Int. J. Contemp. Math. Sciences, Vol. 3, 2008, no. 1, 37 - 42

Dynamic Multi-Secret Sharing Scheme

Han-Yu Lin* and Yi-Shiung Yeh

Department of Computer Science
National Chiao Tung University
Hsinchu, 300, Taiwan, Republic of China
* e-mail: hanyu.cs94g@nctu.edu.tw

Abstract

Secret sharing schemes are very important techniques for the key management. To provide more efficient and flexible alternatives for the applications of secret sharing, this paper presents a dynamic multi-secret sharing scheme. A significant characteristic of the proposed scheme is that each participant has to keep only one master secret share which can be used to reconstruct different group secrets according to the number of threshold values. By applying successive one-way hash functions and the exclusive OR (XOR) operation, the proposed scheme is secure against the notorious conspiracy attack even though the pseudo secret shares are compromised. Further, when one of the group secrets is updated with a new one, each participant's master secret share is still unchanged, i.e., these master secret shares are truly multi-use instead of one-time-use.

Keywords: dynamic, secret sharing, key management, Lagrange Interpolation, threshold

1. Introduction

Since Diffie and Hellman [2] introduced the first public key system based on the intractability of solving the discrete logarithm problem (DLP) [2, 9] in 1976, public key cryptosystems [3, 12] had been widely used in various applications such as e-cash [1], e-voting [11] and e-market [8], etc. One important issue of the public key system is the key management. Consider the operations of enterprise organizations in reality that lots of confidential documents are usually kept in a safe deposit. The management of the secret key to the safe deposit thus becomes crucial. Someone who is responsible for keeping the secret key has to be a trusted one. Yet, the case that the secret key is lost still might occur. A better alternative would be reducing the risk of key loss by distributing the responsibility of key management among many persons, i.e., the secret key is divided into several pieces which are then sent to different individuals. If sufficient participants are willing to cooperate with each other, they can reconstruct the original secret key by offering their key shares.

In 1979, Shamir [13] put the idea into practice and proposed a (t, n) threshold

secret sharing scheme in which the master secret was divided into n secret shares and were delivered to different users. Any t or more of the n users could cooperatively reconstruct the master secret while less than or equal to $t - 1$ could not. However, Shamir's scheme could not share multiple secrets and once the master secret was updated with a new one, the system had to reissue renewed secret shares to each user, which was considered to be system resources consuming and impracticable. To eliminate the weaknesses, in 1994, He and Dawson [5] proposed a multistage secret sharing scheme based on the one-way function. By applying successive one-way hash functions, the He-Dawson scheme realized the notion of multi-secret sharing. Yet, in 2007, Geng *et al*. [4] pointed out that the He-Dawson scheme was actually the one-time-use scheme [6] and further proposed a new multi-secret sharing scheme with multi-policy. Preserving the merit of Geng *et al*.'s scheme that different group secrets are reconstructed according to the number of threshold values, this paper presents a dynamic multi-secret sharing scheme which also outperforms Geng *et al*.'s scheme in terms of the computation complexity.

2. Dynamic Multi-Secret Sharing Scheme

This section introduces the dynamic multi-secret sharing scheme. The proposed scheme can be divided into three stages: the system initialization, the pseudo secret share generation, and the group secret reconstruction stages.

***The system initialization stage*:**

Let SA be the system authority (SA) who is responsible for initializing the following public parameters:

- p : a large prime;
- g : a primitive element over GF(p);
- $h(\cdot)$: a secure one-way hash function which accepts input of any length and generates a fixed length output;
- ID_j : the identifier with respect to the user U_j, for $j = 1, 2, \ldots, n$;

***The pseudo secret share generation stage*:**

Assume the SA wants to share k group secrets s_i, for $i = 1, 2, \ldots, k$, among n users. The SA can perform the following steps to generate pseudo secret shares and distribute master secret shares.

Step 1 Choose distinct $x_j \in_R Z_p^*$, for $j = 1, 2, \ldots, n$, as the master secret shares;

Step 2 Construct a polynomial $f_i(x)$ of degree $(i - 1)$, for $i = 1, 2, \ldots, k$, as

$$f_i(x) = s_i + d_1 x + \ldots + d_{i-1} x^{i-1} \text{ where } f_i(0) = s_i; \quad (1)$$

Step 3 For $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n$, compute

$$V_{ij} = f_i(ID_j), \quad (2)$$

$$c_{ij} = h^i(x_j) \oplus x_j, \tag{3}$$

$$R_{ij} = V_{ij} - c_{ij} \bmod p; \tag{4}$$

Here, c_{ij}'s are the pseudo secret shares, the symbol '⊕' denotes the exclusive OR (XOR) operation and $h^i(x_j)$ denotes i successive applications of h to x_j.

Step 4 Deliver the master secret share x_j , for j = 1, 2, …, n, to each user U_j via a secure channel and publish all R_{ij}'s;

***The group secret reconstruction stage*:**

To reconstruct the lth group secret, say s_l, at least l participants or more of n users must cooperatively perform the following steps with the group secret combiner:

Step 1 Each U_j, for j = 1, 2, …, l, computes his pseudo secret share as

$$c_{lj} = h^l(x_j) \oplus x_j, \tag{5}$$

and then sends it to the group secret combiner;

Step 2 Upon receiving all c_{lj}'s, for j = 1, 2, …, l, the group secret combiner reconstructs the lth group secret as

$$s_l = \sum_{j=1}^{l} (c_{lj} + R_{lj}) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r} \bmod p\,. \tag{6}$$

The correctness of Eq. (6) can be assured as the proof of Theorem 1.

Theorem 1. After receiving at least l pseudo secret shares, the group secret combiner can reconstruct the lth group secret, s_l, by Eq. (6).

Proof: From the right-hand side of Eq. (6), we have

$$\sum_{j=1}^{l} (c_{lj} + R_{lj}) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r}$$

$$= \sum_{j=1}^{l} (c_{lj} + V_{lj} - (h^l(x_j) \oplus x_j)) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r} \quad \text{(by Eqs. (3) and (4))}$$

$$= \sum_{j=1}^{l} ((h^l(x_j) \oplus x_j) + V_{lj} - (h^l(x_j) \oplus x_j)) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r} \quad \text{(by Eq. (3))}$$

$$= \sum_{j=1}^{l} (V_{lj}) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r}$$

$$= \sum_{j=1}^{l} f_l(ID_j) \prod_{r=1, r\neq j}^{l} \frac{-ID_r}{ID_j - ID_r} \quad \text{(by Eq. (2))}$$

$= s_l \pmod{p}$ (by Lagrange Interpolation [14])

which equals to the left-hand side of Eq. (6).

Q.E.D.

3. Security Considerations and Performance Evaluation

In this section, we will discuss some security considerations of the proposed scheme followed by the performance evaluation.

3.1 Security Considerations

The security assumptions of our proposed scheme are the one-way hash function (OHF) [2, 9] and the XOR operation. The definitions of OHF are briefly restated below: Let h be an OHF. It is computationally infeasible to derive m from $h(m)$. In addition, finding a pair (m, m') which satisfies $h(m) = h(m')$ is also infeasible. In the following, we analyze some security considerations of the proposed scheme.

(1). The confidentiality of pseudo secret shares: To derive a user's pseudo secret share c_{ij} from Eq. (4), an attacker may first obtain the corresponding public information R_{ij}. However, the pseudo secret share c_{ij} is protected by the integer V_{ij} which can only be computed from the secret polynomial $f_i(ID_j)$. Hence, the attacker will fail to make it. On the contrary, if the attacker attempts to directly derive c_{ij} from Eq. (3), he has to know the master secret share x_j first, which is also computationally infeasible.

(2). The confidentiality of master secret shares: The master secret share x_j is randomly chosen by the SA and then delivered to each user U_j via a secure channel. Even though the pseudo secret share c_{ij} is compromised, any malicious adversary can not successfully derive x_j from Eq. (3) under the protection of the OHF and the XOR operation.

(3). The confidentiality of group secrets: Consider the notorious conspiracy attack that $l - 1$ malicious insiders cooperatively attempt to reconstruct the lth group secret s_l. Unfortunately, according to the analysis of the confidentiality of pseudo secret shares, they can not obtain sufficient valid c_{lj}'s to reconstruct the group secret s_l.by Eq. (6). It is also computationally infeasible to compute c_{lj} from c_{l-1j} under the protection of the XOR operation.

From the above discussions, it can be seen that our proposed scheme is secure based on the hardness of XOR and OHF assumptions.

3.2 Performance Evaluation

In the subsection, we compare the proposed scheme (LY for short) with

Geng *et al*.'s scheme (GFH for short) [4] in terms of the computation complexity. For facilitating the comparisons, we first define the following notations:

T_h: the time for performing a one-way hash function h
T_m: the time for performing a modular multiplication computation
T_e: the time for performing a modular exponentiation computation

The time for performing the modular addition and the exclusive OR (XOR) operation is ignored because they are negligible as compared to the others. Mitchell *et al*. [10] also stated that a hash function would not take longer time than that of a modular multiplication computation. Consequently, we can use one T_m to approximate one T_h without affecting the correctness in the evaluation. For ease of comparisons, the above various computations can be utilized into the same unit of modular multiplication computation [7, 10]. The detailed comparisons are listed as Table 1. In parentheses, the rough estimation in terms of T_m is given. As the result shown in Table 1, it can be seen that the proposed scheme outperforms Geng *et al*.'s scheme for the entire protocol.

Table 1. Comparisons of the computation complexity

Stages	Scheme	Time complexity	Rough Estimation*
System initialization	GFH	0	
	LY		
Pseudo secret share generation	GFH	$k(2n + 1)T_e + knT_h$	$k(481n + 240)T_m$
	LY	knT_h	knT_m
Group secret reconstruction	GFH	$2kT_e + (k^2 - k)T_m + k^2T_h$	$(2k^2 + 479k)T_m$
	LY	$(k^2 - 2k)T_m + k^2T_h$	$(2k^2 - 2k)T_m$

Remark: * $T_h \approx T_m$, $T_i \approx 3T_m$, and $T_e \approx 240T_m$.

4. Conclusions

In this paper, we have proposed a dynamic multi-secret sharing scheme based on the one-way hash function. The major characteristics of its design are multi-use of the master secret shares and that different group secrets can be reconstructed according to the number of threshold values, which provides more flexibility. By applying successive one-way hash functions and the XOR operation, the proposed scheme is secure against notorious conspiracy attacks even though the pseudo secret shares are compromised. Besides, our scheme also outperforms recently proposed Geng *et al*.'s scheme in terms of the computation complexity.

References

[1] S. Brands, Untraceable off-line cash in wallet with observers, Advances in Cryptology – CRYPTO'93, Springer-Verlag, 1993, pp. 302-318.
[2] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (6) (1976) 644-654.
[3] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory IT-31 (4) (1985) 469-472.
[4] Y.J. Geng, X.H. Fan, F. Hong, A new multi-secret sharing scheme with multi-policy, The 9th International Conference on Advanced Communication Technology, Vol. 3, 2007, pp. 1515-1517.
[5] J. He, E. Dawson, Multistage secret sharing based on one-way function, Electronics Letters, 30 (19) (1994) 1591-1592.
[6] W.A. Jackson, K. M. Martin, C. M. O'Keefe, On sharing many secrets, Advances in Cryptology – ASIACRYPT'94, Springer-Verlag, 1994, pp. 42-54.
[7] N. Koblitz, A. Menezes, S. Vanstone, The state of elliptic curve cryptography, Designs, Codes and Cryptography 19 (2-3) (2000) 173-193.
[8] Y. Li, B. Huang, W. Liu, H. Gou, C. Wu, An electronic market architecture for virtual enterprises, 2001 IEEE International Conference on Systems, Man, and Cybernetics, 2001, Vol. 3, pp. 2028-2033.
[9] A. Menezes, P. Oorschot, S. Vanstone, Handbook of applied cryptography, CRC Press, Inc., 1997.
[10] C.J. Mitchell, F. Piper, P. Wild, Digital signature, In: Simmons, G.J. (Ed.), Contemporary Cryptology: The Science of Information Integrity, IEEE Press, 1992, pp. 325-378.
[11] I. Ray, N. Narasimhamurthi, An anonymous electronic voting protocol for voting over the Internet, Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, 2001, pp. 188-190.
[12] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (2) (1978) 120-126.
[13] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612-613.
[14] B. Wendroff, Theoretical Numerical Analysis, Academic Press Inc., 1996.

Received: July 29, 2007