Novel verifiable multi-secret sharing scheme using bilinear pairing

Hung-Yu Chien

Department of Information Management, National Chi Nan University

Abstract

Secret sharing schemes as basic building blocks for many applications in distributed systems should be efficient and robust to any possible attacks. Based on bilinear pairing, this paper will propose a new (*t, n*) threshold verifiable multi-secret sharing scheme and a GGOC verifiable multi-secret sharing scheme that improve the security and minimizes the number of published values.

Keywords: cryptography, security, bilinear pairing, cheating, secret sharing, threshold cryptosystem.

1. Introduction

Many distributed systems (for examples, secure database access and distributed signatures) adopt some kinds of secret sharing schemes to achieve better availability and robust security [20]. Therefore, the secret sharing schemes as building blocks of the distributed systems should be efficient and robust to any possible attacks. One of the important secret sharing schemes is the (t,n) threshold secret sharing scheme that was firstly proposed by Shamir [11] and Blakley [1] independently. In a (t,n) threshold secret sharing scheme, the *dealer* or the *secret distributor* (*SD*) first divides the secret into *n* pieces (called the secret *shadows*) and secretly distributes these shadows to the *n* participants such that the secret can be reconstructed using the

co-operation of t or more members, while the secret cannot be reconstructed if only $t-1$ or fewer members are willing to co-operate. During the re-construction of the secrets, the members of the group use their shadows to compute the so-called *sub-shadows* to the other members such that the group can re-construct the them using the *sub-shadows*.

If there are multiple secrets to be shared among the participants, then the secret sharing scheme is called a multi-secret sharing scheme; otherwise, it is just called a secret sharing scheme. In a multi-secret sharing scheme, there are multiple secrets to be distributed during one secret sharing process [22]. This generalization has several applications [22]. One is to protect several secrets with the same amount of data usually needed to protect one secret. Two is to partition one large secret into l pieces with these pieces protected with a smaller amount of data than is needed to protect the entire secret. Regarding multi-secret sharing schemes, Jackson et al. [22] further had their classification- the one-time-use scheme and the multi-use scheme. In a one-time-use scheme, the *SD* must redistribute fresh shadows to each participant once some particular secrets have been reconstructed. On the other hand, in a multi-use scheme, the shadows owned by one participant still remain secret to others, even after multiple secret-reconstruction operations have been performed. The *SD*, therefore, does not need to redistribute fresh shares. To redistribute shares is a very costly process with respect to both time and resources. Since it is costly to distribute the shadows, it is more desirable to have the multi-use, multi-secret sharing schemes. The schemes [3, 7-9, 21-22] are multi-use schemes, while the schemes [1, 4, 5, 6, 10, 11, 13-16, 20, 23-26] are one-time-use schemes. Based on the linear block code [27], Bertilsson and Ingemar [28], and Karnin et al. [29] independently proposed their secret sharing schemes. However, the construction of the generator matrix is complicated and inefficient. Among the referenced schemes, Chien et al.'s scheme [21]

is very promising because it is efficient, multi-use, and multi-secret sharing, and is able to provide a unified approach to various secret-sharing problems (threshold secret sharing, weighted threshold secret sharing, dynamic weighted threshold secret sharing and GGOC). However, the cheating detection is not considered in the scheme [21].

In a secret sharing process, any dishonest entity (the dealer or the participants) might deviate from the protocol for its own benefits. Therefore, a secret sharing scheme should be able to detect any possible cheating [23-25], in addition to its efficiency. A verifiable secret sharing scheme (VSS) [4] is a secret sharing scheme that allows the participants detecting the cheating of the dealers or the cheating of other participants. However, all of these cheater identification schemes [4, 12, 14-16] demand large numbers of public values.

Regarding the cheating detection property, some schemes allow the participants detecting whether the dealer is cheating and whether his/her shadow is valid [4, 12, 14-16], while some schemes allow the participants detecting whether other participants are honest or not when they provide the sub-shadows (that is computed from their private shadows) for re-constructing the secrets [12, 14-16]. Chor et al.'s [4] verifiable secret sharing scheme (VSS) can detect the cheating of the dealer, and Stadler's [12] VSS scheme can further detect the cheating of the participants. However, the schemes [4, 12, 14-16] can only deal with one secret in one secret sharing process, and require publishing too many values for cheating detection. Harn's scheme [5] and Chen et al.'s scheme [2] can deal with multiple secrets. However, Lin and Wu [10] had pointed out the weaknesses of Harn's scheme and Chen et al.'s scheme as follows. In Harn scheme, (1) it requires heavy computations ($O(n!/(n-t)!\cdot t!)$) to verify one's shadows, (2) new secrets cannot be dynamically added by the dealer, and (3) it needs the costly interactive verification protocol to

verify other participants' sub-shadows. In Chen et al.'s scheme, the dealer needs to record all participants' shadows and compute an *n*-dimensional verification vector for each shared secret. Lin and Wu, therefore, proposed a (*t*, *n*) threshold *verifiable multi-secret sharing* (VMSS) scheme, but He and Wu [7] showed that a malicious participant could submit a fake sub-shadow to cheat other honest participants. Recently, Chang, Hwang, and Yang [19] proposed a new (*t*, *n*) threshold scheme to overcome all the above weaknesses. The scheme has the merits: (1) the dealer can dynamically add new secrets to be shared, (2) each participant's shadow is re-usable, (3) the number of published values for detecting cheating is greatly reduced, and (4) each participant can verify his shadow and detect the cheating of other participants using non-interactive process. They also claimed that even a participant who has derived some secrets still needs other (*t*-1) participants' co-operation to recover the remaining secrets. The Chang-Hwang-Yang scheme is called the CHY scheme and the Lin-Wu scheme is call the LW scheme for short in this paper.

However, we find that the CHY scheme still owns several weaknesses: (1) a participant who has derived some secrets could recover the remaining secrets with high probability without any co-operation of other participants, (2) an outside adversary can derive the un-disclosed secrets with very high probability if he happened to know some disclosed secrets, and (3) it still needs many public values for detecting cheating. This paper will show the weaknesses. After that, we shall propose a new (*t*, *n*) threshold verifiable multi-use, multi-secret sharing (VMSS) scheme that enables the participants to verify his shadow from the dealer and the sub-shadows from other participants during the secret re-construction process. This new scheme is based on the bilinear pairing, and is efficient and robust to all the possible attacks. The scheme owns all the mentioned merits and minimizes the number of published values.

In addition, our scheme can be extended to to the *generalized group-oriented*

cryptosystem (*GGOC*), but both the CHY scheme and the LW scheme cannot. The *generalized group-oriented cryptosystem* (*GGOC*) [23-26] is another important secret sharing problem of which the sharing policies could not be expressed in a threshold-based style [25]. In *GGOC* , the sharing policy is expressed by dividing the group of users into the qualified subsets and the unqualified subsets. Only through the co-operation of members of the qualified subset can a secret be reconstructed. In *GGOC* , the qualified subsets of the users can be determined using an *access structure* Γ_0 that consists of those *minimal authorized subgroups* [25]. This *access structure* Γ_0 is usually denoted in disjunctive normal form (*DNF*). For example, $\Gamma_0 = User_1User_2 + User_3User_4$, where $User_1User_2$ and $User_3User_4$ are the *minimal authorized subgroups*. This means that those groups which consist of at least $User_1$ and $User_2$ are all qualified subsets. So are those groups consisting of at least $User_3$ and $User_4$.

The rest of this paper is organized as follows. To demonstrate the weaknesses of the recent research, Section 2 reviews the CHY scheme and shows the weaknesses. Section 3.1 proposes the new (*t, n*) VMSS scheme, and Section 3.2 extends the scheme to our GGOC VMSS scheme. The security analysis and performance evaluation are provided in Section 4. Finally, Section 5 states our conclusions.

2. Comments on the CHY scheme

Even though the CHY scheme has improved many weaknesses of its predecessors, it still incurs several security weaknesses. We, therefore, review the scheme and examine the weaknesses as follows.

2.1 Review of the CHY scheme

The CHY scheme consists of four phases - (1) the initialization phase, (2) the shadow generation and verification phase, (3) the credit ticket generation phase, and (4) the sub-shadow verification and secret reconstruction phase.

(1) Initialization

There is one notice board (NB) that could be read by every participant but be modified by the dealer only. The dealer first chooses an RSA modulus $N = p \cdot q$ and a generator g, where $p = 2p'+1$ and $q = 2q'+1$ are two large primes, g is a generator for a subgroup of Z_N^* with order $R = p' \cdot q'$. e and d separately denote the public and private key in the RSA algorithm, where $e \cdot d = 1 \bmod \phi(N)$. The dealer finally publishes $\{N, e, g\}$ on the NB and keeps $\{R, d\}$ secret.

(2) Shadow generation and verification

Let $S = \{S_1, S_2, ..., S_m\}$ be a set of m secrets to be distributed among $G = \{U_1, ..., U_n\}$ - a group of n participants, where each U_i has his identity ID_i. The dealer performs the following steps to distribute the shadows.

(2.1) Randomly generate a polynomial $f(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1} \bmod R$, where each $a_k \in_R Z_R$, compute and publish a check vector $V = [V_0, ..., V_{k-1}]$ for each coefficient a_k as $V_k = g^{a_k} \bmod N$ for k=0~(t-1).

(2.2) Compute a secret shadow $x_i = f(ID_i) \cdot p_i^{-1} \bmod R$, where $p_i = \prod_{U_k \in G, U_k \neq U_i} (ID_i - ID_k) \bmod R$ and publish the value $y_i = g^{x_i} \bmod N$ as U_i's public key on the NB.

(2.3) Distribute $\{ g^{p_i} \bmod N, x_i \}$ secretly to U_i. U_i can verify his shadow by

checking whether $(g^{p_i})^{x_i} \stackrel{?}{=} \prod_{k=0}^{t-1} (V_k)^{(ID_i)^k} \bmod N$ holds. If the equation does not hold, then the shadow x_i is not valid.

(3) Credit ticket generation

The dealer performs the following steps to compute the *m* credit tickets $C_1,...,C_m$ for each secret $S_1,...,S_m \in S$.

(3.1) Randomly choose *m* distinct integers $r_1,...,r_m \in Z_R$ for each secret $S_1,...,S_m \in S$.

(3.2) Compute the credit ticket $C_j = g^{r_j \cdot d} \bmod N$ and the value $h_j = (g^{a_0 \cdot r_j \cdot d} \bmod N) \oplus S_j$ for *j*=1~*m*. The dealer puts the 3-tuple $\{r_j, C_j, h_j\}$ on the NB.

In addition, if the dealer wants to add a new secret S_{new} for sharing, he/she only needs to generate a new 3-tuple $\{r_{new}, C_{new}, h_{new}\}$ for S_{new} and put them on the NB.

(4) Sub-shadow verification and secret re-construction

Let *W* (|*W*|=*t*) be any subset of *t* participants in *G* want to co-operate to reconstruct a secret S_j. Without lose of generality, we may assume *W*=$\{U_1,...,U_t\}$. Each $U_i \in W$ obtains the 3-tuple $\{r_j, C_j, h_j\}$ from the NB and uses his/her secret shadow x_i to compute a sub-shadow $A_{ij} = C_j^{x_i} \bmod N$. Then, U_i releases A_{ij} to the other co-operators in *W*. Any co-operator can verify the validity of U_i's sub-shadow A_{ij} by checking whether the equation $A_{ij}^{e} \stackrel{?}{=} (y_i)^{r_j} \bmod N$ holds. If the equation does not hold, the verifier announces that U_i is cheating and stops

the co-operation.

If all the A_{ij} s released by the *t* participants are valid, every participant in *W* can reconstruct the secret S_j as follows.

$S_j = h_j \oplus (\prod_{U_i \in W} (A_{ij})^{\Delta_i} \bmod N)$, where $\Delta_i = (\prod_{U_k \in W, U_k \neq U_i} -ID_k) \cdot (\prod_{U_k \in G, U_k \notin W} (ID_i - ID_k))$

Then, all the secrets $S_1, ..., S_m \in S$ can be recovered by performing this phase repetitively.

2.2 Weaknesses of the CHY scheme

We now show the weaknesses of the CHY scheme.

(1) A malicious participant who has recovered some secrets with other *t*-1 participants can alone recover the remaining secrets with high probability

Chang et al. claimed that a malicious participant U_i who has obtained some previously recovered secrets can not reveal any remaining secret in *S* without the co-operation of other (*t*-1) co-operators. That is, even if U_i has recovered the secrets $S_a \in S$ and $S_b \in S$ with the other (*t*-1) co-operators and can derive the values $g^{a_0 \cdot r_a \cdot d} \bmod N$ and $g^{a_0 \cdot r_b \cdot d} \bmod N$, he still cannot derive the values $g^{a_0 \cdot d} \bmod N$ and t $(g^{a_0 \cdot d})^{r_c} \bmod N$ and then S_c for some remaining secret S_c.

However, we find that the scheme fails to commit this security requirement. Assume U_i be the malicious participant who has reconstructed some previous secrets ($S_a \in S$, $S_b \in S$) and so the values $g^{a_0 \cdot r_a \cdot d} \bmod N$ and $g^{a_0 \cdot r_b \cdot d} \bmod N$, with the other (*t*-1) co-operators. With the public values (r_a, r_b), we have the following two cases

Case 1. $\gcd(r_a, r_b) = 1$. For such a case, U_i can derive two values *x* and *y* using the

extended Euclidean algorithm such that $x \cdot r_a + y \cdot r_b = 1$. Using the two values *x* and *y*, U_i computes the value $(g^{a_0 \cdot r_a \cdot d})^x \cdot (g^{a_0 \cdot r_b \cdot d})^y = g^{a_0 \cdot d \cdot (x \cdot r_a + y \cdot r_b)} = g^{a_0 \cdot d} \bmod N$. Now U_i can alone compute any remaining secret $S_c = h_c \oplus ((g^{a_0 \cdot d})^{r_c} \bmod N)$, where h_c and r_c are public values.

Case 2. $\gcd(r_a, r_b) = k$ and $k \neq 1$. For such cases, U_i can derive two values *x* and *y* using the extended Euclidean algorithm such that $x \cdot r_a + y \cdot r_b = k$. Using the two values *x* and *y*, U_i computes the value $(g^{a_0 \cdot r_a \cdot d})^x \cdot (g^{a_0 \cdot r_b \cdot d})^y = g^{a_0 \cdot d \cdot (x \cdot r_a + y \cdot r_b)} = g^{a_0 \cdot d \cdot k} \bmod N$. It is highly probable that the *k* is a small integer and $k \mid r_c$ for some remaining secret S_c. For such a secret S_c, U_i can alone compute the secret $S_c = h_c \oplus ((g^{a_0 \cdot d \cdot k})^{\frac{r_c}{k}} \bmod N)$, where the exponent $\frac{r_c}{k} \bmod R$ is computable since $k \mid r_c$.

From the analysis of case 1 and case 2, we conclude that a malicious participant can recover some remaining secrets alone with a high probability when he had ever co-operated with other *t*-1 participants before. The probability would get higher as the number of disclosed secrets increases, since the probability for $gcd(r_a, r_b)=k$ and *k* is a small integer would get higher as the number of disclosed secrets increases.

(2) **A outside adversary who happened to know some disclosed secrets can alone recover the remaining secrets with high probability**

A secret sharing scheme should prevent an adversary from getting any un-disclosed secrets even he has got some secrets for any reasons: this might happen if some secrets were compromised or if some obsolete secrets were disclosed. Assume an adversary *E* who happened to know some secrets $S_a \in S$ and $S_b \in S$ in the CHY scheme. Now *E* can compute $(g^{a_0 \cdot d \cdot r_a} \bmod N) = h_a \oplus S_a$ and

$(g^{a_0 \cdot d \cdot r_b} \bmod N) = h_b \oplus S_b$. As the analysis in the previous paragraph, E can derive $g^{a_0 \cdot d \cdot k} \bmod N$ using the published values r_a and r_b with high probability, and can derive those secrets S_c where the r_c is divisible by k.

3. New schemes using bilinear pairings

3.1 New *(t, n) VMSS*

Now we propose a new (*t*, *n*) threshold multi-secret sharing scheme, based on the bilinear pairing (the bilinear map). Before introducing the scheme, we first introduce the group and the bilinear map.

Definition 1 (Non-degenerate, bilinear, computable map) [18]**.** Let G_1 and G_2 be cyclic groups of prime order *q*, where G_1 is additive and G_2 is multiplicative. Let $\hat{e}$: G_1 *x* G_1 → G_2 be a map with the following properties below.

(1) Non-degenerate: There exists $X,Y \in G_1$ such that $\hat{e}(X,Y) \neq 1$.

(2) Bilinear: $\hat{e}(X_1+X_2, Y) = \hat{e}(X_1,Y) \cdot \hat{e}(X_2,Y)$ and $\hat{e}(X,Y_1+Y_2) = \hat{e}(X, Y_1) \cdot \hat{e}(X, Y_2)$.

(3) Computable: There is an efficient algorithm for evaluating $\hat{e}$.

Group with such a map could be derived from subgroups of elliptic curve groups $E(F_p)$, whose order *q* divides $p^k - 1$ but does not divide $p^i - 1$ for $0 < i < k$, and *q* is large enough to resist the MOV attack [17], but small enough to make the Tate or Weiling pairing feasible.

Now we are ready to present our scheme. The scheme also consists of four phases.

(1) Initialization

There is one notice board (NB) that could be read by every participant but be

modified by the dealer only. The dealer selects finite Abelian groups G_1 and G_2 of prime order q such that there is a non-degenerate, bilinear, computable map $\hat{e}: G_1 \times G_1 \rightarrow G_2$. The group G_1 is derived from subgroups of elliptic curve groups $E(F_p)$, whose order q divides $p^k - 1$ but does not divide $p^i - 1$ for $0 < i < k$, and q is large enough to resist the MOV attack, but small enough to make the Tate or Weiling pairing feasible. He also selects a generator P of order q for the group G_1, selects his private key s, and computes the public key $P_{Dealer} = sP$. The dealer finally publishes $\{ G_1, G_2, P, P_{Dealer}, \hat{e}, p, q\}$ on the NB and keeps $\{s\}$ secret.

(2) Shadow generation and verification

Let $S = \{S_1, S_2, ..., S_m\}$ be a set of m secrets to be distributed among $G = \{U_1, ..., U_n\}$- a group of n participants, where each U_i has his identity ID_i. The dealer performs the following steps to distribute the shadows.

(2.1) Randomly generate a polynomial $f(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1} \bmod q$, where each $a_k \in_R Z_q$, compute and publish a check vector $V = [V_0, ..., V_{k-1}]$ for each coefficient a_k as $V_k = a_k P$, for k=0~(t-1).

(2.2) Compute U_i's secret shadow as $f(ID_i)$ and the public key $Y_i = f(ID_i)P$ for i= 1~n. The dealer sends $f(ID_i)$ to U_i and publishes $\{Y_i\}$ on the NB.

(2.3) Each participant verifies his shadow by checking whether the following equation holds. If so, the shadow is valid.

$$f(ID_i)P \stackrel{?}{=} \sum_{j=0}^{t-1} ((ID_i)^j \bmod q) V_j \qquad (1)$$

(3) Credit ticket generation

The dealer performs the following steps to compute the m credit tickets $C_1, ..., C_m$

for the secrets $S_1,...,S_m \in S$, where $S_j \in (1, p)$.

(3.1) Randomly choose m distinct integers $r_1,...,r_m \in Z_R$ for each secret $S_1,...,S_m \in S$.

(3.2) Compute the credit ticket $C_j = r_j P_{Dealer}$ and $h_j = x(a_0 C_j) \oplus S_j$ for j=1~m, where $x(a_0 C_j)$ denotes the x-coordinate of the point $a_0 C_j$. The dealer publishes the 2-tuple (C_j, h_j), j=1~m, on the NB.

(4) Sub-shadow verification and secret re-construction

Let W ($|W|=t$) be any subset of t participants in G want to co-operate to reconstruct a secret S_j. Without lose of generality, we may assume $W=\{U_1,...,U_t\}$. Each $U_i \in W$ obtains the 2-tuple $\{C_j, h_j\}$ from the NB and uses his/her secret shadow $f(ID_i)$ to compute a sub-shadow $A_{ij} = f(ID_i)C_j$ $(= (f(ID_i)r_j)P_{Dealer})$. Then, U_i releases A_{ij} to the other co-operators in W. Any co-operator can verify the validity of U_i's sub-shadow A_{ij} by checking whether Equation (2) holds. If the equation does not hold, the verifier announces that U_i is cheating and stops the co-operation.

$$\hat{e}(A_{ij}, P) \stackrel{?}{=} \hat{e}(C_j, Y_i) \tag{2}$$

If all the A_{ij}s released by the t participants are valid, every participant in W can reconstruct the secret S_j as follows, where $x(G)$ denotes the x-coordinate of the point G.

$$S_j = h_j \oplus x(\sum_{i=1}^{t}(\prod_{U_k \in W, k \neq i} \frac{-ID_k}{(ID_i - ID_k)} \bmod q) A_{ij}) \tag{3}$$

Then, all the secrets $S_1,...,S_m \in S$ can be recovered by performing this phase repetitively.

3.2 New GGOC VMSS

Now we can easily extend our scheme to the GGOC VMSS using the bilinear pairing. To simplify our presentation, we assume only one secret S to be distributed among n users, and the qualified subsets are specified by the access structure $\Gamma_0 = f_1 + f_2 + ... + f_w$, where $f_i s$ are *minimal qualified subsets* in the following presentation. The reader can easily extend it to the case of multiple secrets. The Initialization phase (1) and the shadow generation and verification phase (2) are the same as before. Now we describe its corresponding credit ticket generation phase and the sub-shadow verification and secret re-construction phase as follows.

(3) Credit ticket generation

To distribute a secret S with the access structure $\Gamma_0 = f_1 + f_2 + ... + f_w$, the SD executes the following steps.

(3.1) Randomly choose an integer $r \in Z_R$.

(3.2) Compute the credit $C = rP_{Dealer}$ and $h_k = S \oplus x(\sum_{U_i \in f_k} f(ID_i)C)$, $k = 1 \sim w$, for each minimal qualified subset f_k. $x(\sum_{U_i \in f_k} f(ID_i)C)$ denotes the x-coordinate of the point $\sum_{U_i \in f_k} f(ID_i)C$. The dealer finally publishes the vector $(C, h_1, h_2, ..., h_w)$ on the NB.

(4) Sub-shadow verification and secret re-construction

When those users corresponding to a *minimal qualified subset* f_j $(1 \le j \le w)$

of Γ_0 are willing to recover the secret *S*, then they execute the following steps.

(4.1) For each $U_i \in f_k$, he computes $A_i = f(ID_i)C$ and contributes A_i to his group.

(4.2) All the members of the group f_k can verify the sub-shadows from other members by checking whether Equation (2) holds. After all group members' sub-shadows are verified, each member can recover the secret using the following equation (4).

$$S = h_k \oplus x(\sum_{U_i \in f_k} A_i) \tag{4}$$

4. Security analysis and performance evaluation

We show the correctness of our schemes in Section 4.1, analyzes the security in Section 4.2, and evaluate its performance in Section 4.3.

4.1 Correctness of the schemes

Before analyzing the security, we first prove the correctness of Equation (1~4) as follows.

(1) If the dealer honestly distributes the shadows, then Equation (1) holds as follows.

$$f(ID_i)P = (\sum_{j=0}^{t-1} a_j \cdot ID_i^{\,j} \bmod q)P = \sum_{j=0}^{t-1} (a_j \cdot ID_i^{\,j} \bmod q)P = \sum_{j=0}^{t-1} (ID_i^{\,j} \bmod q)(a_j P) =$$

$$= \sum_{j=0}^{t-1} ((ID_i)^j \bmod q)V_j$$

(2) If U_i honestly submits his sub-shadow A_{ij} in the (*t*, *n*) VMSS, then Equation (2) holds as follows.

$$\hat{e}(A_{ij}, P) = \hat{e}(f(ID_i)C_j, P) = \hat{e}(f(ID_i)r_j P_{Dealer}, P) = \hat{e}(r_j P_{Dealer}, f(ID_i)P)$$
$$= \hat{e}(C_j, Y_i)$$

(3) If U_i honestly submits his sub-shadow A_{ij} in the GGOC VMSS, then Equation (2) holds as follows.

$$\hat{e}(A_i, P) = \hat{e}(f(ID_i)C, P) = \hat{e}(f(ID_i)rP_{Dealer}, P) = \hat{e}(rP_{Dealer}, f(ID_i)P)$$
$$= \hat{e}(C, Y_i)$$

(4) If all the A_{ij} s released by the *t* participants are valid in the (*t*, *n*) VMSS, then the secret S_j can be correctly reconstructed using equation (3) as follows.

$$h_j \oplus x(\sum_{i=1}^{t}(\prod_{U_k \in W, k \neq i} \frac{-ID_k}{(ID_i - ID_k)} \bmod q) A_{ij})$$
$$= h_j \oplus x(\sum_{i=1}^{t}(\prod_{U_k \in W, k \neq i} \frac{-ID_k}{(ID_i - ID_k)} \bmod q) f(ID_i) C_j)$$
$$= h_j \oplus x(\sum_{i=1}^{t}(\prod_{U_k \in W, k \neq i} \frac{-ID_k}{(ID_i - ID_k)} \bmod q) f(ID_i) r_j P_{Dealer})$$
$$= h_j \oplus x(r_j \cdot (\sum_{i=1}^{t} \prod_{U_k \in W, k \neq i} \frac{-ID_k}{(ID_i - ID_k)} f(ID_i) \bmod q) P_{Dealer})$$
$$= h_j \oplus x((r_j \cdot a_0) P_{Dealer}) = h_j \oplus x(a_0 C_j) = x(a_0 C_j) \oplus S_j \oplus x(a_0 C_j) = S_j$$

(5) If all the sub-shadows released by the members of the group f_k are valid in the GGOC VMSS, then the secret *S* can be re-constructed using Equation (4) as follows.

$$h_k \oplus x(\sum_{U_i \in f_k} A_i)$$
$$= S \oplus x(\sum_{U_i \in f_k} f(ID_i)C) \oplus x(\sum_{U_i \in f_k} A_i)$$
$$= S \oplus x(\sum_{U_i \in f_k} f(ID_i)C) \oplus x(\sum_{U_i \in f_k} f(ID_i)C)$$
$$= S$$

4.2 Security analysis

Now we are ready to show the security through a serious analysis of possible attacks and property analysis.

Attack 1. An adversary tries to derive any secret values *s*, a_k and $f(ID_i)$ from the public values $P_{Dealer} = sP$, $V_k = a_k P$ and $Y_i = f(ID_i)P$. He fails in such an attack because the adversary has to face the ECDLP problem.

Attack 2. A malicious participant U_i who has obtained some previously recovered secrets now tries to reveal any remaining secret in *S* without the co-operation of other (*t*-1) co-operators. To derive any remaining secret S_c, it is required to get the values a_0 or $a_o r_c P$. However, knowing some secrets S_a, S_b, and the corresponding values $C_a = r_a P_{Dealer}$, $C_b = r_b P_{Dealer}$, $x(a_0 C_a)$ and $x(a_0 C_b)$, U_i cannot derive any one of the secret values a_0, $a_0 P$, r_a or r_b, because of the ECDLP problem. Therefore, U_i cannot derive any remaining secret without the co-operation of other (*t*-1) participants.

Attack 3. The dealer tries to distribute fake shadows without being detected by Equation (1). After the dealer has published the checking vector $V = [V_0, ..., V_{k-1}]$ where $V_k = a_k P$, then any shadow that is not derived as $f(ID_i)$ cannot satisfies Equation (1). Therefore, any fake shadow would be detected.

Attack 4. A dishonest participant tries to submit a fake sub-shadow A_{ij}. Since the dealer has published the values Y_i and C_j, a fake sub-shadow A_{ij} that cannot satisfy Equation (2) can be easily detected.

Attack 5. Assume an adversary happened to know some disclosed secrets S_a and S_b. Now he tries to recover any remaining secrets. From the disclosed secrets S_a and S_b, the adversary can further derive the values $C_a = r_a P_{Dealer}$, $C_b = r_b P_{Dealer}$, $x(a_0 C_a)$ and $x(a_0 C_b)$, but he cannot derive the secret

value a_0 because of the ECDLP problem. He, therefore, cannot derive any remaining secret further.

(*t, n*) access property: Assume that there were $t' < t$ members trying to solve the equations in (3). They cannot uniquely determine the value due to the Lagrange interpolating equation. Therefore, our scheme realizes the (*t*, *n*) threshold access rule.

GGOC access property: In the GGOC case, to derive the secret *S* from the public values $(C, h_1, h_2, ..., h_w)$ **,** one should first derive at least one value $x(\sum_{U_i \in f_k} f(ID_i)C)$ for any f_k **.** However, only through the co-operation of the users corresponding to the minimal qualified subset f_j $(1 \le j \le w)$ can the value $x(\sum_{U_i \in f_k} f(ID_i)C)$ and the secret *S* can be acquired. Our scheme therefore realizes the access structure Γ_0.

4.2 Performance evaluation

Now we compare the performance of our scheme with that of the CHY scheme and that of the LW scheme. The comparisons are summarized in Table 1. The CHY scheme would disclose the remaining secrets when a participant (or an adversary) has obtained some secrets, and the participant in the LW scheme can submit fake sub-shadows without being detected (the fake sub-shadow problem). Only our scheme can resist all the attacks. Due to the difference of the shadow distribution, our scheme can be extended to the GGOC case, while the CHY scheme and the LW scheme cannot be extended to the GGOC case.

Regarding the computational performance, we first denote some symbols of

computational operations. T_{pair} denotes one pairing operation, T_{scalar} denotes one scalar multiplication in elliptic curves, $T_{\exp}$ denotes one modular exponentiation, T_{inv} denotes one modular inverse operation, and T_{mul} denotes one modular multiplication. T_{pair} is the most expensive computation among these operations among these operations. To detect the cheating of other *t*-1 users, our scheme requires each user to perform $2(t\text{-}1)T_{pair}$, while the CHY scheme requires only $2(t\text{-}1)T_{\exp}$ and the LW scheme requires around $5(t\text{-}1)\ T_{\exp}$. So, the CHY scheme is more computationally efficient in detecting the cheating. However, to reconstruct one secret by the *t* users, the CHY scheme needs $tT_{mul}+t((n\text{-}1)+(n\text{-}t))\ T_{\exp}$ and the LW scheme requires around $(2t\text{-}1)\ T_{mul}+t((n\text{-}1)+(n\text{-}t))\ T_{\exp}$, while our scheme needs $t(t\text{-}1)T_{inv}+tT_{scalar}+(2t\text{-}1)T_{mul}$. Please notice that the *t* users in the CHY scheme (and the LW scheme) do not know the modulus *R*; therefore, the exponent $\Delta_i=(\prod_{U_k\in W,U_k\neq U_i}-ID_k)\cdot(\prod_{U_k\in G,U_k\notin W}(ID_i-ID_k))$ could not be reduced by the modulus *R* and the computation increases as the number of the participants increases. That is, the computational performance of the CHY scheme (and the LW scheme) will degrade very quickly when the number of participants increases. Overall, the CHY scheme has better computational performance when the number of participants is small, but it degrades quickly as the number of the participants increases.

Regarding the number of published values in the (*t*, *n*) VMSS schemes, our scheme outperforms the CHY scheme and the LW scheme. To share *m* secrets, the CHY scheme publishes 3*m* values and the LW scheme publishes 4*m* values; while our scheme only publishes 2*m* values.

5. Conclusions

Based on bilinear pairing, this paper has proposed a (t, n) threshold verifiable multi-secret sharing scheme and a GGOC verifiable multi-secret sharing scheme that can serve as robust and efficient building blocks for many distributed systems. Compared to Chang-Hwang-Yang's recent research, our scheme can further reduce 33% of the number of published values and can prevent any entity from illegally deriving any remaining secrets even assuming the entity has gained the previous secrets.

Table 1. Comparisons of our scheme and its counterparts

	CHY	LW	Ours
Extension to GGOC	NO	NO	Yes
Fake sub-shadow problem	NO	Yes	No
A participant who obtained some secrets could derive remaining secrets without others' co-operation	Yes	No	No
An adversary who obtained some secrets could derive remaining secrets	Yes	No	No
Detect cheating of dealer (done by U_i)	$(2t+1)T_{\exp}$	$(2t+1)T_{\exp}$	$tT_{scalar}+(t+1)T_{\exp}$
Detect cheating of other t-1 users (done by U_i)	$2(t-1)T_{\exp}$	$5(t-1)T_{\exp}$	$2(t-1)T_{pair}$
One secret re-construction	$tT_{mul}+t((n-1)+(n-t))\ \ T_{\exp}$	$(2t-1)T_{mul}+t((n-1)+(n-t))$ $T_{\exp}$	$t(t-1)T_{INV}+tT_{scalar}+(2t-1)T_{mul}$
Public values for constructing m secrets	$3m$	$4m$	$2m$

References

[1] Blakley, G.R., 1979, "Safeguarding Cryptographic Keys," *in Proceedings of the National Computer Conference 1979, American Federation of Information Processing Societies Proceedings*, New York, Vol. 48, pp. 313-317.

[2] Chen, L., Gollmann, D., Mitchell, C. J., and Wild, P., "Secret sharing with reusable polynomials", Proceedings of ACISP'97, pp.183-193.

[3] Chien, H. Y., Jan, J. K., and Tseng, Y. M., "A practical (t, n) multi-secret sharing scheme", IEICE Trans. Fundamentals E83-A (12), 2000, pp. 2762-2765.

[4] Chor, B., Goldwasser, S., Micali, S., Awerbuch, B., "Verifiable secret sharing and achieving simultaneity in the presence of faults", Proc. of 26th IEEE Symp. FOCS, 1985, pp. 251-260.

[5] Harn, L., 1995, "Efficient Sharing (Broadcasting) of Multiple Secrets," *IEE Proceedings- Computers and Digital Techniques*, Vol. 142, No. 3, pp. 237-240.

[6] He, J., and Dawson, E., 1995, "Multi-Secret Sharing Scheme Based on One-Way Function," *Electronics Letters*, 31(2), pp. 93-94.

[7] He, W. H., and Wu, T. S., "Comment on Lin-Wu (t, n) threshold verifiable multi-secret sharing scheme", IEE Proc. Comput. Digit. Tech. 148 (3), 2001, pp. 139.

[8] Jackson, W.-A., Martin, K. M., and O'Keefe, C. M.,1994, "On sharing many secrets," in Advances in Cryptology-*Asiacrypt'94*, Springer-Verlag, pp. 42-54.

[9] Lin, T. Y., and Wu, T. C., "(t, n) threshold verifiable multi-secret sharing scheme based on factorization intractability and discrete logarithm modulo a composite problems", IEE Proc. Comput. Digit. Tech. 146 (5), 1999, pp.264-268.

[10] Pinch, R. G. E., 1996, "Online multiple secret sharing," *Electronics Letters*, Vol. 32, No. 12, pp. 1087-1088.

[11] Shamir, A., 1979, “How to Share a Secret,” *Communications of the ACM,* 22(11), pp. 612-613.

[12] Stadler, M., “Publicly verifiable secret sharing”, Eurocrypt’96, 1996, pp. 190-199.

[13] Sun, H. M., 1999, “On-line multiple secret sharing based on a one-way function,” *Computer Communications 22*, pp. 745-748.

[14] Tan, K. J.,, Zhu, H-W.,, and Gu, S.J., 1999, “Cheater Identification in *(t,n)* Threshold Scheme,” *Computer Communications,* 22 , pp. 762-765.

[15] Tompa, M., Woll, H., “How to share a secret with cheaters”, Journal of Cryptology, 1988, pp. 133-138.

[16] Wu, T.C., and Wu, T.S., 1995, “Cheating detection and cheater identification in secret sharing schemes,” *IEE Proceedings- Computers and Digital Techniques*, Vol 142, pp. 367-369.

[17] Menezes, A. J., Okamato, T. and Vanstone, S. A., “Reducing elliptic curve logarithms to logarithms in a finite field”, *IEEE Trans. on Info*., 39(5), 1639-1646, 1993.

[18] Boneh, D., and Franklin, M., “Identity-based encryption from the Weil pairing”, Crypto’01, LNCS 2139, 2001, pp. 213-229.

[19] Chang, T. Y., Hwang, M. S., and Yang, W. P., “An improvement on the Lin-Wu (*t, n*) threshold verifiable multi-secret sharing scheme”, Applied Mathematics and computation 163, 2005, pp. 169-178.

[20] Naor, M., and Wool, A., “Access control and signatures via quorum secret sharing”, IEEE trans. Parallel and Distributed Systems, vol. 9, no. 9, 1998, pp. 909-922.

[21] Chien, H. Y., Jan, J. K. and Tseng, Y. M. ”An unified approach to secret sharing schemes with low distribution cost”, Journal of the Chinese Institute of Engineers,

Vol. 25, No. 6, pp. 723-733 (2002).

[22] Jackson, W.-A., Martin, K. M., and O'Keefe, C. M.,1994, "On sharing many secrets," in Advances in Cryptology-*Asiacrypt'94*, Springer-Verlag, pp. 42-54.

[23] H.Y. Lin, and L. Harn, "A Generalized secret sharing scheme with cheater detection," *AsiaCrypt'91*, pp. 83-87, 1991.

[24] C.S. Laih, and L. Harn, "Generalized threshold cryptosystems," *AsiaCrypt'91*, pp. 88-92, 1991.

[25] J. Benaloh, and J. Leichter, "Generalized secret sharing and monotone functions," *EuroCrypt'90*, pp. 27-35, 1990.

[26] Ito, Saito, and Nishizeki, "Secret sharing scheme realizing general access structure," IEEE Global Telecommunications Conference, pp. 99-102, 1987.

[27] Deng, R.H., Gong L., Lazar, A.A., and Guo, W., 1995, "Authenticated Key Distribution and Secure Broadcast Using No Conventional Encryption: A Unified Approach Based on Block Codes," *IEEE GLOBE Telecommunication Conference 1995*, pp. 1193-1197.

[28] Bertilsson, M., and Ingemarsson, I., 1992, "A construction of practical secret sharing schemes using linear block codes," in Advances in Cryptology-*Auscrypt'92*, Springer-Verlag, pp. 2-21.

[29] Karnin, Greene, J. W., and Hellman, M. E., 1983, "On secret sharing systems," *IEEE Transactions on Information Theory*, IT-29, pp. 35-41.