Integrating Multisignature Scheme into the Group Signature Protocol

Hung Dao Tuan[1(✉)], Hieu Minh Nguyen[2], Cong Manh Tran[3], Hai Nam Nguyen[2], and Moldovyan Nikolay Adreevich[4]

[1] National Laboratory of Information Security, Ha Noi, Viet Nam
daotuanhung@gmail.com

[2] Academy of Cryptography Techniques, Ha Noi, Viet Nam

[3] Le Quy Don Technical University, Ha Noi, Viet Nam

[4] Saint-Petersburg Electronical University, St. Petersburg, Russia

Abstract. This paper proposes two new variants of group signature protocols with and without distinguished signing authorities based on the multisignature signature scheme to reduce significantly the signature length and masking signers public keys. The proposed protocols do not include a secret sharing and knowledge proving procedure. Thus, these protocols allow a flexible modification of the group structure by the group manager. Compared to the known group signature protocols, our protocols are designed based on integrating the multisignature scheme into the group signature protocol.

Keywords: Digital signature · Distinguished signing responsibilities · Discrete logarithm problem · Group signature · Multisignature · Public key

1 Introduction

Nowadays, the electronic transactions over the internet are performed not only between two individuals but also between groups of people or different organizations. Therefore, the informative authentication is to identify the information of a group of people or an organization. To solve this problem, several schemes such as multisignature [1], group signature [2], ring signature [3] and traditional signatures [4,5] have been proposed. Recently, group-oriented digital signatures [1–3] are widely used.

The multisignature of an electronic document is generated when each of the designated users signs the document [6–8]. Thus, this signature can be considered as a representative of m individual signatures. Each of the signers is responsible for the content of document, which is signed. To verify a given multisignature, the public keys of all users who generated the signature need to be used. In addition, the multisignature schemes and individual signature schemes can use the same public key infrastructure (PKI). The implementation of these schemes allows changing a set of signers arbitrarily. The last two properties of the protocols represent the important characteristics of the multisignature.

M. Akagi et al. (eds.), *Advances in Information and Communication Technology*,
Advances in Intelligent Systems and Computing 538, DOI 10.1007/978-3-319-49073-1_32

The group signature of an electronic message is generated by a group of signers, and one of them is a group manager [2,9–11]. To verify the group signature, a group public key needs 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 members can sign the document. Secondly, the group manager, who has both the document and the 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. Group manager is a trusted party of the group signature protocol. He creates the secret parameters, which are used to generate the signature by the signers.

In this paper, two new group signature schemes are proposed to provide both the features of the multi-signature scheme and the group signature. The proposed schemes implement signing steps using the methods generating the signature of both the multi-signature and the group signature schemes.

This design provides a possibility to keep the individual public keys of group members in secret. In order to obtain this property, the group signature is generated in two steps. First, the pre-signature of group is provided. Second, the group manager computes the group signature using a given approval based on pre-signature. The pre-signature can be generated by any group member or a subset of group members. Since only the group manager can reveal the original signers of the document, the distribution of the secret values and using the PKI are not required in the proposed group signature protocol. Moreover, the set of signers involved in the group can be arbitrarily changed by the group manager, and the public key of the group manager is used as the group public key. As the result, the group signature protocols provide a processing procedure of document very close to daily practices such as a letter document preparation, signing and the approval.

Furthermore, in the proposed schemes, the group manager has an ability to define the individual of the group to sign a part of the document, and signing authorities group signature are distinguished. The proposed group signature protocols can significantly reduce the signature length up to 640 bits in the case of 80-bit security. The design of the proposed protocols is only based on the computational difficulty of the elliptic curve discrete logarithm problem.

The rest of the paper is organized as follows. In Sect. 2, the implementation of the protocols based on the elliptic curve over finite field is presented. The analysis results of the proposed signature group protocols are discussed in Sect. 3. Finally, the conclusion is given in Sect. 4.

2 The Proposed Group Signature Protocols

Currently, cryptographic protocols based on elliptic curves (EC) over finite field have been widely used, and digital signature schemes based on ECDLP have attracted many researches [12]. In this paper, we propose two protocols for the group digital signature based on the multisignature with the ECschnorr digital signature scheme [12].

2.1 Proposed Group Signature Protocol 1

In the proposed protocol, the group manager (GM) will generate text messages needed to sign M. Group manager will define a structure of the group of people who will sign (Group manager will decide the members involved in the signing process). Determining the members, who will sign the text message, is based on the calculation of random values z_i. The random values z_i are calculated based on three parameters such as one-way functions (P_i) of public keys of signing members, hash values (h) of the text message which needs to be signed, and a secret value which is known by only the Group manager.

The group manager will calculate the values $h = H(M)$ and $z_i = H(H(h||P_i|| SE)||h||P_i)$. Then, the values (z_i, h) are sent to i-th signer. To generate group signature, the following procedures are performed.

Step 1: Key Generation Phase

1. Each member in the group of signers generates their private key as a random number k_i $(1 < k_i < q)$ and public key computed as the point $P_i = k_iG$, with $i = (1, 2, \ldots, m)$.

2. The group manager computes his public key as the point $P_{gm} = k_{gm}G$, where k_{gm} is his private key. The public group key P_{gm} is used to verify the group signature.

Step 2: Group Signature Generation Phase

1. The group manager computes EC point $U = z_1P_1 + z_2P_2 + \ldots + z_mP_m$, which serves as the first element of the group signature.

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

3. The group manager generates the random number $1 < t_{gm} < q$. Then, EC points $R_{gm} = t_{gm}G, R = R_{gm} + R_1 + R_2 + \ldots + R_m$ are computed. Moreover, the second element of the group signature $e = H(M||x_R||x_U)$, where x_R and x_U are x-coordinates of EC points R and U, respectively, is also calculated. He sends the value e to the group members who initiated the protocol.

4. Each i-th signer computes their signature share $s_i = t_i + k_iz_ie \bmod q$ and sends it to the group manager.

5. The group manager verifies the correctness of each s_i by checking $R_i = s_iG - z_ieP_i$. If all signature shares s_i satisfy the verification procedure, then he computes his share $s_{gm} = t_{gm} + k_{gm}e \bmod q$ and the third element of the group signature $s = s_{gm} + s_1 + s_2 + \ldots + s_m$.

The group signature of the document M is a tuple (U, e, s), which consists of one EC point and two numbers.

Step 3: Group Signature Generation Phase

1. The verifier computes the hash of the document M as $h = H(M)$.

2. Using the group public key P_{gm} and the signature (U, e, s), he computes the EC point $\tilde{R} = sG - e(U + P_{gm})$.

3. He computes the value $\tilde{e} = H(M||x_{\tilde{R}}||x_U)$ and compares the values $\tilde{e}$ and e. If $\tilde{e} = e$ then the verifier concludes that the group signature is valid.

2.2 Proposed Group Signature Protocol 2 with Distinguished Signing Authorities

In this protocol, the group manager will generate text messages needed to sign $M = M_1||M_2||\ldots||M_m$ (we only consider cases in which the number of text messages needed to sign equals the number of persons who signed).

Group manager will define a structure of the group of people who will sign (Group manager will decide the members involved in the signing process). Determining the members, who will sign the text message, is based on the calculation of random values z_i. The random values z_i are calculated based on three parameters such as one-way functions (P_i) of public keys of signing members, hash values (h_i) of the text message which needs to be signed, and a secret value which is known by only the group manager.

The group manager will calculate the values $h_i = H(M_i)$ and $z_i = H(H(h_i||P_i||SE)||h_i||P_i)$. Then, the values (z_i, h_i) are sent to i-th signer. To generate group signature, the following procedures are performed.

Step 1: Key Generation Phase (similar to step 1 of protocol 1)

1. Each member in the group of signers generates their private key as a random number k_i $(1 < k_i < q)$ and public key computed as the point $P_i = k_iG$, with $i = (1, 2, \ldots, m)$.

2. The group manager computes his public key as the point $P_{gm} = k_{gm}G$, where k_{gm} is his private key. The public group key P_{gm} is used to verify the group signature.

Step 2: Group Signature Generation Phase

1. The group manager computes EC point $U = h_1z_1P_1 + h_2z_2P_2 + \ldots + h_mz_mP_m$, which serves as the first element of the group signature.

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

3. The group manager generates the random number $1 < t_{gm} < q$. Then, EC points $R_{gm} = t_{gm}G, R = R_{gm} + R_1 + R_2 + \ldots + R_m$ are computed. Moreover, the second element of the group signature $e = H(M||x_R||x_U)$, where x_R and x_U are x-coordinates of EC points R and U, respectively, is also calculated. He sends the value e to the group members who initiated the protocol.

4. Each i-th signer computes their signature share $s_i = t_i + k_iz_ih_ie \bmod q$ and sends it to the group manager.

5. The group manager verifies the correctness of each s_i by checking $R_i = s_iG - z_ih_ieP_i$. If all signature shares s_i satisfy the verification procedure, then he computes his share $s_{gm} = t_{gm} + k_{gm}e \bmod q$ and the third element of the group signature $s = s_{gm} + s_1 + s_2 + \ldots + s_m$.

The group signature of the document M is a tuple (U, e, s), which consists of one EC point and two numbers.

Step 3: Group Signature Generation Phase

1. The verifier computes the hash of the document M as $h = H(M)$.

2. Using the group public key P_{gm} and the signature (U, e, s), he computes the EC point $\tilde{R} = sG - e(U + P_{gm})$.

3. He computes the value $\tilde{e} = H(M||x_{\tilde{R}}||x_U)$ and compares the values $\tilde{e}$ and e. If $\tilde{e} = e$ then the verifier concludes that the group signature is valid.

3 Analysis of the Proposed Group Signature Protocols

3.1 Correctness

Proof of the protocol 1 correctness is as follows:

$$\tilde{R} = (s_{gm} + \sum_{i=1}^{n} s_i)G - e(P_{gm} + \sum_{i=1}^{m} z_i P_i)$$

$$= (t_{gm} + k_{gm}e + \sum_{i=1}^{n}(t_i + k_i z_i e))G - e(k_{gm}G + \sum_{i=1}^{m} z_i k_i G)$$

$$= (t_{gm} + k_{gm}e + \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} k_i z_i e - e k_{gm} - \sum_{i=1}^{n} k_i z_i e)G$$

$$= (t_{gm} + \sum_{i=1}^{n} t_i)G = t_{gm}G + (\sum_{i=1}^{n} t_i)G$$

$$= R_{gm} + \sum_{i=1}^{n} R_i = R \Rightarrow \tilde{e} = H(M||x_{\tilde{R}}||x_U) = H(M||x_R||x_U) = e$$

Proof of the protocol 2 correctness is as follows:

$$\tilde{R} = (s_{gm} + \sum_{i=1}^{n} s_i)G - e(P_{gm} + \sum_{i=1}^{m} h_i z_i P_i)$$

$$= (t_{gm} + k_{gm}e + \sum_{i=1}^{n}(t_i + k_i z_i h_i e))G - e(k_{gm}G + \sum_{i=1}^{m} z_i k_i h_i G)$$

$$= (t_{gm} + k_{gm}e + \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} k_i z_i h_i e - e k_{gm} - \sum_{i=1}^{n} k_i z_i h_i e)G$$

$$= (t_{gm} + \sum_{i=1}^{n} t_i)G = t_{gm}G + (\sum_{i=1}^{n} t_i)G$$

$$= R_{gm} + \sum_{i=1}^{n} R_i = R \Rightarrow \tilde{e} = H(M||x_{\tilde{R}}||x_U) = H(M||x_R||x_U) = e$$

3.2 Signature Length

In the proposed protocols, In the case of 80-bit security, it is possible to use EC with parameter q having size approximately 160 bits. Digital signatures on document M consists of three components (U, e, s), the signature length is equal to 640 bits. Compared to the lengths of previous group signature schemes [9–11], the signature length of the proposed protocols are significantly reduced.

3.3 Mechanism of Masking Public Keys

In the proposed protocols, the first element U of the group signature is used to reveal the group members who generated the signature. Element U is computed based on randomizing values z_i, depending on the document to be signed, and public keys of the group members. Values z_i are computed using the secure one-way hash function and a secret value SE known only to the group manager. Namely, each individual masking parameter zi is computed as follows $z_i = H(H(h||P_i||SE)||h||P_i)$ (or $zi = H(H(h_i||P_i||SE)||h_i||P_i)$), where SE is the additional secret key of the group manager. This formula defines the individual masking parameter for each user since it depends on their public key. The given value z_i of i-th user is different for each document because the hash value from each document is included in the argument of the specified hash function 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 since parameter z_i also depends on the secret value SE known only to group manager.

3.4 Traceability

When revealing the identity of the signers, the group manager needs to provide the proof that the identified set of signers really produced the given group signature. To do this, the group manager has to present values $H(h||P_i||SE)$, h and P_i (or $H(h_i||P_i||SE)$, h_i and P_i). Therefore, he has to keep his secret parameter SE unrevealed. The revealing of the value $H(h||P_i||SE)$ (or $H(h_i||P_i||SE)$) does not give any information to a potential malefactor since this parameter is valid only for the given document and one particular public key P_i. It can not be used to identify the group members who signed another document.

3.5 Unforgeability

The group members, who signed the document, depend on the authorization of the group manager. The computation of the group signature includes calculating the member signatures and the signature of the group manager. Therefore, only the group manager has the ability to properly authenticate and generate the group signature for the signing group.

3.6 Anonymity

The given value z_i of i-th user is different for each document because the hash value from each document is included in the argument of the specified hash-function H. Only the group manager could reveal the value of the masked public key of the given signer and document since parameter z_i also depends on secret value SE which known only by the group manager. Therefore, the proposed mechanism provides the anonymity of the signers for any person including the original signers, who verifying the group signature.

3.7 Exculpability

In the proposed schemes, no group member (or even some members join to do together) can forge the signatures of other group members. Because, the calculation of the signature value of each individual depends on not only the private key of each member (P_i), but also the random value (z_i) which calculated for each group member by the group manager. Therefore, in order to forge the signature of a group member, they need to pass the signature check equation for each member of the group manager. That means they have to break the ECDLP problem.

3.8 Unlinkability

In the proposed schemes, identifying the two different signatures generated by one member (or group of members) is impossible, except for the group manager. This is because of that the public keys of one member (or group of members) in the group have been masked by the calculation procedure of the group manager, and the verification of signatures only use the public key of the group manager. Thus, no one can identify which member (or which group of members) who signed on a document.

4 Conclusion

In this paper, the implementation of the two new group signature protocols based on usage of the multisignature scheme has been proposed to allow decreasing in the signature length and increasing in the efficiency of the signature generation procedure. The masking mechanism is used to conceal the original group members and do not influence on the size of generated signature. Moreover, we proposed a mechanism for masking public keys of the original signers. This mechanism is based on computing the hash-function value from the argument depending on the document to be signed such as the public keys of the original signers and an additional secret value of the group manager. These proposed protocols possess the features of both multisignature and group signature, therefore, they ensure higher capacity of applications in practice compared with single multisignature or group signature scheme alone.

References

1. Itakura, K., Kakamura, K.: A public-key cryptosystem suitable for digital multisignatures. NEC Res. Dev. **71**, 1–8 (1983)
2. Chaum, D., Van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) Advances in Cryptology – EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
3. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). doi:10.1007/3-540-45682-1_32

4. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). doi:10.1007/3-540-39568-7_2
5. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM. **21**, 120–126 (1978)
6. Ham, L.: Digital multisignature with distinguished signing authorities. Electron. Lett. **35**, 294–295 (1999)
7. Lin, S., Wang, B., Li, Z.: Digital multisignature on the generalized conic curve over Zn. Comp. Secur. **28**, 100–104 (2009)
8. Farashb, M.S., Biswasc, G.P., Khand, M.K., Obaidate, M.S.: A pairing-free certificateless digital multisignature scheme using elliptic curve cryptography. Int. J. Comput. Math. 1–18 (2015)
9. Kiayias, A., Yung, M.: Group signatures with efficient concurrent join. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 198–214. Springer, Heidelberg (2005). doi:10.1007/11426639_12
10. Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-based group signatures with logarithmic signature size. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 41–61. Springer, Heidelberg (2013). doi:10.1007/978-3-642-42045-0_3
11. Benoit, L., San, L., Fabrice, M., Khoa, N., Huaxiong, W.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. Cryptology ePrint Archive Report (2016)
12. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T.N., Vercauteren, F.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, Boca Raton (2005)