Abstract—This paper proposes a new variant of group digital signature based on masking user public keys, which reduces significantly the signature length. The proposed scheme does not include secret sharing and knowledge proving procedure. Therefore, this scheme allows flexible modification of group structure including change of group manager. Comparing to the known group signature protocols based on similar approach, the security of proposed protocol does not depend on the development of breakthrough factorization methods.

Keywords—Cryptographic protocol, public key, digital signature, group signature, collective signature, discrete logarithm problem, factorization problem, one-way function.

I. INTRODUCTION

Digital signature protocols are widely used in the information technologies to process electronic legal messages and documents. To solve a variety of different practical tasks, different types of signature are proposed: usual (individual) signature [1]; aggregate signature [2]; blind signature [3, 4]; group signature [5, 6]; collective signature [7]. The collective signature refers to a signature generated by a declared set of signers, while group signature refers to a signature formed on behalf of group of signers headed by a person called group manager or dealer.

The collective signature of an electronic document means that each of the designated users signs the document. Thus, this signature can be considered as a representative of \( m \) individual signatures. Each of the signers is responsible for the content of document to be signed. To verify a given collective signature, the public keys of all users who generated the signature need to be used. Furthermore, the collective signature protocols and individual signature protocols can use the same public key infrastructure. The implementation of the protocol allows changing a set of signers arbitrarily. The last two properties of the protocol represent important practical characteristics of the collective signature.

The group signature of an electronic message is generated by a group of signers, and one of them is a group manager. To verify the group signature, group public key need to be used, and he/she can not reveal which particular group member signed the document. The group signature has the following important properties. Firstly, only group member can sign a document. Secondly, group manager, who has both document and valid group signature can reveal the group members signed the document. Finally, non-group members could not reveal the original signers, who generate the group signature. The group manager is a trusted party of the group signature protocol. He creates the secret parameters used by the signers to generate signature.

Recently a new design of the approvable group signature was proposed in [8]. This design provided a possibility to keep individual private keys of group members in secret. In order to obtain this property, the group signature is generated in two steps. In the first step, the “pre-signature” of group is provided. In the second step, group manager computes group signature using a given approval based on “pre-signature”. The “pre-signature” can be generated by any group member or subset of group members. Since only group manager can reveal the original signers of the document, the presented group signature protocol requires no distribution of the secret keys and uses the standard public key infrastructure. Moreover, the set of signers involved in the group can be arbitrarily changed by the group manager, whose public key is used as group public key. As the result, the group signature protocol, which has properties mentioned above, provides a procedure of document processing very close to everyday practices such as paper document preparation, signing and approval [5].

The implementation of the group signature protocol is based on two hard problems-factorization problem and discrete logarithm problem [8]. However, the development of the breakthrough methods of one of these problems leads to the protocol compromise.
Apart from this, the signature length is comparatively large and equals to 3012 bits in the case of 128-bit security (or equals to 1344 bits in the case of 80-bit security). In this paper, we propose a new variant of the group protocol, which can reduce significantly the signature length - up to 480 bits in case of 80-bit security. The design of the proposed scheme is based only on the computational difficulty of the elliptic curve discrete logarithm problem, and eliminates the dependency of the protocol security on factorization problem.

The rest of the paper is organized as follows. In the section 2, the brief description of the group protocol described in [8] is given. In the section 3, we present the implementation of the protocol based on elliptic curve over finite field. A new approach for masking public keys of the signers based on secure one-way hash-function is provided in section 4. Finally, the implementation results and conclusion are given in section 5.

II. GROUP SIGNATURE PROTOCOL

In the protocol described in [8], the following parameters are used: 1) sufficiently large prime \( p \) (with length more than 1024 bits), such that number \( p - 1 \) contains large prime divisor \( q \) (with length \( \geq 160 \) bits); 2) number \( \alpha \) which order is equal to \( q \) modulo \( p \). Each group member generates his private key \( k_j \) and computes his public key \( P_j = \alpha^{k_j} \mod p \), where \( j = 1, 2, 3, \ldots, g \); and \( g \) – number of group members. The public key \( L \) of the group manager is a public key of the group and is calculated as follows \( L = \alpha^z \mod p \), where \( z \) is a private key of the group manager. The public key \( L \) is used to verify group signature. The group manager has also internal public key represented by a pair of numbers \( (e, n) \) which are generated as in RSA cryptosystem [9]: \( n \) is a composite number that is hard to factor, \( e = d^{-1} \mod \phi(n) \), \( d \) is additional private key of the group manager; \( \phi(n) \) is the value of Euler phi-function for integer \( n \).

The group signature protocol is described as follows. Let \( m \) group members have public keys \( y_i = \alpha^{x_i} \mod p \) and corresponding private keys \( x_i, i = 1, 2, \ldots, m \), to sign document \( M \). They report about their intention to a group manager and send him document \( M \). Then a signing procedure is initialized. To generate group signature, the following procedure is implemented.

The group manager computes hash value from document \( H = F_H(M) \), where \( F_H \) is some specified hash functions, calculates randomizing exponents \( \lambda_i = (H + P_i) \mod n \) and sends value \( \lambda_i \) to a corresponding \( i \)-th group member for \( i = 1, 2, \ldots, m \). Then the group manager computes the first element of the group signature

\[
U = \prod_{i=1}^{m} P_i^{\lambda_i} \mod p .
\]  

Each \( i \)-th group member \( (i = 1, 2, \ldots, m) \) generates a random number \( t_i < q \), computes the value \( R_i = \alpha^{l_i} \mod p \) and sends \( R_i \) to the group manager.

The group manager generates the random number \( T < q \) and computes values such as

\[
R' = \alpha^T \mod p ,
\]

\[
R = R' (R_1 R_2 \ldots R_m) \mod p = \alpha^{2^{T} + \lambda_1 + \lambda_2 + \ldots} \mod p
\]

\[
E = F_H(M||R||U).
\]

where \( E \) is the second element of the group signature. He sends value \( R \) to the group members who have initiated the protocol.

Each \( i \)-th signer \( (i = 1, 2, \ldots, m) \) computes his signature share \( S_i = t_i + k_i \lambda_i E \mod q \) and sends it to the dealer.

The group manager verifies the correctness of each share \( S_i \) by checking equality \( R_i = P_i^{\lambda_i E} \alpha^{S_i} \mod p \). If all signature shares \( S_i \) satisfy verification procedure, then he computes his share \( S' = T + E \mod q \) and the third element of the group signature

\[
S = S' + S_2 + \ldots + S_m \mod q.
\]

The tuple \((U, E, S)\) generated by the above procedure represents group signature of the document \( M \).

The verification procedure includes the following steps. The verifier computes the hash-function value from the document \( M: H = F_H(M) \).

Using the group public key \( L \) and signature \((U, E, S)\) he computes value

\[
\hat{R} = (UL)^{E} \alpha^{S} \mod p .
\]

He computes value \( \hat{E} = F_H(M \parallel \hat{R} \parallel U) \) and compares the values \( \hat{E} \) and \( E \). If \( \hat{E} = E \), then the verifier concludes that the group signature is valid. Otherwise, he rejects the signature.
The first element of group signature $U$ contains information about the group members who signed the document $M$. This information is available only to the group manager. In case of necessity, he can reveal this information by implementing the following steps. Using his internal private key $d$ the group manager computes the values

$$\lambda_j = (H + P_j)^d \mod n, j = 1, 2, 3, \ldots, g. \quad (5)$$

For each group member and then multiplies all possible combinations of the values $P_j^3 \mod p$, until he gets value $U$. The combination that gives the value $U$ identifies the group members who produced group signature $(U, E, \Sigma)$. The exponentiation of the value $H + P_j$ by secret degree $d$ means that the group manager signed these values. The last item is an important item, which provides that only group manager have possibility to open the group signature and identify original signers of the document.

An adversary can reveal group members only if he can factorize number $n$. To forge a group signature it is required to solve the discrete logarithm problem. In the protocol described above, two hard problems are used for different purposes. Correcting function of the protocol computational difficulty of the both problems is significant. It is noted that the protocol is based on computing of two hard problems.

### III. Implementation of Group Signature Protocol Using Elliptic Curves

Currently, cryptographic protocols based on elliptic curves (EC) over finite field have been applied. The EC over finite field $GF(p^3)$ represents a set of pairs of elements $(x, y)$ belonging to finite field $GF(p^3)$, where $p$ is characteristic of the field $GF$, $s$ is degree of $GF$ over its prime subfield. The addition operation ($+$), commutative and associative are defined over the EC points. The sum of the points $A = (x_A, y_A)$ and $B = (x_B, y_B)$ represents the point $C = (x_C, y_C)$ which are computed by using quite simple formulas including values $x_A, y_A, x_B, y_B \in GF(p^3)$.

In the proposed protocol, we use the EC, which order contains a sufficiently large prime divisor $q$ (more than 256 bits) and a point $G$ having order equal to $q$.

Each member in the group of signers generates their private secret key as a random number $k$ and public key computed as the point $P = KG$. The group manager calculates his public key as the point $L = zG$, where $z$ is his private secret key. The internal public key of the group manager is calculated as described in the previous section. Then the group signature protocol is described as follows.

Let $m$ group members report about their intention to sign the document $M$ to the group manager and send it to him. To generate group signature, the following procedure is implemented.

The group manager computes hash value from document $H = F_H(M)$, where $F_H$ is some specified one-way hash-functions, calculates randomizing exponents $\lambda_i = (HP_i)^d \mod n$ and sends each value $\lambda_i$ to a corresponding $i$-th group member. Then the group manager computes EC point

$$U = \lambda_1P_1 + \lambda_2P_2 + \ldots + \lambda_mP_m. \quad (6)$$

which serves as the first element of the group signature.

Each $i$-th group member ($i = 1, 2, \ldots, m$) generates a random number $t_i < q$, computes the value $R_i = t_iG$ and sends $R_i$ to the group manager.

The group manager generates the random number $t' < q$, computes EC points $R' = t'G$, $R = R' + R_1 + R_2 + \ldots + R_m$ and the second element of the group signature $e = F_H(M || x_R || x_U)$, where $x_R$ and $x_U$ are $x$-coordinates of EC points $R$ and $U$, respectively. He sends the value $e$ to the group members who initiated the protocol.

Each $i$-th signer ($i = 1, 2, \ldots, m$) computes their signature share $s_i = t_i + k_i\lambda_i e \mod q$ and sends it to the group manager.

The group manager verifies the correctness of each $s_i$ by checking $R_i = s_iG - \lambda_i eP_in, i = 1, 2, \ldots, m$. If all signature shares $s_i$ satisfy verification procedure, then he computes his share $s' = t' + ze \mod q$ and the third element of the group signature $s = s' + s_1 + s_2 + \ldots + s_m$.

The group signature to of the document $M$ is a tuple $(U, E, \Sigma)$, consisted of one EC point and two numbers. The verification procedure includes the following steps. The verifier computes the hash of the document $M: H = F_H(M)$.

Using the group public key $L$ and signature $(U, E, \Sigma)$ he computes the EC point

$$\bar{R} = sG - e(U + L). \quad (7)$$

He computes the value $\bar{e} = F_H \left( M \parallel x_R \parallel x_U \right)$ and compares the values $\bar{e}$ with $e$. If $\bar{e} = e$, then the verifier concludes that the group signature is valid.

In case of 80-bit security, it is possible to use EC with parameters $p$ and $q$ having size approximately 160 bits. If EC point $U$ is represented by its abscissa $x_U$ and an extra bit is defined by one of two possible EC points with abscissa $x_U$, the signature length equals to 480 bits.
Thus, the implementation of the proposed protocol allows decreasing the signature length 3 times, compared to the original variant of the protocol implementation.

Proof of the protocol correctness is as follows:

\[
\begin{align*}
\tilde{R} &= \left( s + \sum_{i=1}^{n} s_i \right) \cdot G - e \left( L + \sum_{i=1}^{m} \lambda_i P_i \right) = \\
&= t' + z e + \sum_{i=1}^{n} (t_i + k_i \lambda_i e) \cdot G - e \left( z G + \sum_{i=1}^{m} \lambda_i k_i G \right) = \\
&= t' + z e + \sum_{i=1}^{n} t_i + k \sum_{i=1}^{n} k_i \lambda_i e = \\
&= t' + \sum_{i=1}^{n} t_i \cdot G = t' G + \sum_{i=1}^{n} t_i \cdot G = \\
&= R' + \sum_{i=1}^{n} R_i = R \Rightarrow \tilde{e} = F_H \left( M \parallel s_R \parallel x_U \right) = \\
&= F_H \left( M \parallel s_R \parallel x_U \right) = e.
\end{align*}
\]

(8)

IV. NEW MECHANISM OF MASKING PUBLIC KEYS OF THE ORIGINAL SIGNERS

In order to eliminate the dependency of the protocol security on the factorization problem we propose a new mechanism for identification of the original signers. In the original protocol, the first element \( U \) of the group signature is used to reveal the group members who generated the signature. Similarly to the original protocol the element \( U \) is calculated based on randomizing values \( \lambda_i \), depending on the document to be signed and public keys of the group members. Values \( \lambda_i \) are computed using secure one-way hash-function and a secret value \( \delta \) known only to the group manager. Namely, each individual masking parameter \( \lambda_i \) is calculated as follows \( \lambda_i = F_H(H[P_i || \delta]) \), where \( F_H \) is a hash-function, \( H \) is the hash-function value computed from a document to be signed, \( P_i \) is the public key of the \( i \)-th group member, \( \delta \) is the secret key of the group manager. || is the operation of string concatenation. This formula defines individual masking parameter for each user as it depends on their public key. The given value of \( i \)-th user \( \lambda_i \) is different for each document because the hash value from each document is included in the argument of the specified hash-function \( F_H \). On the other hand, no one except group manager could reveal the value of the masked public key of the given signer and document as parameter \( \lambda_i \) also depends on secret value \( \delta \) known only to group manager.

Furthermore, the proposed mechanism provides the anonymity of the signers for any person verifying the group signature, including the original signers.

However, such mechanism for computing the masking parameters has a problem associated with the identification of the original signers. When revealing the identity of the signers the group manager needs to provide proof that the identified set of signers really produced the given group signature. To do this, the group manager has to present masking coefficients and to demonstrate that they were calculated using the public keys of identified signers and the formula \( \lambda_i = F_H(H[P_i || \delta]) \). This means that the group manager has to open his secret parameter \( \delta \). Hence, every time when the users have to produce group signature, the group manager has to generate new secret value \( \delta \). This necessity can cause certain inconvenience to the group manager.

This disadvantage can be eliminated by using another formula for calculating individual masking parameters: \( \lambda_i = F_H(F_H(H[P_i || \delta]) || H[P_i]) \). In this case when proving the identity of the signers, the group manager has to present values \( F_H(H[P_i || \delta]) \), \( H \) and \( P_i \). Therefore, he has to keep his secret parameter \( \delta \) unrevealed. The revealing of the value \( F_H(H[P_i || \delta]) \) does not give any information to potential malefactor as this parameter is valid only for the given document and one particular public key \( P_i \). It can not be used to identify group members who signed another document.

V. CONCLUSION

In this paper, the implementation of the group signature protocol based on usage of elliptic curve is proposed, which allows decreasing of the signature length and increasing efficiency of the signature generation procedure. The implementation of the masking mechanism is used to conceal original group members and do not influence on the size of generated signature. To remove the dependency of the protocol implementation on factorization problem we propose a new mechanism for masking public keys of the original signers. This mechanism is based on computing the hash-function value from argument depending on the document to be signed such as public keys of the original signers and an additional secret value of the group manager.

REFERENCES


