CONSTRUCTING NEW COLLECTIVE SIGNATURE SCHEMES BASE ON TWO HARD PROBLEMS FACTORING AND DISCRETE LOGARITHM

Tuan Nguyen Kim[1], Nguyen Tran Truong Thien[1],
Duy Ho Ngoc[2] and Nikolay A. Moldovyan[3]

[1]School of Computer Science - Duy Tan University, Da Nang, Vietnam
[2]Department of Information Technology, Ha Noi, Vietnam
[3]SPIIRAS, St. Petersburg, Russia

ABSTRACT

In network security, digital signatures are considered a basic component to developing digital authentication systems. These systems secure Internet transactions such as e-commerce, e-government, e-banking, and so on. Many digital signature schemes have been researched and published for this purpose. In this paper, we propose two new types of collective signature schemes, namely i) the collective signature for several signing groups and ii) the collective signature for several individual signings and several signing groups. And then we used two difficult problems factoring and discrete logarithm to construct these schemes. To create a combination of these two difficult problems we use the prime module p with a special structure: p = 2n +1. Schnorr's digital signature scheme is used to construct related basic schemes such as the single signature scheme, the collective signature scheme, and the group signature scheme. The proposed collective signature schemes are built from these basic schemes. The proposed signature scheme is easy to deploy on existing PKI systems. It can support PKIs in generating and providing a unique public key, a unique digital signature, and a unique digital certificate for a collective of many members. This is essential for many collective transactions on today's Internet.

KEYWORDS

Network security, digital signature authentication, collective signature, group signature, signing group.

1. INTRODUCTION

To ensure the security of transactions on the Internet, people often use authentication systems based on digital signatures. A digital signature not only supports "authentication" of the origin of information but also helps to check the "integrity" of information when it is transmitted from source to destination and prevent the "non-repudiation" of a communication partner.

Most of the existing authentication systems are built on the basis of single digital signature schemes, so it can only support the validation of an individual signer, it is difficult to validate for a collective of many signers. In this paper, we propose and build a signature scheme that can support authentication for a group of signers, with different functions, with only a single public key and signature. This new authentication request is described below.

Assume that there is a collective made up of several groups, each of which has a large number of members and is managed by a group leader. There are another few individual members in this collective that do not belong to any groups, but they are functionally equivalent to the group

DOI: 10.5121/ijcnc.2022.14207

leaders. The problem is how to create a single digital signature [1-2] that represents this collective. The requirement of digital signature-based authentication [3-4] for a multi-functional collective is quite common in today's cyberspace. Both group signature protocols [5-9] and collective signature [10] ones can be used to produce a unique signature for a group of multiple signers, but they cannot be used to generate a common signature for a multi-level signing collective as described above. The reason for this is that the group signature scheme [11] can only create a common signature for each group, and the collective signature scheme [12] can only generate a common signature for the group leaders and individual members, or for all collective members [12]. Therefore, we propose a new type of multi-signature scheme, the representative collective signature scheme, which is structured from the combination of the group signature scheme and the collective signature scheme.

Two stages are required to create the representative collective signature. Firstly, the group signature protocol is used to establish group signatures for each group of the collective. The collective signature protocol is then used to generate collective signatures from each group and every other individual. The final signature represents a signing collective made up of several signing groups and individual signers, and it comprises the information of everyone who participated in the formation of this signature.

Most of the digital signature schemes can be built based on a difficult problem or at the same time two difficult problems [13-15]. In this article, we utilize Schnorr's digital signature standard [16] to develop two types of representative collective signature schemes using two tough challenges simultaneously. For the discrete logarithm problem [17-18], we use a specially structured prime modulo, $p = 2n + 1$, where $n = q'q$; q' and q are two large primes of magnitude 512 bits, or 1024 bits, used as the signer's private key. When attempting to find q' and q from n, the factorization problem [19-20] is applied.

2. THE RELATED BASE DIGITAL SIGNATURE SCHEMES

The Schnorr digital signature protocol is built on the difficult problem of the discrete logarithm in prime fields, with the input parameter set selected according to the DSA digital signature standard, but without constraints on size and structure of p and q. We propose a modification from the Schnorr scheme by i) Choosing prime modulus with special structure, $p = 2n + 1$, where $n = q'q$; q' and q are large prime numbers having the 512 bit size or more (the primes q' and q are such that the value 3 does not divide $q' - 1$ nor $q - 1$); ii) Change the expression for calculating the value S in the the signature generation procedure and iii) Change the expression R^* in the signature checking procedure (S is replaced by the parameter S^2). A new prime modulus has been used for constructing the randomized signature security of which is based on the factorization of the value $n = (p - 1)/2$.

2.1. The Single Signature Scheme (The SDS-2.1 scheme)

In this scheme we select the parameter α having the order $n\ modulo\ p$. The primes q' and q are elements of the private key.

We assume that the signer has a secret key $x\ (1 < x < n - 1)$, x is chosen at random. The private key of the signer is x. His/Her corresponding public key y: $y = \alpha^x\ mod\ p$.

Let F_H be a one-way hash function such as SHA-1 or SHA- 2, which produces the hash value H from the document M: $H = F_H(M)$.

The signature scheme based on factoring and discrete logarithm problems is described as below:

- **The signature generation procedure on the document M:**

It includes the following steps:

1. The signer generates the random value k, $k < n$, and then computes the value R:

$$R = \alpha^{k} \, mod \, p \tag{1}$$

2. The signer computes the value E:

$$E = RH \, mod \, \delta, \tag{2}$$

where δ is a large prime, $|\delta| = 160$ bits; and H is a hash value of the document M.

The value E is the first part of the signature.

3. The signer computes the value S:

$$S = (k + xE)^{1/2} mod \, n \tag{3}$$

such that:

$$R = \alpha^{S^2} y^{-E} mod \, p \tag{4}$$

The pair of value (E, S) is the signer's signature on the document M.

- **The signature verification procedure on the document M:**

It includes the following steps:

1. The verifier computes the value R^*:

$$R^* = \alpha^{S^2} y^{-E} mod \, p \tag{5}$$

2. The verifier computes the value E^*:

$$E^* = R^* H \, mod \, \delta, \tag{6}$$

3. The verifier compares values E^* with E. If $E^* = E$: The signature is valid; Otherwise the signature is invalid. It is rejected.

- **Proof of correctness of the SDS-2.1 scheme**:

To prove the correctness of this signatue scheme we only need to prove the existence of the equation $E^* = E$.

It is easy to see $R^* = R$. Indeed:

$$\begin{aligned} R^* &= \alpha^{S^2} y^{-E} \, mod\, p \\ &= \alpha^{(k+xE)} (\alpha^x)^{-E} \, mod\, p \\ &= \alpha^{k+xE} \alpha^{-xE} mod\, p \\ &= \alpha^k \, mod\, p \, = \, R \end{aligned}$$

Since $R^* = R$ so $E^* = E$ ($E^* = R^* H\, mod\, \delta = RH\, mod\, \delta = E$) is always exists.

The correctness of the SDS.2-1 scheme has been proved.

The collective signature scheme described below (the CDS-2.2 scheme) is built on the basis of this signature scheme (the SDS-2.1 scheme).

2.2. The Collective Signature Scheme (the CDS-2.2 scheme)

We assume that there are m signers in the signing collective, $1 \le i \le m$, to sign the same document M. Each signer randomly selects an integer x_i from the interval $[1, n-1]$ and computes a corresponding public key: $y_i = \alpha^{x_i} \, mod\, p$ (x_i is the secret key of the i-th user).
The collective signature scheme based on factoring and discrete logarithm problems (CDS-2.2) is described as below:

- **The collective signature generation procedure on the document M**

It includes the following steps:

1. Each signer selects a random number k_i, $k_i \in [1, n-1]$, and then computes the value R_i:

$$R_i = \alpha^{k_i} \, mod\, p \qquad (7)$$

The signer sends R_i to all other signers in the signing collective.

2. One of the signers in the signing collective, or a element in the PKI system, calculates the common randomization value R:

$$R = \prod_{i=1}^{m} R_i \, mod\, p \qquad (8)$$

Anh calculates the first part of the collective signature:

$$E = RH\, mod\, \delta \qquad (9)$$

where δ is a large prime, $|\delta| = 160$ bits; and H is a hash value of the document m.

The value E is sent to all signers in the signing collective.

3. Each signer computes it's a shared signature S_i:

$$S_i = (k_i + x_i E)^{1/2} mod\, n. \qquad (10)$$

4. One of the signers in the signing collective, or a element in the PKI system, calculates the

second element of the collective digtal signature S:

$$S = (\sum_{i=1}^{m} S_i{}^2)^{1/2} \, mod\, n \tag{11}$$

The pair of value (E, S) is the collective digital signature of the signing collective, there are m signers, on the message M.

- **The signature verification procedure on the document M**

It includes the following steps (the verifier can be a element in the PKI system):

1. The verifier computes the collective public key y:

$$y = \prod_{i=1}^{m} y_i \; mod\, p \tag{12}$$

2. The verifier computes the value R^*:

$$R^* = \alpha^{S^2} y^{-E} mod\, p. \tag{13}$$

3. The verifier computes the value E^*:

$$E^* = R^* H \, mod\, \delta. \tag{14}$$

4. The verifier compares values E^* and E. If $E^* = E$: The signature is valid; Otherwise the signature is invalid. It is rejected.

- **Proof of correctness of the CDS-2.2 scheme**:

To prove the correctness of this signatue scheme we only need to prove the existence of the equation $E^* = E$.

It is easy to see $R^* = R$. Indeed:

Substituting the value $S = (\sum_{i=1}^{m} S_i{}^2)^{1/2} \, mod\, n$ in the right part of the verification equation $R^* = \alpha^{S^2} y^{-E} \, mod\, p$, we get:

$$R^* = \alpha^{\sum_{i=1}^{m} S_i{}^2} \prod_{i=1}^{m} y_i^{-E} \, mod\, p$$
$$= \prod_{i=1}^{m} \alpha^{S_i^2} \prod_{i=1}^{m} \alpha^{x_i(-E)} \, mod\, p$$
$$= \prod_{i=1}^{m} \alpha^{k_i + x_i E} \prod_{i=1}^{m} \alpha^{x_i(-E)} \, mod\, p$$

$$= \prod_{i=1}^{m} \alpha^{k_i} \, mod \, p$$

$$= \prod_{i=1}^{m} R_i \, mod \, p = R$$

Since $R^* = R$ so $E^* = E$ ($E^* = R^* H \, mod \, \delta = RH \, mod \, \delta = E$) is always exists.

The correctness of the signature scheme has been proved.

It is easy to see that, in this scheme, none of the signers generates his/her individual signature. The signer generates only its shared signature in the collective signature that corresponds exactly to the given document M and to the assigned set of m users. Besides, it is computationally difficult to manipulate with shares $S_1, S_2, \ldots, S_m$, and compose another collective digital signature, relating to some different set of users.

3. The proposed Signature Schemes

In this part, we first construct a group signature scheme for a signing group of m members using the group signature protocol provided in [8]. Then, we utilize this scheme and the collective signature scheme mentioned in section 2.2, as the basic schemes, to build two types of the representative collective signature scheme: i) the collective signature for several signing groups and ii) the collective signature for several individual signings and several signing groups

3.1. Constructing The Group Signature Scheme (GDS-3.1)

Suppose there is a signing group of m signers who want to sign the document M. Each of the signers selects a private key x. His/Her corresponding public key is $y_i = \alpha^{x_i} \, mod \, p$, $i = 1, 2, \ldots, m$. The public key Y of the group manager is a public key of the group and is calculated as follows $Y = \alpha^X \, mod \, p$, where X is the manager's private key. The group manager, can be a element in the PKI system. The value Y is used in the signature verification procedure of the GDS-3.1 scheme. Let F_H is some specified hash function.

The group signature scheme based on factoring and discrete logarithm problems (GDS-3.1) is described as follows:

- **The group signature generation procedure on the document M**

It consists of stages:

1. The group manager does the following tasks:

- Computes hash value from document M:

$$H = F_H(M) \quad (15)$$

- Calculates masking coefficients λ_i:

$$\lambda_i = F_H(H \,||\, y_i \,||\, F_H(H \,||y_i||\, X)) \quad (16)$$

- Sends each value λ_i to the corresponding *i-th* group member
- Computes the first element of the group signature U:

$$U = \prod_{i=1}^{m} y_i^{\lambda_i} \, mod\, p \tag{17}$$

2. Each *i-th* signer in the signing group does the following tasks:

- Generates a random number $k_i, k_i < n$, anh then computes the value R_i:

$$R_i = \alpha^{k_i} \, mod\, p \tag{18}$$

- Sends $R_i$to the group manager

3. The group manager does the following tasks:

- Generates the random number K, $K < q$, and then computes the values R', R, E:

$$R' = \alpha^{K} mod\, p \tag{19}$$

$$R = R' \prod_{i=1}^{m} R_i \, mod\, p = \alpha^{K+\sum_{i=1}^{m} k_i} \tag{20}$$

and

$$E = F_H(M||R||U) \, mod\, \delta \tag{21}$$

where δ is a large prime, |δ| = 160 bit.

- Sends value E to all signers in signing group

E is the second element of the group signature.

4. Each *i-th* signer in the signing group does the following tasks:

- Computes his/her shared signature S_i:

$$S_i = (k_i + x_i \lambda_i E)^{1/2} mod\, n \tag{22}$$

- Sends S_i to the group manager

5. The group manager does the following tasks:

- Verifies the correctness of each shared signature S_i by checking equality:

$$R_i = \alpha^{{S_i}^2} y^{-\lambda_i E} mod\, p \tag{23}$$

- If all signature shared signatures S_i satisfy the last verification equation, then he/she computes his shared signature:

$$S' = (K + XE)^{\frac{1}{2}} \, mod \, n \qquad (24)$$

- Computes the third element of the group signature S:

$$S = (S'^2 + \sum_{i=1}^{m} S_i{}^2)^{1/2} \, mod \, n \qquad (25)$$

The tuple (U, E, S) is a group signature of the signing group on the document M.

- **The signature verification procedure on the document M**

It includes the following steps (The verifier can be a element in the PKI system):

1. The verifier computes the hash function value from the document *M:*

$$H = F_{\mathrm{H}}(M)$$

2. The verifier computes value R^*:

$$R^* = \alpha^{S^2}(UY)^{-E} mod \, p \qquad (26)$$

3. The verifier computes value E^*:

$$E^* = F_H(M||R^*||U) \, mod \, \delta \qquad (27)$$

4. The verifier compares the values E^* with E. If $E^* = E$: The group signature is valid; Otherwise, the group signature is invalid. It is rejected.

- **Proof of correctness of this signature scheme**:

To prove the correctness of this signatue scheme we only need to prove the existence of the equation $E^* = E$.

It is easy to see $R^* = R$. Indeed:

$$\begin{aligned} R^* &= \alpha^{S^2}(UY)^{-E} \, mod \, p \\ &= \alpha^{S'^2 . \sum_{i=1}^{m} S_i{}^2} \left(\alpha^X \prod_{i=1}^{m} y_i{}^{\lambda_i} \right)^{-E} \mathrm{mod}\, p \\ &= \alpha^{(K+XE). \sum_{i=1}^{m} (k_i + x_i \lambda_i E)} \left(\alpha^{-XE} \prod_{i=1}^{m} \alpha^{-x_i \lambda_i E} \right) \mathrm{mod}\, p \\ &= \alpha^{K + \sum_{i=1}^{m} k_i} \, mod \, p = R \end{aligned}$$

Since $R^* = R$ so $E^* = E$ $(E^* = R^* H \, mod \, \delta = RH \, mod \, \delta = E)$ is always exists.

The correctness of the signature scheme has been proved.

3.2. Constructing the Collective Digital Signature For Several Signing Groups

Let g signing groups with public keys $Y_j = \alpha^{X_j} \bmod p$, where $j = 1,2,\dots,g$. X_j is the secret key of the *j-th* goup manager, have intention to sign the document M. Suppose also the *j-th* signing goup inclues m_j active individual signers (persons appointed to act on behalf of the *j-th* signing goup).

The collective signature scheme for several signing group (RCS.01-3.2) is described as below.

- **The collective signature generation procedure on the document M**

It consists of stages:

1. Each *j*-th *group* manager in the signing collective does the following tasks:

- Based on the group signature generation procedure described above (section 3.1) to generals masking parameters λ_{ji} for the signers of *j*-th *group*.
- Computes the value U_j (where $i = 1, 2, \dots, m_j$):

$$U_j = \prod_{i=1}^{m_j} y_{ji}^{\lambda_{ji}} \bmod p \tag{28}$$

U as the shared element of the *j*-th *group* in the first element of the collective signature.

- Comutes the randomizing parameter R_j:

$$R_j = R'_j \prod_{i=1}^{m_j} R_{ji} \bmod p \tag{29}$$

- Sends values U_j and R_j to all other group managers in the signing collective.

2. Each *j-th* group manager in the signing collective computes values U, R and E:

$$U = \prod_{j=1}^{g} U_j \bmod p \tag{30}$$

$$R = \prod_{j=1}^{g} R_j \bmod p = \alpha^{\sum_{j=1}^{g} k_j} \bmod p \tag{31}$$

and

$$E = F_H(M||R||U) \bmod \delta \tag{32}$$

U and E are the first and second elements of the collective signature.

3. Each *j-th* group manager does the following tasks:

- Computes the shared signature of j-th group:

$$S_j = \left(S_j'^2 + \sum_{i=1}^{m_j} S_{ji}^2 \right)^{1/2} \quad mod\ n \tag{33}$$

Where S_{ji} in the shared signature of the *i-th* signer in the *j-th* group,

- Sends S_j to other group managers in the signing collective.

4. Each *j-th* group manager does the following tasks:

- Can verify the correctness of each shared signature S_j by cheaking equality:

$$R_j^* = \alpha^{S_j^2} (U_j Y_j)^{-E}\ mod\ p \tag{34}$$

- If all shared signatures S_j satisfy the last verification equation, then the third element S of the collective signature is computed:

$$S = (\sum_{j=1}^{g} S_j{}^2)^{1/2}\ mod\ n \tag{35}$$

The tuple (U, E, S) is the collective signature on the document M of the signing collective there are g signing groups.

- **The signature verification procedure on the document M**

It includes the following steps (The verifier can be a element in the PKI system):

1. The verifier computes the collective public key shared by all signing groups:

$$Y_{col} = \prod_{j=1}^{g} Y_j\ mod\ p \tag{36}$$

2. The verifier computes the value R^*:

$$R^* = \alpha^{S^2} (UY_{col})^{-E}\ mod\ p \tag{37}$$

3. The verifier computes the value E^*:

$$E^* = \ F_H(M||R^*||U)\ mod\ \delta \tag{38}$$

4. The verifier Compares the values E^* with E. If $E^* = E$: The collective signature is valid. Otherwise, the collective signature is invalid. It is rejected.

- **Proof of correctness of this signature scheme**:

To prove the correctness of this signatue scheme we only need to prove the existence of the equation $E^* = E$.

It is easy to see $R^* = R$. Indeed:

$$R^* = \alpha^{S^2}(UY_{col})^{-E} \, mod\, p$$

$$= \alpha^{\sum_{j=1}^{g} S_j{}^2}(\prod_{j=1}^{g} U_j \prod_{j=1}^{g} Y_j)^{-E} \, mod\, p$$

$$= \prod_{j=1}^{g} \alpha^{S_j{}^2}(Y_j U_j)^{-E} \, mod\, p$$

$$= \prod_{j=1}^{g} R_j \, mod\, p = R$$

Since $R^* = R$ so $E^* = E$ ($E^* = F_H(M||\, R^*||\, U) = F_H(M||\, \mathrm{R}||\, U) = E$) is always exists.

The correctness of the signature scheme has been proved.

3.3. Constructing the Collective Digital Signature Scheme for Several Individual Signers and Several Signing Groups

The collective signature generation procedure of this scheme is similar to that of the RCS.01-3.2 scheme, but for individual signers, Uj is equal to 1.

Suppose x_j and $y_j = \alpha^{x_j}$, where $j = g+1, g+2, \dots, g+m$, are a private key and a public key, correspondingly, of m individual signers participating in the protocol for generating the collective digital signature for *g* signing groups and *m* individual signers.

The collective signature scheme for m individual signers g signing groups (RCS.02-3.3) is described as below.

- **The signature generation procedure on the document M**

It consists of stages:

1. Each *j*-th *group* manager in the signing collective does the following tasks:

- Based on the group signature generation procedure described above (section 3.1) to generals masking parameters λ_{ji} for the signers of *j*-th *group*.
- Computes the value U_j (where $i = 1,2, \dots, m_j$):

$$U_j = \prod_{i=1}^{m_j} y_{ji}^{\lambda_{ji}} \, mod\, p \qquad (39)$$

U as the shared element of the *j*-th *group* in the first element of the collective signature.

- Computes the randomizing parameter R_j:

$$R_j = R_j' \prod_{i=1}^{m_j} R_{ji} \bmod p \tag{40}$$

- Send values U_j and R_j to all other managers and all individual signers in the signing collective.

2. Each *j-th* individual signer ($j = g+1, g+2, \dots, g+m$) does the following tasks:

- Generates a random value K_j, $K_j < n$, and then computes the value R_j:

$$R_j = \alpha^{K_j} \bmod p \tag{41}$$

- Sent R_j to all group managers and other individual signers in the signing collective.

- Each *j-th* group manager and each *j-th* individual signer in the signing collective computes values U, R and E:

$$U = \sum_{j=1}^{g+m} U_j \bmod p \tag{42}$$

$$R = \sum_{j=1}^{g+m} R_j \bmod p \tag{43}$$

And

$$E = F_H(M||R||U) \bmod \delta \tag{44}$$

where δ is a large prime having, $|\delta| = 160$ bits; $U = 0$ for $j = g+1, g+2, \dots, g+m$.

U and E are *the first and second elements* of the signature.

3. a) Each *j-th* group manager computes the shared signature of *j-th* group S_j:

$$S_j = ({S_j'}^2 + \sum_{i=1}^{m_j} S_{ji}^2)^{1/2} \bmod n \tag{45}$$

where S_{ji} is the shared signature of the *i-th* signer in the *j-th* signing group.

And sends S_j to all individual signers and other group managers.

b) Each *j-th* individual signer computes his/her shared signature S_j:

$$S_j = (K_j + X_j E)^{1/2} \bmod n \tag{46}$$

And sends S_j to all group managers and other individual signers.

4. Each *j-th* group manager and each individual signers does the following tasks:

- Can verify the correctness of each share signatures S_j by checking equality:
-

$$R_j^* = \alpha^{S_j^2}(U_jY_j)^{-E} \bmod p \tag{47}$$

For $j = 1, 2, \ldots, g$ and

$$R_j^* = \alpha^{S_j^2}Y_j^{-E} \bmod p \tag{48}$$

For $j = g+1, g+2, \ldots, g+m$.

- If all shares $\mathrm{S_j}$ satisfy the last verification equation, then the third element Sof the collective signature is computed:

$$S = (\sum_{j=1}^{g+m} {S_j}^2)^{1/2} \bmod n \tag{49}$$

The tuple (U, E, S) is the collective signature on the document M of the signing collective there are g signing groups and m individual signers.

The first element U of the collective signature contains information about the all group members of each signing group who signed the document M.

- **The signature verification procedure on the document M**

It includes the following steps (The verifier can be a element in the PKI system):

1. The verifier computes the collective public key shared by all signing groups and individual signers:

$$Y_{col} = \prod_{j=1}^{g+m} Y_j \bmod p \tag{50}$$

2. The verifier computes the value R^*:

$$R^* = \alpha^{S^2}(UY_{col})^{-E} \bmod p \tag{51}$$

3. The verifier computes the value E^*:

$$E^* = F_H(M||\, R^*||\, U) \tag{52}$$

4. The verifier Compares the value E^* with E. If $E^* = E$: The collective signature is valid; Otherwise, the collective signature is invalid. It is rejected.

- **Proof of correctness of this signature scheme:**

To prove the correctness of this signatue scheme we only need to prove the existence of the

equation $E^* = E$.
It is easy to see $R^* = R$. Indeed:

$$R^* = \alpha^{S^2}(UY_{col})^{-E} \, mod \, p$$

$$= \alpha^{\sum_{j=1}^{g+m} S_j{}^2}(\prod_{j=1}^{g+m} U_j \prod_{j=1}^{g+m} Y_j)^{-E} \, mod \, p$$

$$= \alpha^{\sum_{j=1}^{g} S_j{}^2 + \sum_{j=g+1}^{g+m} S_j{}^2}(\prod_{j=1}^{g} U_j \prod_{j=1}^{g} Y_j \prod_{j=g+1}^{g+m} Y_j)^{-E} \, mod \, p$$

$$= \prod_{j=1}^{g} \alpha^{S_j{}^2}\left(U_jY_j\right)^{-E} \prod_{j=g+1}^{g+m} \alpha^{S_j{}^2} Y_j{}^{-E} \, mod \, p$$

$$= \prod_{j=1}^{g} R_j \prod_{j=g+1}^{g+m} R_j \, mod \, p = R$$

Since $R^* = R$ so $E^* = E$ ($E^* = F_H(M||\, R^*||\, U) = F_H(M||\, \mathrm{R}||\, U) = E$) is always exists.

The correctness of the signature scheme has been proved.

4. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

4.1. Security analysis of the proposed collective digital signature schemes

4.1.1. Security level of the single digital signature (SDS-2.1)

It is easy to see that, the solution of the discrete logarithm problem in $GF(p)$ is not sufficient for breaking this signature scheme. To break the scheme it is required to know the factorization of n. Indeed, the solution of the discrete logarithm problem leads to the computation of the secret key x and the possibility to calculate the value $k + xE \, mod \, n$. However, to calculate the signature element$S$ is required to extract the *2-th* root modulo n from $k + xE \, mod \, n$. This requires factoring the modulus n. This is the second difficult problem.

4.1.2. Security level of the group digital signature scheme (GDS-3.1)

With the group signature scheme, there are two main types of attacks: Internal attacks and external attacks. In external attacks, the attacker only knows the system parameters and the public keys, along with the document M, while in internal attacks, the attacker will know a lot more information.

Let's take a look at the most likely successful case where the attacker is the group manager, since he has the most information.

- **Attack to reveal secret key:**

Assuming the signing group consists of m members. Since the group manager knows the values S_m, R_m, y_m so if he wants to attack the *m-th* person in the signing group he can do the following: He needs to calculate: $x_m = log_\alpha y_m \, mod \, p$; or computes: $k_m = \log_\alpha R_m \, mod \, p$; and then computes: $x_m = S_m^2 - xE \, mod \, n$. These require solving the discrete logarithm problem.

- **Signature forgery attack:**

Assuming the signing group consists of m members. The group manager of this signing group knows the values S_m, R_m, y_m so if the group manager wants to attack the *m-th* person in the signing group he perform the following steps:

Choose $X \in [1, n-1]$ and calculate his public key:

$$Y = y_m^{\lambda_m} \alpha^X mod\, p \tag{53}$$

And calculate the common public value of group.

$$U = \prod_{i=1}^{m} y_i^{\lambda_i}\, mod\, p \tag{54}$$

Choose $K \in [1, n-1]$ and compute:

$$R' = R_m \alpha^K\, mod\, p \tag{55}$$

Compute R and E, send E to all other member of group.

$$R = R' \sum_{i=1}^{m} R_i\, mod\, p \tag{56}$$

$$E = F_H(M||R||U)\, mod\, \delta \tag{57}$$

Compute:

$$S' = (S_m^2 + K + XE)^{1/2} mod\, n \tag{58}$$

And:

$$S = \left(S'^2 + \sum_{i=1}^{m} S_i^2\right)^{1/2} mod\, n \tag{59}$$

The tuple (U, E, S) still satisfy the test equation $R = \alpha^{S^2}(UY)^E\, mod\, p$. Because:

$$\begin{aligned}
R^* &= \alpha^{S^2}(UY)^E\, mod\, p \\
&= \alpha^{S'^2 + \sum_{i=1}^{m} S_i^2} \left(y_m^{\lambda_m} \alpha^X \prod_{i=1}^{m} y_i^{\lambda_i}\right)^{-E} mod\, p \\
&= \alpha^{S_m^2 + (K+XE) + \sum_{i=1}^{m} S_i^2} y_m^{-E\lambda_m} \alpha^{-EX} \prod_{i=1}^{m} y_i^{-E\lambda_i}\, mod\, p \\
&= \alpha^{(k_m + x_m\lambda_m E) + (K+XE) + \sum_{i=1}^{m}(k_i + x_i\lambda_i E)} \alpha^{-x_m\lambda_m E} \alpha^{-XE} \alpha^{\sum_{i=1}^{m} -x_i\lambda_i E} mod\, p \\
&= \alpha^{k_m + K + \sum_{i=1}^{m} k_i} = R' \sum_{i=1}^{m} R_i\, mod\, p = R
\end{aligned}$$

When deploying the scheme to prevent this type of attack, it is necessary to have a trusted department to act as the group manager. The PKI plays an important role in this case [21-22].

When building a signing group, that department is responsible for receiving the public key of each signing member, then calculating and publishing the public public key of the signing group, the public keys of the members must also be made public. Publicly announced in the signing group for all members of the group to know. The private-public keys of the members and the public keys of the whole group are fixed, and the attacker will not be able to recompute them as shown in expression (53). So the scheme is safe if implemented correctly (53).

The security level of the collective digital signature and the collective digital signature of signing groups are similar to Security level of digital signature for signing group we mention above.

4.2. Performance evaluation of the proposed collective digital signature schemes

The performance of a digital signature scheme can be evaluated by calculating the time cost of signature generation and the time cost of signature verification. We do it this way. The time costs of representative collective signature schemes proposed in this paper are shown in Table 1.

Notations: T_h: Time cost of a hash operation in Z_p; T_s: Time cost of a scalar multiplication in Z_p; T_{inv}: Time cost of a inverse operation in Z_p; T_e: Time cost of an exponent operation in Z_p; T_m: Time cost of a modular multiplication in Z_p. According to [23]: $T_h \approx T_m, T_s \approx 29T_m, T_{inv} \approx 240T_m,$

$T_e \approx 240T_m, T_{sqrt} \approx 290T_m.$

Table 1. Time cost of the proposed collective signature scheme: RCS.01-3.2 and RCS.02-3.3

The scheme	Time for Signature generation	Time for Signature verification
RCS.01-3.2	$U = \sum_{j=1}^{g}(243m_j + 1)\,T_m$ $e = [\sum_{j=1}^{g}(241m_j + 240) + 1]T_m$ $S = [\sum_{j=1}^{g}(1254m_j + 1781) + 290]T_m$ $Sum = [\sum_{j=1}^{g}(1738m_j + 2022) + 291]T_m$	$(723 + g)T_m$
RCS.02-3.3	$U = \sum_{j=1}^{g}(243m_j + 1)\,T_m$ $e = [\sum_{j=1}^{g}(241m_j + 240) + 241m + 1]T_m$ $S = [\sum_{j=1}^{g}(1254m_j + 1781) + 1250m + 290]T_m$ $Sum = [\sum_{j=1}^{g}(1738m_j + 2022) + 1491m + 292]T_m$	$(723 + g + m)T_m$

Table 1 shows that the time cost for the generation of signature components and for the signature verification of the proposed collective signature schemes are is much higher than that of the similar signature scheme in [24]. This is considered as a limitation that needs to be overcome for schemes built on two difficult problems factoring and discrete logarithm [25-27].

5. CONCLUSION

In this paper, we have shown that there is a new authentication requirement that requires collective key generation and signature generation algorithms to satisfy. Our proposed collective signature can meet this new requirement.

In addition, we have succeeded in using simultaneously two difficult problems factoring and discrete logarithm to build two types of representative collective signature schemes, which are: i) the collective signature scheme for many signing groups and ii) the collective signature scheme for many individual signers and many signing groups. These types of schemes are essential for the multi-level authentication requirements of many information exchange applications in today's network environment and it is also easy to deploy on existing PKI systems.

The simultaneous combination of two difficult problems factoring and discrete logarithm is demonstrated by choosing a prime modulo p with a special structure, $p = 2n + 1$ with $n = q'q$, q' and q are large prime numbers having the 512 bit size or 1024 bit. The security level of the proposed collective signature schemes is inherited from the base scheme which has been analyzed in section 4.1. That is, to break the proposed collective signature scheme, the attacker must also solve two difficult problems simultaneously.

The paper also calculated and compared the performance of the two proposed schemes with the performance of some other schemes.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Pieprzyk J., Hardjono T. & Seberry J., (2003) “Fundamentals of Computer Security”, Springer-Verlag, Berlin Heidelberg.

[2] National Institute of Standards & Technology, (2009) “Digital Signature Standard”, Federal Information Processing Standards Publication 186-3.

[3] Ganeshkumar K. & Arivazhagan D., (2014) “Generating A Digital Signature Based On New Cryptographic Scheme For User Authentication And Security”, Indian Journal of Science and Technology.

[4] Girault M., Poupard G. & Stern J., (2006) “On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order”, In Journal of Cryptology, no.19, pp.463-487.

[5] Seetha R. & Saravanan R., (2016) “Digital Signature Schemes for group communication: A Survey”, International Journal of Applied Engineering Research, no.11, pp.4416–4422.

[6] Enache A. C., (2012) “About Group Digital Signatures”, Journal of Mobile, Embedded and Distributed Systems, no.4, pp.193–202.

[7] Alamélou Q., Blazy O., Cauchie S. & Ph. Gaborit, (2017) “A code-based group signature scheme”, Designs, Codes and Cryptography, vol.82, no.1-2.

[8] Moldovyan A. A. & Moldovyan N. A, (2014) “Group signature protocol based on masking public keys”,Quasigroups and related systems, no.22, pp.133–140.

[9] Xie R., Xu C., He C. & Zhang X., (2016) “A new group signature scheme for dynamic membership”, International Journal of Electronic Security and Digital Forensics, vol.8, no.4.

[10] Moldovyan N. A., Nguyen Hieu Minh, Dao Tuan Hung & Tran Xuan Kien, (2016) “Group Signature Protocol Based on Collective Signature Protocol and Masking Public Keys Mechanism”, International Journal of Emerging Technology and Advanced Engineering, no.6, pp.1–5.

[11] Rajasree R. S., (2014) “Generation of Dynamic Group Digital Signature”, International Journal of Computer Applications, no.98, pp.1–5.

[12] Moldovyan N. A., (2011) “Blind Collective Signature Protocol”, Computer Science Journal of Moldova, no.19, pp.80–91.

[13] Tahat N., Ismail E., and Ahmad R., (2009) “A New Blind Signature Scheme Based on Factoring and Discrete Logarithms,” International Journal of Cryptology Research, vol.1, no.1, pp.1-9.

[14] Minh N., Binh D., Giang N. & Moldovyan N. A., (2012) “Blind Signature Protocol Based on Difficulty of Simultaneous Solving Two Difficult Problems”, Journal of Applied Mathematical Sciences, vol.6, no.139, pp.6903-6910.

[15] Berezin A., Moldovyan N. A. & Victor S., (2013) “Cryptoschemes Based on Difficulty of Simultaneous Solving Two Different Difficult Problems”, Computer Science Journal of Moldova, vol.21, no.2, pp.280-290.

[16] Schnorr C. P., (1991) “Efcient signature generation by smart-cards”, In Journal of Cryptology, vol.4, no.3, pp.161-174.

[17] Camenisch J. L., Piveteau J. -M. & Stadler M. A., (1995) “Blind Signatures Based on the Discrete Logarithm Problem”, In: Advances in Crypology – EUROCRYPT'94 Proc, Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York, vol.950, pp.428–432.

[18] Moldovyan N. A. & Moldovyan A. A, (2010) “Blind Collective Signature Protocol Based on Discrete Logarithm Problem”, Int. Journal of Network Security, no.11, pp.106–113.

[19] Nimbalkar A. B., (2018) “The Digital Signature Schemes Based on Two Hard Problems: Factorization and Discrete Logarithm”, Advances in Intelligent Systems and Computing, Cyber Security. vol.729, pp.493.498.

[20] Moldovyan N. A., (2011) “Blind Signature Protocols from Digital Signature Standards”, Int. Journal of Network Security, no.13, pp.22–30.

[21] Selvakumaraswamy S. & Govindaswamy U., (2016) “Efficient Transmission of PKI Certificates using ECC and its Variants”, The International Arab Journal of Information Technology, vol.13, no.1, pp.38-43.

[22] Shivkumar S. and Umamaheswari G., (2015) “Efficient Transmission of PKI Certificates using Elliptic Curve Cryptography and its Variants”, The International Arab Journal of Information Technology, pp. 38-43.

[23] 21Popescu C., (1999) “Blind signature and BMS using elliptic curves”, Studia univ babes–bolyai, Informatica, pp.43-49.

[24] 22Tuan N. K., Van V.L., Moldovyan D. N., Duy H. N. & Moldovyan A. A., (2018) “Collective signature protocols for signing groups”, In Proc. Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, India.

[25] 23Moldovyan N. A., (2008) “Digital Signature Scheme Based on a New Hard Problem”, Computer Science Journal of Moldova, no.16, pp.163–18.

[26] 24Lee J., Kim H., Lee Y., Hong S. M. & Yoon H., (2017) “Parallelized scalar multiplication on elliptic curves defined over optimal extension field”, International Journal of Network Security, vol.4, p.99-106.

[27] 26Chaum D., (1983) “Blind Signatures for Untraceable Payments”, Advances in Cryptology: Proc. of CRYPTO’82, Plenum Press, p.199–203.

AUTHORS

Tuan Nguyen Kim was born in 1969, received B.E., and M.E from Hue University of Sciences in 1994, and from Hanoi University of Technology in 1998. He has been a lecturer at Hue University since 1996. From 2011 to the present (2021) he is a lecturer at School of Computer Science, Duy Tan University, Da Nang, Vietnam. His main research interests include Computer Network Technology and Information.

Nguyen Tran Truong Thien was born in 1997, received B.E from Duy Tan University in 2020. He has been a security researcher at Duy Tan University since February 2021. His main research interests include is Network Security, Information Security and Machine learning for Cybersecurity.

Duy Ho Ngoc was born in 1982. He received his Ph.D. in Cybersecurity in 2007 from LETI University, St. Petersburg, Russia Federation. He has authored more than 45 scientific articles in cybersecurity.

Nikolay A. Moldovyan is an honored inventor of Russian Federation (2002), a laboratory head at St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, and a Professor with the St. Petersburg State Electrotechnical University. His research interests include computer security and cryptography. He has authored or co-authored more than 60 inventions and 220 scientific articles, books, and reports. He received his Ph.D. from the Academy of Sciences of Moldova (1981).