Hybrid deniable and short-key encryption protocols based on the authentication procedure

1st Tan Sy Ho
Institute of Cryptographic Science and Technology
Ha Noi, Viet Nam
hstan@vgisc.com

2nd Moldovyan Alexander Andreevich
St.Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
St. Petersburg, Russian
maa1305@yandex.ru

3rd Moldovyan Nikolay Andreevich
St.Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
St. Petersburg, Russian
nmold@mail.ru

4th Canh Ngoc Hoang
Thuongmai University
Ha Noi, Viet Nam

5th Minh Hieu Nguyen
Institute of Cryptographic Science and Technology
Ha Noi, Viet Nam
hieuminhmta@ymail.com

6th Lich Van Luu
Academy of Cryptography Techniques
Ha Noi, Viet Nam

7th Tien Van Nguyen
Le Quy Don Technical University
Ha Noi, Viet Nam

8th Manh Cong Tran
Le Quy Don Technical University
Ha Noi, Viet Nam
manhtc@gmail.com

***Abstract*—To ensure resistance to attacks with coercion to disclose a secret key by an active adversary, the protocol of deniable encryption includes a procedure for mutual authentication of the sender and receiver of a message with their long-term public keys, which is combined with a hidden exchange of single-use public keys used to generate a single-use shared key, on which the secret message is encrypted. Long-term public keys are used to generate a shared secret key, on which a fake message is encrypted. The produced intermediate ciphertexts are converted into a single ciphertext, which is computationally indistinguishable from the ciphertext obtained by probabilistic encryption of a fake message. This approach allows us to build an encryption protocol that is resistant to bi-sided attacks with coercion, since the disclosure of users' private keys gives access only to the fake message and cannot be used to prove the possibility of alternative decryption of the ciphertext transmitted over a public communication channel. The authentication of the single-use public keys is also used to implement protocols for secure encryption using short shared keys.**

Keywords— information protection, cryptography, encryption, deniable encryption, probabilistic encryption, public keys.

I. INTRODUCTION

The interest in deniable encryption (DE) algorithms and protocols is associated with the prospects of their use in secure distributed computing, electronic voting systems [1], [2] and as a special information protection mechanism in complex computer security tools based on cheating traps [3].

A characteristic feature of DE protocols is the possibility of alternative decryption of ciphertexts produced during the course of protocols. Alternative decryption underlies the protocol resistance to potential coercive attacks, in which the adversary is supposed to receive the original message and the encryption key (in addition to the intercepted ciphertext). It is usually assumed that a coercive attack is carried out by a passive adversary [1], who has access to the communication channel used by protocol participants. The adversary intercepts messages that are transmitted during the execution of the protocol, and after a communication session, he forces the sender and/or receiver to open the secret key and the original message. Due to the possibility of ambiguous decryption of ciphertexts transmitted during the protocol, the attacker is provided with some fake message and a fake key that are associated with intercepted ciphertexts by some probabilistic encryption algorithm. Since the ciphertexts could potentially be obtained during the probabilistic encryption of the disclosed message with the public key, the attacker's requirements are considered fulfilled and he does not have reasonable evidence of incompleteness of the disclosed data, i.e. protocol participants can plausibly deny the fact of transmitting a secret message.

Deniable encryption protocols can be built on the basis of both symmetric and asymmetric cryptographic schemes. In the first case, the receiver and the sender of the message share two encryption keys (fake and secret) and in the case of a coercive attack, they reveal only the fake key. In the second case, the sender of the message encrypts with the receiver's public key, i.e. he must disclose only the original message, and the receiver must disclose his private key associated with the public key used to encrypt the message. Approaches to the construction of practically significant DE protocols with a shared secret key are presented in [4]–[6], and DE protocols with a public key in [7]–[9], [15]–[18].

To protect against coerced attacks by the active adversary, in [7] it is proposed to include in the DE protocol a procedure for mutual authentication of protocol participants using public keys, during which a hidden exchange of single-use public keys is carried out. The latter is used to generate a single-use shared key, by which a secret message is converted into ciphertext, disguised as random data used for probabilistic encryption of a fake message using the receiver's public key.

In this paper, we consider the construction of a hybrid protocol of deniable encryption, in which public keys are used during the stage of mutual authentication of the sender and receiver of the message and additionally applied to perform the public key-agreement procedure for generating a shared fake key used to encrypt a fake message. To encrypt a secret message, a single-use shared secret key is used, which is formed using random values exchanged by the protocol participants during the course of mutual authentication procedure.

The following requirements are accepted as the main design criteria:

- high performance;

- resistance to attack with simultaneous coercion of the sender and receiver of the message by the active adversary;

Computational indistinguishability of the ciphertext produced by the protocol, which is to be developed, from the ciphertext produced by protocol of probabilistic hybrid encryption.

The technique of the authentication of the single-use public keys is also applied to design the protocols for secure encryption using small shared keys.

II. Hybrid Deniable Encryption Protocol

As a method for protecting the secret message in the case of active attacks with coercion, the protocol of mutual authentication of the sender and receiver of the message with their public keys has been included in the protocol. This allows you to detect both the violator impersonating the sender of the message, and the violator impersonating the receiver. In the developed protocol, the Diffie-Hellman key exchange protocol [10] and the Schnorr digital signature algorithm [11] are used.

A. Basic cryptosystems

In the Diffie-Hellman protocol, the numbers p and α are specified as general parameters. The number p is a prime of sufficiently large size and α is a primitive element modulo p. The value of p is chosen so that finding a discrete logarithm modulo p is a computationally intractable task using the best know algorithm. To use this protocol, each user generates private key in the form of a uniformly random number x ($1 < x < p - 1$) and computes his public key y accordingly to the formula

$$y = \alpha^x \bmod p.$$

Users place their public keys in some certification authority, which confirms the authenticity of the public keys by sending out digital public key certificates to everyone who wants it. The proposed DE protocol assumes that each participant in this protocol knows the genuine public key of the other party.

The procedure of public key-agreement in framework of which a shared secret key is computed by two remote users is as follows. The sender (user A), using the receiver's public key (user B) y_B and his private key x_A, computes the shared secret key according to the formula $Z = y_B^{x_A} \bmod p$. The receiver, using the sender's public key y_A, and his private key x_B computes the shared secret key according to the formula $Z = y_A^{x_B} \bmod p$. The shared key Z will be to be used to encrypt the fake message, therefore, it can be called shared fake key.

During the DE protocol, the described procedure of public key-agreement is also used to generate a single-use Q_{AB} shared key. To do this, the sender generates his single-use private key in the form of a uniformly random number k_A ($1 < k_A < p - 1$) and computes his single-use public key R_A accordingly to the formula

$$R_A = \alpha^{k_A} \bmod p.$$

Similarly, the receiver generates his single-use private key in the form of a uniformly random number k_B ($1 < k_B < p - 1$) and computes his single-use public key R_B according to the formula

$$R_B = \alpha^{k_B} \bmod p.$$

Since the number α is a primitive root modulo p, the generated numbers R_A and R_B take on every value from the set $\{1, 2, ..., p - 1\}$ whit the same probability, i. e., the numbers R_A and R_B are uniformly random values (see Statement 1).

Statement 1. *Suppose x is a uniform random variable taking on the values in the area of integers $\{1, 2, \ldots, p - 1\}$. Then the formula $y = \alpha^x \bmod p$ defines the uniform random variable y taking on the values in the area of integers $\{1, 2, \ldots, p - 1\}$.*

Proof. The value of the function $y = \alpha^x \bmod p$ is random due to random argument. Since α is a primitive element, every value computed as $y = \alpha^i \bmod p$ for $i = 1, 2, \ldots, p - 1$ is unique and lie in the set $\{1, 2, \ldots, p - 1\}$. Probability to get some fixed value y_0 from the last set is equal to probability to select the single integer x_0 such that $y_0 = \alpha^{x_0} \bmod p$. Thus, we have $\text{Prob}(y = y_0) = \text{Prob}(x = x_0) = (p - 1)^{-1}$, i.e. the value $y = \alpha^x \bmod p$ is uniform random one.

The values R_A and R_B are sent via a communication channel as random requests of the handshake protocol performed during the mutual authentication procedure. The values R_A and R_B are also used to compute a single-use shared secret key Q:

$$Q = R_B^{k_A} \bmod p = R_A^{k_B} \bmod p.$$

The Schnorr digital signature algorithm is based on the computational complexity of the discrete logarithm problem modulo prime p having a sufficiently large size (for example, 2048 bits) such that $p - 1$ is divisible by another large prime r (for example, 192 – 256 bits) [12]. A number g is also specified, the order of which modulo p is r. The public key is computed accordingly to the formula $y = g^x \bmod p$, where x is the private (secret) key of the owner of the public key y. In addition to performing digital generation and authentication procedures, public keys of this kind can also be used to implement the protocol for public key-agreement procedure for generating a shared secret key using the Diffie-Hellman key exchange protocol with replacing the primitive element α by the number g.

The calculation of the signature for the message M includes the following steps:

1. A random number k is generated, $1 < k < r - 1$.
2. The value of the single-use public key $Y = g^k \bmod p$ is computed.
3. The number Y is concatenation to the message M and the hash function H of the value of $M \| Y$ is computed:

$$E = H(M \| Y).$$

4. The value of S is computed as follows:

$$S = k + xE \bmod r.$$

The signature is a pair of numbers (E, S).

The signature verification procedure is as follows:

1. The value of Y' is computed accordingly to the formula $Y' = y^{-E} g^S \bmod p$.
2. The number Y' is concatenation to the message M and the value of the hash function $E' = H(M \| Y')$ is computed.

3. A comparison is made between the values of E and E'. If $E = E'$, then the signature is accepted. Otherwise, the signature is rejected.

B. Protocol description

The developed hybrid DE protocol uses a prime modulus p that satisfies the requirements of the Schnorr digital signature algorithm and the parameters g and α associated with the prime p. It is assumed that each party of this protocol generates and registers a public key y in the certifying center (certification authority) in advance. The public key y is computed using the formula $y = g^x \bmod p$ and used to perform both digital signature verification procedures and public key-agreement procedure for generating the shared fake key Z. The single-use public keys, used in the protocol, are computed by users using the number α, which is a primitive element modulo p, which ensures the uniformity of random values used when for performing the procedure of mutual authentication of the sender and receiver of the message.

The DE protocol described below can be attributed to the hybrid type of crypto schemes, since it uses public-key cryptoschemes to generate the single-use shared keys U and K and, using the latter, a simultaneous cryptographic transformation of a fake and secret messages is performed using a symmetric encryption algorithm. It is assumed that the parties of the secret communication session, users A (sender) and B (receiver), are the owners of the public keys y_A and y_B registered in the certification authority, respectively. Users take the values $y_A = g^{x_A} \bmod p$ and $y_B = g^{x_B} \bmod p$ from the public key directory or from each other's digital certificates. The sender transmits a secret message T via an insecure (public) communication channel in the following way.

1. The sender generates a random number k_A satisfying the condition $1 < k_A < p - 1$, and computes the value $R_A = \alpha^{k_A} \bmod p$ that is single-use public key. Then, using his private key x_A, in accordance with the signature generation procedure in the Schnorr scheme, generates his $\mathrm{Sign}_A(R_A)$ signature to the value R_A and sends the values R_A and $\mathrm{Sign}_A(R_A)$ to the receiver.

2. The receiver, using the public key y_A, authenticates the $\mathrm{Sign}_A(R_A)$ signature. If the signature is genuine, then it generates a random number k_B satisfying the condition $1 < k_B < p - 1$, and computes the value $R_B = \alpha^{k_B} \bmod p$ that is single-use public key. Then, using his private key x_B, forms his signature $\mathrm{Sign}_B(R_A)$ to the value R_A and his signature $\mathrm{Sign}_B(R_B)$ to the value R_B and sends the values R_B, $\mathrm{Sign}_B(R_A)$ and $\mathrm{Sign}_B(R_B)$ to the sender.

3. The sender, using the public key y_B, verifies the $\mathrm{Sign}_B(R_A)$ signature to the random value that he sent to user B, and the $\mathrm{Sign}_B(R_B)$ signature to the value R_B. If both signatures are authentic, then it encrypts and transmits the secret message T ($T < p$) to the receiver, performing the following steps below. (Otherwise, it interrupts the communication session.)

   3.1. Computes the value
   $Q_{AB} = R_B^{k_A} \bmod p = \alpha^{k_B k_A} \bmod p$ that is a single-use shared secret key.

   3.2. Computes the value
   $Z_{AB} = y_B^{x_A} \bmod p = g^{x_B x_A} \bmod p$ that is a shared secret key, the value of which depends on each bit of the public keys y_A and y_B.

   3.3. Computes the session key $K = Z_{AB} R_A R_B \bmod p$.

   3.4. Generates a fake message $M < p$.

   3.5. Generates the ciphertext C as a solution $C = (C_1, C_2)$ of the following system of equations in the finite ground field GF(p) with the unknowns C_1 and C_2:

$$\begin{cases} Q_{AB} C_1 + Q_{AB}^2 C_2 = T \bmod p \\ K C_1 + K^2 C_2 = M \bmod p \end{cases}.$$

The probability that this system will not have a solution is p^{-1}, i.e. this case can be neglected.

   3.6. Computes its signature $\mathrm{Sign}_A(C)$ to the ciphertext $C = (C_1, C_2)$.

   3.7. Sends the ciphertext $C = (C_1, C_2)$ and the signature $\mathrm{Sign}_A(C)$ to the receiver.

4. The receiver verifies the signature $\mathrm{Sign}_A(C)$. If the signature is false, it ignores the C ciphertext and terminates the communication session. If the signature is genuine, then he decrypts the secret message T performing the following steps:

   4.1. Computes the value
   $Q_{BA} = R_A^{k_B} \bmod p = \alpha^{k_B k_A} \bmod p$ that is a single-use shared secret key, depending on random values of R_A and R_B.

   4.2. Recovers secret message T accordingly to the formula

$$T = (Q_{BA} C_1 + Q_{BA}^2 C_2) \bmod p.$$

   4.3. Computes the value

$$Z_{BA} = y_A^{x_B} \bmod p = g^{x_A x_B} \bmod p$$

that is a shared fake key, depending on the registered public keys y_A and y_B.

   4.4. Computes the session key $K = Z_{BA} R_A R_B \bmod p$.

   4.5. Recovers a fake message M:

$$M = (K C_1 + K^2 C_2) \bmod p.$$

In the event of a coerced attack, both sides of the secret communication session declare the mutual authentication procedure, after which the probabilistic encryption of message M was performed using the protocol described in the next section.

C. Associated Probability Encryption Protocol

The ciphertext sent to the receiver is computationally indistinguishable from the ciphertext that is potentially generated by the following probabilistic encryption protocol (which is associated with the DE protocol) used to securely transmit the message M via an insecure (public) channel:

1. The sender and receiver exchange uniform random values of R_A ($1 < R_A < p$) and R_B ($1 < R_B < p$).

2. The sender, using the receiver's public key y_B, computes the shared secret key according to the formula $Z_{AB} = y_B^{x_A} \bmod p$ and the session key $K = Z_{AB} R_A R_B \bmod p$. Then he encrypts the message M by following these steps:

2.1. Generates two random values ρ_1 ($1 < \rho_1 < p$) and ρ_2 ($1 < \rho_2 < p$), such that the inequality $\rho_1 K^2 - \rho_2 K \neq 0 \bmod p$ holds.

2.2. Generates the ciphertext C as a solution $C = (C_1, C_2)$ of the following system of equations in the finite ground field GF(p) with respect to the unknown C_1 and C_2:

$$\begin{cases} \rho_1 C_1 + \rho_2 C_2 = 1 \bmod p \\ K C_1 + K^2 C_2 = M \bmod p \end{cases}.$$

D. Coerced Attack Resistance Discussion

Undergoing a bi-sided coerced attack, users reveal their secret keys x_A and x_B and the fake message M. An attacker who previously intercepted the ciphertext $C = (C_1, C_2)$ and random values R_A and R_B transmitted over an open channel computes a session key:

$$Z_{AB} = y_B^{x_A} \bmod p;$$

$$K = Z_{AB} R_A R_B \bmod p.$$

Then, using the session key K, decrypts the cipher text C and recovers the message M:

$$M = (K C_1 + K^2 C_2) \bmod p.$$

The attacker is convinced that the value of M is opened correctly. Knowledge of the private keys x_A and x_B cannot be used by an attacker to compute the single-use shared secret key $Q = Q_{BA} = Q_{AB}$, because for this he needs to solve the computationally difficult problem of discrete logarithm modulo prime p having large size. It also cannot convince users of fraud by identifying the difference between random values R_A and R_B from uniform random values, since when choosing uniform random values k_A ($1 < k_A < p - 1$) and k_B ($1 < k_B < p - 1$), random values R_A ($1 < R_A < p$) and R_B ($1 < R_B < p$) are also uniform. Thus, the attacker cannot prove that, within the framework of the described protocol, users agreed on a single-use shared secret in accordance with the Diffie – Hellman key exchange protocol and cannot convincingly argue that the ciphertext contains not only the message M, but also some other meaningful message.

Active coercive attacks are prevented by the fact that the receiver must sign a random number R_A generated by the sender, and the latter must sign the ciphertext formed by him. Moreover, by agreement on the model of a coerced attack, it is assumed that the transmitted message and private keys are disclosed to the attacker after the ciphertext has been transmitted over the communication channel. Before the private keys are revealed to the attacker, he cannot compute the correct signature values to random requests when performing the procedure of mutual authentication of the sender and receiver, i.e. he cannot impose the execution of a false DE protocol, impersonating the true sender or receiver of a message. The fact that the ciphertext size exceeds the size of the recovered message M is argued by users that they use the probabilistic encryption procedure to improve the statistical properties of the ciphertext.

III. Secure encryption on small shared keys

In this section we propose another application of the idea of applying single-use public keys, namely, for implementing secure encryption using small shared keys. Suppose the sender and the receiver of a secret message share a short secret key k.

Secure encryption of some secret message M can be achieved using a shared secret of small size and performing public encryption on the single-use public keys, the shared secret key being used for authenticating the single-use public keys. Consider the protocols implementing this idea.

A. Secure communication protocols

Protocol 1.

1. The sender, using generator of the uniform random sequence of bits, forms his local single-use secret key $x_1 < p$ and computes his single-use public key $y_1 = \alpha^{x_1} \bmod p$. Then, using a secure symmetric-encryption algorithm E, encrypts the value y_1 on the key k and obtains the ciphertext $C_1 = E_k(y_1)$ that is sends to the receiver.
2. The receiver, using generator of the uniform random sequence of bits, forms his local single-use secret key $x_2 < p$ and computes his single-use public key $y_2 = \alpha^{x_2} \bmod p$. Then he encrypts the value y_2 on the key k and obtains the ciphertext $C_2 = E_k(y_2)$ that is sends to the sender.
3. The sender computes the common secret key $Z = y_2^{x_1} \bmod p$, encrypts the message M, obtaining the ciphertext $C_3 = E_Z(M)$. Then he sends the value C_3 to the receiver.
4. The receiver computes the common secret key $Z = y_1^{x_2} \bmod p$, decrypts the ciphertext C_3, opening the secret message $M = D_Z(C_3)$, where D_Z is the decryption.

Protocol 2.

1. The sender generates two random strong [8] primes r1 and q1, such that the number 3 divides none of the numbers r1 - 1 and q1 - 1. Then he computes the value n1 = r1q1 and sent n1 (sender's single-use public key) to the receiver.
2. The receiver generates two random strong primes r_2 and q_2, such that the number 3 divides none of the numbers $r_2 - 1$ and $q_2 - 1$. Then it generates his single-use public key $n_2 = r_2 q_2$, computes the value $S = n_2^3 \bmod n_1$ and ciphertext $C_1 = E_K(S)$ and sends C_1 to the sender.
3. The sender decrypts the ciphertext C_1, getting the value $S = D_K(C_1)$, where D_K is the decryption function, the inverse of the encryption function E_K. Then it computes the values $d = 3^{-1} \bmod (r_1 - 1)(q_1 - 1)$ and $n_2 = S^d \bmod n_1$. After that, the sender encrypts M using the single-use public key of the receiver, forming a $C_2 = M^3 \bmod n_2$, which is sends to the receiver.
4. The receiver computes the value $D = 3^{-1} \bmod (r_2 - 1)(q_2 - 1)$ and decrypts the ciphertext C_2, recovering the message $M = C_2^D \bmod n_2$.

The encryption procedure using the single-use public key used in this protocol corresponds to the encryption procedure in the RSA cryptosystem [13] when choosing the exponent of the public key equal to number 3. In order to have the possibility of unambiguous decryption, the sender and

receiver of the message use primes, from which the Euler function is not divided by 3. In protocol 2, the most time-consuming procedure is the generation of sufficiently large primes r_1, q_1, r_2 and q_2, therefore it is inferior in performance to protocol 1.

The computational complexity of the secure small-key encryption procedure is reduced by the following protocol, which also uses a single-use public key encryption in accordance with the El-Gamal encryption algorithm [14] using the prime number p and the primitive element $\alpha \bmod p$.

Protocol 3.

1. The receiver, using the generator of a uniform random sequence of bits, generates his single-use local secret key $x < p$ and computes a single-use public key $y = \alpha^x \bmod p$. Then he encrypts the value y on the key k and obtains the ciphertext $C = E_k(y)$. After that, the value of C is sent to the sender.
2. The sender computes the single-use public key of the receiver $y = D_k(C)$. Then it generates a random number $k < p$, computes the first $R = \alpha^k \bmod p$ and the second element of the ciphertext $S = My^k \bmod p$, where M is a secret message ($M < p$) and sends the ciphertext (R, S) to the receiver.

The receiver, using his single-use secret key x, decrypts the ciphertext as follows: $M = SR^{-x} \bmod p$.

B. Security Discussion

The security of the protocol means the high security of the cryptographic transformations used in its framework and the rather small probability that the active intruder can impersonate a legitimate participant in the protocol (in the considered protocols, as the sender or receiver of a secret message). The security of the protocols described earlier is determined by the security of the public-key cryptographic schemes used in them, the authentication of single-use public keys with the shared secret key k and the conversion using the latter in such a way that the potential attacker is forced to guess the value of k, since there is no criterion for recognizing the true value k, when the exhaustive search of the key value (due to the use of a shared key of small size, the exhaustive search is possible). Moreover, even if the value of k is chosen by the attacker correctly, he still can not read the secret messages sent between authorized participants in the protocol. However, if you know the shared secret key, the intruder can impersonate the sender or receiver and impose a false session in which it is possible to impose a false message or elicit a secret message. Consider each of the proposed three protocols separately and their features.

Security of protocol 1 is determined by the fact that to compute a single-use shared secret key Z from single-use public keys y_1 and y_2, it is required to solve the discrete logarithm modulo p problem, which is computationally impossible for the foreseeable time. Moreover, the situation for a potential cryptanalyst is significantly aggravated by the fact that the values y_1 and y_2 are transmitted in an encrypted form (encrypted on the key k). To compute the shared secret key k by exhaustive search when decrypting the ciphertext C_1 or C_2, it is necessary to recognize the case of the correct value y_1 or y_2, respectively. However, the statement 1 about the randomness and uniformity of value of the single-use public key does not leave a cryptanalyst of a computationally effective criterion for recognizing the true value of a shared key.

In Protocol 2, the shared secret key is also actually used not to directly encrypt the secret message, but to authenticate the public key of the message receiver and authenticate the sender of the message. Without knowledge of the shared secret key, the intruder cannot impersonate the receiver, nor can he impose a false message on the receiver. The shared secret key is used to encrypt the S value, which actually represents the ciphertext obtained by public encryption of the receiver's single-use public key with the sender's single-use public key. The value of S is computationally indistinguishable from an equiprobable random value, therefore, the potential attacker does not have a computationally effective criterion for recognizing the correct value of the secret key k when performing exhaustive search through the space of possible values of k. As a criterion, you can use the recovery of the value n_2, however, for this it is necessary to solve the problem of factoring the number n_1, which is computationally intractable. Without solving the problem of factoring n_1, the attacker has to guess at random a certain value as the correct k, which will give him the potential in the next communication sessions to impersonate the sender or receiver. However, in the current communication session, observing which, he managed to guess the value of k, he does not receive the practical ability to read the secret message, since it was encrypted using a single-use public key of the receiver. If the value k is incorrectly selected, the communication session imposed by the attacker will be interrupted. The attacker will be able to experience the new key value only in a new imposed communication session. Since the verification of the correctness of the current selected key value requires the imposition of a false communication session, the probability of guessing 2^{-32} or less is practically acceptable. Since the probability of guessing a random k-bit key is 2^{-k}, it is sufficient to use the k key of 32 bits or more (40 and 56 bits) in the considered protocols.

Protocol 3 is similar to protocol 1, because it also uses two single-use public keys and the formation of a single-use shared secret key. Indeed, the El-Gamal public encryption algorithm is actually a hybrid cryptosystem in which secret keys are distributed in accordance with the Diffie-Hellman key exchange protocol [10], and message encryption is performed by multiplying the message modulo p by a single shared secret key. The difference between protocols 1 and 3 mainly consists in the fact that in the second single-use public key of the sender is sent to the receiver in the clear. In view of the latter, the potential attacker in the case of protocol 3 directly has the initial parameters needed to solve the discrete logarithm problem, whereas in the case of protocol 1, an attempt to proceed with the solution of this problem requires determining the true value of the shared key k. Thus, in protocol 1, the attacker can gain access to a secret message without finding a shared secret key, solving only the discrete logarithm problem. However, the latter is practically intractable, if a prime modulus p has a sufficiently large size (> 2000 bits).

IV. Conclusion

A hybrid DE protocol has been developed based on the public key-agreement and digital signature schemes. The protocol has a relatively high performance and is resistant to coercive attacks carried out simultaneously on the sender and receiver from both the passive and the active adversary. The use of a mechanism for authenticating users and transmitted

messages within the framework of the protocol makes it possible to detect an active adversary and hiddenly perform additional public key-agreement procedure for generating a single-use shared secret key.

Development of a hybrid DE protocol similar to that proposed in this paper, but based on the use of the signature standards (for example, ECDSA) represent practical interest. Correspondingly, in the said version of the hybrid DE protocol the public key-agreement procedure for generation of the shared keys Z and Q is to be implemented using calculations on an elliptic curve. In this case, an increase in performance and possibility to use the existing public key infrastructure are achieved.

The technique of authenticating the single-use public keys also provides development of the protocols for secure encryption using short shared keys. Such protocols are of considerable interest for ensuring the protection of information transmitted over open channels, in conditions of limited key material.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All authors contributed equally to this paper, and were cooperatively involved in conceptualization, investigation, formal analysis and writing. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENT

This research was funded by the Ministry of Science and Technology (MOST) under grant KC.01.22/16-20/.

REFERENCES

[1] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable Encryption," *Advances in Cryptology, CRYPTO 1997 Proceedings*, vol. 1294, pp. 90–104, 1997.

[2] Bo Meng, "A Secure Internet Voting Protocol Based on Noninteractive Deniable Authentication Protocol and Proof Protocol that Two Ciphertexts are Encryption of the Same Plaintext," *Journal of Networks*, vol. 4, no. 5, pp. 370-377, 2009.

[3] A. A. Moldovyan, N. A. Moldovyan, and V. A. Shcherbakov, "Bi-Deniable Public-Key Encryption Protocol Secure Against Active Coercive Adversary," *Buletinul Academiei de Stiinte a Republicii Moldova. Matematica*., no. 3 (76), pp. 23-29, 2014.

[4] N. A. Moldovyan, Al-Majmar, N. D. Tam, N. N. Hai, and N. H. Minh, "Deniability of Symmetric Encryption Based on Computational Indistinguishability from Probabilistic Ciphering," in *Information Systems Design and Intelligent Applications: Proceedings of the Fourth International Conference INDIA 2017 and Advances in Intelligent Systems and Computing*. Springer Nature Singapore Pte Ltd., 2018, vol. 672, pp. 209-218, 2018.

[5] N. A. Moldovyan, A. A. Moldovyan, N. D. Tam, N. N. Hai, T. C. Manh, and N. H. Minh, "Pseudo-probabilistic block ciphers and their randomization," *J. Ambient Intelligence and Humanized Computing*., vol. 10, no. 5., pp. 1977-1984, 2019.

[6] N. A. Moldovyan, A. A. Moldovyan, D. N. Moldovyan, and V. A. Shcherbacov, "Stream Deniable-Encryption Algorithms," *Computer Science Journal of Moldova*, vol. 24, no. 1(70), pp. 68-82, 2016.

[7] A. A. Moldovyan and N. A. Moldovyan, "Practical Method for Bi-Deniable Public-Key Encryption," *Quasigroups and related systems*., vol. 22, pp. 277-282, 2014.

[8] N. A. Moldovyan, A. A. Moldovyan, and V. A. Shcherbacov, "Generating Cubic Equations as a Method for Public Encryption," *Buletinul Academiei de Stiinte a Republicii Moldova*., Matematica, no. 3 (79), pp. 60-71. 2015.

[9] N. A. Moldovyan, A. A. Moldovyan, and V. A. Shcherbacov, "Provably sender-deniable encryption scheme," *Computer Science Journal of Moldova*, vol. 23. no. 1(67), pp. 62-71, 2015.

[10] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*. vol. IT-22. pp. 644–654, 1976.

[11] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4. pp. 161-174, 1991.

[12] J. Gordon, "Strong primes are easy to find. Advances in cryptology," *EUROCRYPT'84*. Springer-Verlag LNCS. 1985, vol. 209, pp. 216–223, 1985.

[13] J. Pieprzyk, Th. Hardjono, and J. Seberry, *Fundumentals of Computer Security*, Springer-verlag. Berlin, 2003, 677 p.

[14] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp.469–472, 1985.

[15] N. A. Moldovyan, A. N. Berezin, A. A. Kornienko, and A. A. Moldovyan, "Bi-deniable Public-Encryption Protocols Based on Standard PKI," in *18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, pp. 212–219, 2016.

[16] N. Moldovyan, A. Berezin, A. Kornienko, and A. Moldovyan, "Deniable encryption protocols based on probabilistic public-key encryption," in *Open Innovations Association (FRUCT) 2017 20th Conference of*, pp. 284-289, 2017.

[17] M. T. Barakat, "A New Sender-Side Public-Key Deniable Encryption Scheme with Fast Decryption," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 9, 3231–3249, 2014.

[18] D. Dachman-Soled, "On minimal assumptions for sender-deniable public key encryption," in *Public-Key Cryptography PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography*. Lecture Notes in Computer Science. SpringerVerlag. Berlin, Heidelberg, New York. vol. 8383, 574–591, 2014.