A Structured Multisignature Scheme from the Gap Diffie-Hellman Group

Chih-Yin Lin[1], Tzong-Chen Wu[2] and Fangguo Zhang[3]

[1]Institute of Information Management, National Chiao Tung University
1001 University Road, Hsinchu, 300, Taiwan (Email: lincy@iim.nctu.edu.tw)
[2]Department of Information Management, National Taiwan University of Science and Technology
43, Section 4, Keelung Road, Taipei, 106, Taiwan (Email: tcwu@cs.ntust.edu.tw)
[3]School of Information Technology and Computer Science, University of Wollongong
NSW 2522, Australia (Email: fangguo@uow.edu)

Abstract

In this paper, the authors propose a new structured multisignature scheme that considers the signing order among co-signers. The proposed scheme can resolve signing structures of serial, parallel, and the mix of them. Moreover, the size and the verification of a structured multisignature is the same as those of an individual signature generated by any co-signer. Arithmetically, the proposed scheme makes use of the Gap Diffie-Hellman (GDH) signature scheme recently presented by Boneh, Shacham, and Lynn. Due to the underlying GDH group, our scheme has the merits of simplicity in construction and efficiency in performance.

Keywords: cryptography, multisignature, structured multisignature, Gap Diffie-Hellman, pairing, elliptic curve.

1 Introduction

A multisignature scheme is a digital signature scheme that allows multiple signers to generate a single signature in a collaborative and simultaneous manner [3, 11]. In some applications, co-signers in a signing group may associate with different roles/ positions and therefore have different management liabilities and authorization capabilities. Thus, multisignatures generated by the same group of co-signers with different signing orders often imply different meanings. Some workflow management systems have addressed this concern in literature [5, 6], in which a multisignature has to be checked against the organizational structure of the signing group. Otherwise, the multisignature will be considered as invalid and thus invalidate the outcome of workflow process. In this paper, we refer to such multisignature scheme as the **structured multisignature** scheme. Besides the signature authenticity and message integrity, the signature verification of a structured multisignature has to further assure that the signing structure among co-signers is strictly followed.

In previously proposed structured multisignature schemes, Mitomi and Miyaji proposed two schemes that respectively based on discrete logarithm problem and integer factorization [10]. In [4], Kotzanikolaou *et al.* specified an attack against the Mitomi-Miyaji's discrete logarithm based scheme and proposed a modification. However, their modification is not secure as addressed in [9]. In [1], Burmester *et al.* proposed a scheme extended from an ElGamal-like signature scheme. Their scheme is effective but requires two rounds of structured signing to generate a valid multisignature. In [7], Lin *et al.* proposed an identity-based structured multisignature scheme. Unfortunately, Lin *et al.*'s scheme has a security flaw as specified in [8].

Recently, Boneh, Shacham and Lynn proposed a new signature scheme based on the Gap Diffie-Hellman (GDH) problem [2]. In addition, they made use of pairings on elliptic curves to realize the GDH group and design a concrete signature scheme, in which the signature size is efficiently reduced to 160 bits for equivalent security to a 320-bit DSA. In this paper, we will extend Boneh *et al.*'s GDH signature scheme to a structured multisignature scheme. The proposed scheme has the following characteristics:

i) The signing structure among co-signers can be effectively verified via multisignature verification.

ii) The size of multisignature is the same as that of any individual signature generated by a co-signer, i.e. 160 bits [2].

iii) The size of multisignature is fixed regardless of the number of co-signers.

iv) Partial structured multisignatures created during multisignature generation can be publicly verified.

v) The verification algorithms used in verifying the structured multisignature and all partial multisignatures are the same as that used to verify the individual signature generated by any co-signer.

2 The Gap Diffie-Hellman signature

Firstly, we review the definitions of the GDH problem and the GDH group. Then, we recall the GDH signature scheme proposed by Boneh *et al.* [2]. Assume *G* is a multiplicative cyclic group with order *q*, and *g* is a generator of *G*. The decisional Diffie-Hellman problem (DDH) and the computational Diffie-Hellman problem (CDH) in *G* are defined as:

Decisional Diffie-Hellman – For $a,b,c \in Z_q^*$, given $g, g^a, g^b, g^c \in G$, decide whether $c = ab$.

Computational Diffie-Hellman – For $a,b \in Z_q^*$, given $g, g^a, g^b \in G$, compute $g^{ab} \in G$.

As for the GDH problem, it is defined as that the DDH problem is easy while the CDH problem is hard. Moreover, *G* is a GDH group if the DDH problem can be easily solved while the CDH problem is computationally infeasible in *G*. As in [2], we define the four-tuple of parameters (g, g^a, g^b, g^{ab}) that satisfies the DDH problem as a valid Diffie-Hellman tuple.

Let *H* be a full-domain one-way hash function, where $H:\{0,1\}^* \to G \setminus \{1\}$. Boneh *et al.*'s GDH signature consists of three algorithms: key generation, signature generation and signature verification, stated as follows.

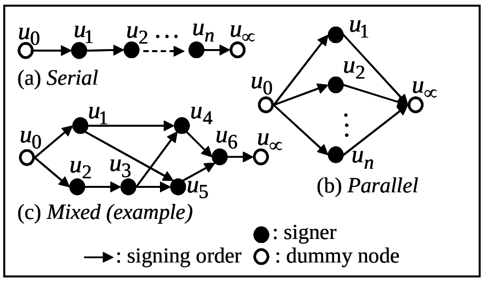
Key generation: For a signer *u*, his secret key $x \in Z_q^*$ is selected at random and his public key is computed by $v = g^x$.

Signature generation: For a message *m*, *u* generates the signature σ by computing $\sigma = H(m)^x$.

Signature verification: For a message *m* and the signature σ generated by *u*, the verifier checks if $(g, v, H(m), \sigma)$ is a valid Diffie-Hellman tuple.

3 The proposed scheme

Before addressing the details of our structured multisignature scheme, we give the notations of the signing structure as follows. Assume $Q = \{u_1, u_2, ..., u_n\}$ is a group of co-signers giving structured multisignatures. As in [7], we define the signing structure Λ for *Q* as a directed graph with all $u_i \in Q$ as real nodes and u_0 and $u_\propto$ as two dummy nodes, where u_0 denotes the starting node and $u_\propto$ the terminal node. A directed edge pointing from u_a to u_b implies u_a has to sign immediately before u_b. For example, in Figure 1, (a) is a serial signing structure, (b) is a parallel signing structure, and (c) is a mixed structure. In Figure 1(c), u_3 has to sign after u_2 and in advance to u_4 and u_5. Moreover, we denote *prev*(u_i) as the set of nodes directly precede to u_i in Λ. Thus for example in Figure 1(c), *prev*(u_1)={u_0} and *prev*(u_5)={u_1, u_3}.



(a) *Serial*

(b) *Parallel*

(c) *Mixed (example)*

Figure 1 – signing structure

The system parameters, e.g. *q*, *G*, *H*, etc., are as defined in the Gap Diffie-Hellman signature scheme. The proposed scheme consists of four algorithms: key generation, multisignature generation and multisignature verification, stated as follows.

Initialization: For each signer u_i in the system, his secret key is selected at random that $x_i \in Z_q^*$, and his public key (certified by the Certificate Authority) is computed by $y_i = g^{x_i}$, thus $y_i \in G$.

Verification key generation: Assume all $u_i \in Q$ agree upon a structured signing structure Λ, the signature verification key for each of them is generated as follows. Let $v_0 = 1$ be the multiplicative identity in *G*. Each $u_i \in Q$ generates his/ her individual signature verification key v_i in accordance with Λ as:

$v_i = (g \cdot \prod_{u_j \in prev(u_i)} v_j)^{x_i}$.

Finally, the verification key for *Q* is $v_Q = \prod_{u_j \in prev(u_\propto)} v_j$. Anyone can verify the authenticity of v_i by checking if $(g, y_i, g \cdot \prod_{u_j \in prev(u_i)} v_j, v_i)$ is a valid Diffie-Hellman tuple.

Multisignature generation: For a message *m* to be signed, each $u_i \in Q$ performs the following steps in accordance with Λ:

Step 1. Compute $M = H(m)$.

Step 2. Verify σ_j from precedent signer u_j, for all $u_j \in prev(u_i)$, by checking if (g, v_j, M, σ_j) is a valid Diffie-Hellman tuple.

Step 3. Compute $\sigma_i = (M \cdot \prod_{u_j \in prev(u_i)} \sigma_j)^{x_i}$, where $\sigma_0 = 1$.

Finally, $\sigma_Q = \prod_{u_j \in prev(u_\propto)} \sigma_j$ serves as the structured multisignature generated by all $u_i \in Q$.

Multisignature verification: For a message *m* and the structured multisignature σ_Q generated by all $u_i \in Q$ in accordance with Λ, the verifier checks if $(g, v_Q, H(m), \sigma_Q)$ is a valid Diffie-Hellman tuple.

4 Discussions

For the structured multisignature scheme, there are several possible attacks as proposed in [1] and [9], and they can be divided into **outsider attack** and **insider attack**. The outsider attack means that an adversary is not a signer in the group $Q = \{u_1, u_2, ..., u_n\}$ and he/she wants to forge a multisignature for some message with any signing structure Λ. We consider the insider attack into two cases: (1), Some malicious signers want to forge a multisignature for some message with any signing structure Λ; (2), Some malicious signers want to change the signing structure Λ for a message. In the followings, we will show that our new structured multisignature scheme is secure against these attacks.

If there is an adversary (not in the signing group *Q*) who wants to forge the multisignature of the message *m* for the signing structure Λ, in other words, the adversary knows (g, v_Q) and wants to find σ_Q, such that $(g, v_Q, H(m), \sigma_Q)$ is a valid Diffie-Hellman tuple. In fact, this is equal to forge Boneh *et al.*'s GDH signature scheme with the public key v_Q. Therefore, based on that Boneh *et al.*'s signature scheme is proven to be secure against existential forgery on adaptively chosen message under the hardness assumption of CDH problem in *G* and the random oracle model, the outsider attack is infeasible.

For the insider attack, once all co-signers collude, they can generate a structured multisignature via a completely reverse order to the predefined signing structure. However, it is reasonable that once all co-signers collude, the actual signing order will no longer a restriction to any individual co-signers. We assume that at least one of the co-signers in the signing group is honest. For the first case of insider attack that some malicious signers may try to forge a multisignature for a message m' with signing structure Λ, we assume that they have had honest signers' some partial signatures of some messages. Then, with these partial signatures and honest signers' public keys, these malicious signers may try to obtain the forged signatures of message m' or to derive honest signers' secret keys. However, deducing the secret key x_i from the public key $y_i = g^{x_i}$ requires the ability to solve the discrete logarithm problem in *G*. On the other hand, since each signer uses Boneh *et al.*'s GDH signature scheme to generate the partial signature, and as stated that Boneh *et al.*'s GDH signature scheme is non-forgeable, this attack is infeasible.

Now we consider another case of insider attack: Some malicious signers have had a structured multisignature of a message *m* with the signing structure Λ, but they want to change the signing structure from Λ to Λ', which has never appeared before. To conduct such attempt, these malicious signers have to construct this new signing structure Λ' firstly. Since a valid signing structure can only be constructed by the cooperation of all $u_i \in Q$ that all of them must agree upon to the structure in advance, we can declare that this attack is not workable. Note that the signing structure consists of the signature verification keys of all signers, the signature verification key for each of signers is generated using their secret key, and each individual signature verification key can be verified. Thus, we can assume that these malicious signers have had another signing structure Λ' and they want to obtain the partial signature of an honest signer for messages *m*. However, this attack is equal to forge Boneh *et al.*'s GDH signature scheme, so, we say that this attack is infeasible, too.

Next, we consider the efficiency of the proposed structured multisignature scheme. As in Boneh *et al.*'s GDH signature scheme, the proposed scheme can be realized by pairings on elliptic curves. Thus, the signature verification that checks the validity of the Diffie-Hellman tuple $(g, v, H(m), \sigma)$ can be efficiently achieved by testing if $\hat{e}(g,\sigma) = \hat{e}(v, H(m))$, where $\hat{e}$ is a modified Weil pairing as defined in [2]. Moreover, like Boneh *et al*.s scheme [2], the signature size of our multisignature can be efficiently reduced to very short, e.g. 160 bits for equivalent security to a 320-bit DSA. Thanks to the underlying GDH signature scheme it turns out that our proposed structured multisignature scheme is simpler and more efficient than existed schemes [1].

5 Conclusions

In this paper, we have proposed a new structured multisignature scheme from the Gap Diffie-Hellman group. Besides achieving signature authenticity and message integrity as in ordinary signature schemes, the proposed scheme can further assure that the predefined signing structure has been strictly followed in the signature verification process. We have shown that the proposed scheme is secure against the insider attack as well as the outsider attack. Due to the underlying GDH group, our scheme has the merits of simplicity in construction and efficiency in key size and in computational cost.

References

[1] M. Burmester, Y. Desmedt, H. Doi, M. Mambo, E. Okamoto, M. Tada and Y. Yoshifuji, "A structured ElGamal-type multisignature scheme", *Proceedings of Third International Workshop on Practice and Theory in Public Key Cryptosystems* (*PKC 2000*), Springer-Verlag, 2000, pp. 466-483.

[2] D. Boneh, H. Shacham and B Lynn, "Short signatures from the Weil pairing", *Advances in Cryptology – AISACRYPT 2001*, Springer-Verlag, 2001, pp. 514-532.

[3] K. Itakura and K. Nakamura, “A public-key cryptosystem suitable for digital multisignature”, *NEC Research and Development*, Vol. 71, October 1983, pp. 1-8.

[4] P. Kotzanikolaou, M. Burmester and V. Chrissikopoulos, “Dynamic multi-signatures for secure autonomous agents”, *Proceedings 12th International Workshop on Database and Expert Systems Applications* (*DEXA 2001*), IEEE Computer Society, 2001, pp. 587–591.

[5] K.R.P.H. Leung and L.C.K. Hui, “Signature management in workflow systems”, *Proceedings of the 23rd Annual International Computer Software and Applications Conference* (*COMPSAC’99*), IEEE, 1999, pp. 424-429.

[6] K.R.P.H. Leung and L.C.K. Hui, “Handling signature purposes in workflow systems”, *The Journal of Systems and Software*, Vol. 55, 2001, pp. 245-259.

[7] C.-Y. Lin, T.-C. Wu and J.-J. Hwang, “ID-based Structured Multisignature Schemes”, *Advances in Network and Distributed Systems Security*, Kluwer Academic Publishers (IFIP Conference Proceedings 206), Boston, 2001, pp. 45-59.

[8] C. J. Mitchell, “An attack on an ID-based multisignature scheme”, Royal Holloway, University of London, Mathematics Department Technical Report RHUL-MA-2001-9, December 2001.

[9] C.J. Mitchell and N. Hur, “On the security of a structural proven signer ordering multisignature scheme”, in: B. Jerman-Blazic and T. Klobucar (eds.), *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security* (*CMS 2002*), Kluwer Academic Publishers (IFIP Conference Proceedings 228), Boston, 2002, pp.1-8.

[10] S. Mitomi and A. Miyaji, “A Multisignature Scheme with Message Flexibility, Order Flexibility and Order Verifiability”, *Proceedings of the 5th Australasian Conference on Information Security and Privacy* (*ACISP 2000*), Spring-Verlag, 2000, pp. 298-312.

[11] S. Micali, K. Ohta and L. Reyzin, “Accountable-subgroup multisignatures: extended abstract”, *Proceedings of the ACM Conference on Computer and Communications Security 2001* (*CCS 2001*), ACM press, 2001, pp. 245-254.