selects a fake message M, computes the numbers $Z_A = R_B^{k_A} \bmod P$, $V = TZ_A \bmod n_B$, $C_1 = (M + V)^{e_B} \bmod n_B$, and $C_2 = V^{e_B} \bmod n_B$, and sends the ciphertext (C_1, C_2) and signature S_A to Bob.

4) Bob verifies Alice's signature to the value $(R_A||R_B)$. If the signature S_A is false he terminates the protocol. If the signature S_A is valid, then he computes the values $Z_B = R_A^{k_B} \bmod P$ and $V = C_2^{d_B} \bmod n_B$. Then he computes the value $T' = VZ_B^{-1} \bmod n_B$ that is equal to T, i.e. he obtains the secret message T sent by Alice.

Proof that computing the secret message is correct is as follows: $Z_B \equiv R_A^{k_B} \equiv 3^{k_A k_B} \bmod P$; $Z_A \equiv R_B^{k_A} \equiv 3^{k_B k_A} \bmod P \Rightarrow$ $\Rightarrow Z_B = Z_A \Rightarrow T' \equiv VZ_B^{-1} \equiv VZ_A^{-1} \equiv TZ_AZ_A^{-1} \equiv T \bmod n_B \Rightarrow T' = T$.

The described protocol includes hidden procedure of exchanging the single-use public keys R_A and R_B, which is masked as sending random values for performing the mutual

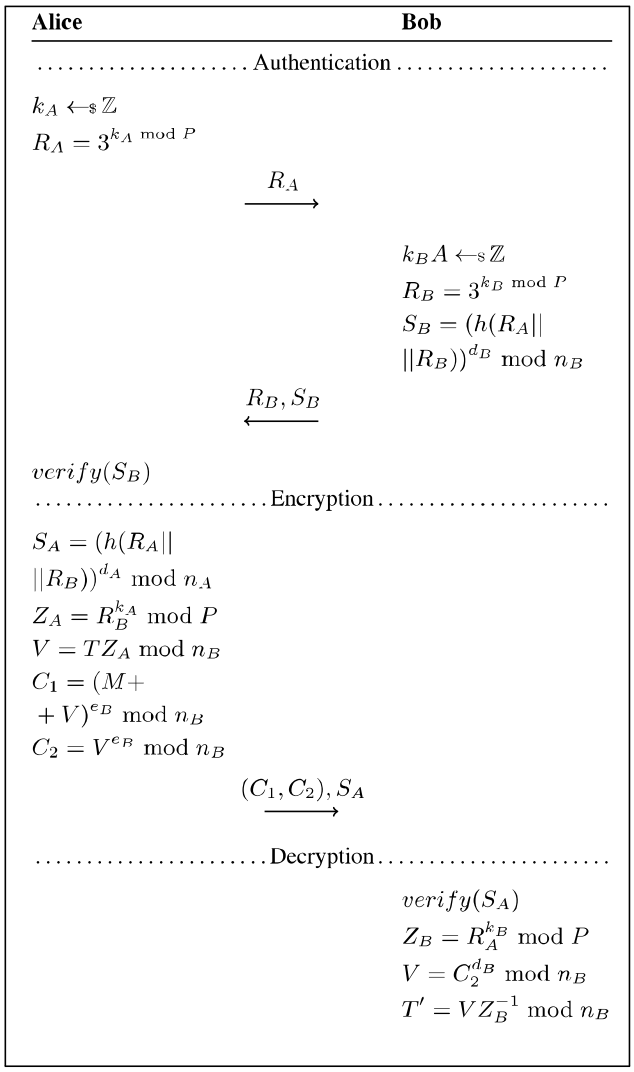

Fig. 8. Deniable public encryption protocol based on RSA

authentication of the users (see Fig 8). Then the public keys R_A and R_B are used to agree the single-use shared secret key $Z = Z_A = Z_B$. The last is used to encrypt the secret message T producing the ciphertext V masked as a random choice in the procedure of the probabilistic public encryption that is performed using the following algorithm.

The protocol can be easily introduced in practice, since it uses the RSA public-key infrastructure.

c) The probabilistic public encryption algorithm associated with the deniable encryption based on RSA: To encrypt a message M Alice performs the following steps:

1) Generate a random number $V < n_B$.

2) Using Bob's public key (n_B, e_B) encrypt the message M producing the ciphertext $C_1 = (M + V)^{e_B} \bmod n_B$.

3) Using Bob's public key encrypt the number V producing the ciphertext $C_2 = V^{e_B} \bmod n_B$.

4) Send the ciphertext (C_1, C_2) to Bob.

Rationality for using the probabilistic public encryption

consists in providing security in the case of encryption of short messages.

d) Dishonest decryption algorithm: When being coerced Bob decrypts the ciphertext (C_1, C_2) as follows:

1) Compute the value $V = C_2^{d_B} \bmod n_B$.

2) Compute the fake message $M = C_1^{d_B} - V \bmod n_B$.

In the case of the coercive attack on Alice, she opens the message M. To prove that the value V is not random the coercer should compute the number k_A or k_B, using the value R_A or R_B, respectively, however this is as difficult as computing discrete logarithm modulo P.

Thus, to distinguish the pseudo-random numbers R_A and R_B from the random numbers the coercer should compute the discrete logarithm modulo P. The last means the bi-deniability of the proposed protocol is based on the computational difficulty of finding discrete logarithms.

e) Comparison of two protocols: The protocol described in section III (see paragraph d) has exponential security against coercive attacks due to using the discrete logarithm problem on elliptic curve for implementing the hidden public key agreement procedure used for creating the single-use shared key z'. The deniable public encryption protocol based on the RSA cryptoscheme has sub-exponential security, since computing discrete logarithm modulo P has sub-exponential complexity.

In the case of the first protocol the users are not needed to generate their public keys having some additional properties. In the case of the second protocol the users are needed to generate their public keys having properties specified by the RSA cryptoscheme and additional property that consists in generating the modulus n such that the value $P = 2n + 1$ is prime and the order of number 3 modulo P is equal to $2n$ or n. Nevertheless one can attribute each of these two protocols to those that are based on the standard PKI for the cases of the cryptosystems [3] and RSA, respectively.

Like ElGamal public encryption algorithm [5], the first of the considered deniable encryption protocols can be attributed to the hybrid cryptoschemes, since the encryption of the messages is performed using the shared single-use secret keys and the last are distributed using the public keys.

VI. Conclusion

It has been used a new criterion of the computational indistinguishability between the probabilistic and deniable encryption for designing the deniable encryption protocols. Two bi-deniable public encryption protocols have been proposed that are against both the passive coercive attack and the active one. Security against active coercive adversary is provided due to including the entity authentication stage in the protocol. The bi-deniability is provided due to using the random values in the form of the single-use public keys, while performing the mutual authentication of the sender and receiver of the message. This is also a novel item in the design of such type cryptoschemes.

The proposed protocols for deniable encryption represent interest for practical application due to the following its merits:

1) using the standard PKI;
2) bi-deniability of the encryption;
3) sufficiently high performance;
4) the size of ciphertext is comparatively low (only about 1.5 times larger in comparison with the ElGamal public-key encryption).

References

[1] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption", *in Adv. in Cryptol.–CRYPTO'97*, 1997,volume 1294, pp. 90-104.

[2] Federal Inf. Process. Stds. (NIST FIPS) - 186. Digital Signature Standard (DSS), 2013.

[3] GOST R 34.10-2012. Russian Federation Standard. Information Technology. Cryptographic Data Security. Produce and Check Procedures of Electronic Digital Signature, 2012 (in Russian).

[4] GOST R 34.11-2012. Russian Federation Standard. Information Technology. Cryptographic Data Security. Hash-Functions., 2012 (in Russian).

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Inform. Theory*, vol. IT-31, 1985, pp. 469-472.

[6] M.H. Ibrahim, "A method for obtaining deniable public-key encryption", *Int. J. of Netw. Secur.*, vol.1, 2009, pp. 1-9.

[7] M.H. Ibrahim, "Receiver-deniable public-key encryption", *Int. J. of Setw. Secur.*, vol. 2, 2009, pp. 159-165.

[8] Y. Ishai, E. Kushilevitz, R. Ostrovsky, M. Prabhakaran, A. Sahai, "Efficient non-interactive secure computation", *Adv. in Cryptol.–EUROCRYPT 2011*, volume 6632, 2011, pp. 406-425.

[9] ISO/IEC 10118-3:2004. Information Technology - Security Techniques - Hash-Functions – Part 3: Dedicated Hash-Functions, 2004.

[10] Federal Inf. Process. Stds. (NIST FIPS) - 202. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, 2015.

[11] ISO/IEC 15946-1:2008. Information Technology - Security Techniques - Cryptographic Techniques Based on Elliptic Curves – Part 1: General, 2008.

[12] M. Klonowski, P. Kubiak, and M. Kutylowski, "Practical deniable encryption", *in SOFSEM 2008: Theory and Pract. of Comput. Sci.*, 2008, pp. 599-609.

[13] N.Koblitz, "Elliptic curve cryptosystems", *Math. of Comput. Adv.*, 1987, vol. 48, pp. 203-209.

[14] 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. of Netw.*, 2009, vol. 5, pp. 370-377.

[15] B. Meng, J. Wang, "A receiver deniable encryption scheme", *in Int. Symp. on Inf. Proc. (ISIP09)*, 2009, pp. 254-257.

[16] B. Meng, J. Wang, "An efficient receiver deniable encryption scheme and its applications", *J. of Netw.*, 2010, vol. 6, pp. 683-690.

[17] V. Miller, "Use of elliptic curves in cryptography", *in Adv. in Cryptol.: Proc. of Crypto'85. Lect. Notes in Comput. Sci.*, 1986, vol. 218, pp. 417-426.

[18] N. A. Moldovyan, "An approach to shorten digital signature length", *Comp. Sci. J. of Moldova*, 2006, vol. 14, no. 3(42), pp. 390-396.

[19] A.A. Moldovyan, N.A. Moldovyan, V.A. Shcherbacov, "Short signatures from difficulty of the factoring problem", *Buletinul Academiei de Stiinte a Republicii Moldova. Matematica*, 2013, vol. 72-73, pp. 27-36.

[20] A. ONeill, C. Peikert, B. Waters, "Bi-deniable public-key encryption", *in Adv. in Cryptol.–CRYPTO 2011*, 2011, vol. 6841, pp. 525-542.

[21] R.L. Rivest, A. Shamir, L.M. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Comm. of the ACM*, 1978, vol. 21, pp. 120-126.