A Publicly Verifiable Secret Sharing Scheme Based on Multilinear Diffie-Hellman Assumption

Qiao Peng¹, Youliang Tian^{1,2} (Corresponding author: Youliang Tian)

College of Science, Guizhou University¹ Institute of Cryptography and Data Security, Guizhou University² Guiyang 550025, China (Email: youliangtian@163.com)² (Received Oct. 22, 2015; revised and accepted Dec. 7 & Dec. 17, 2015)

Abstract

Using multiple linear of multilinear map, we propose a simple, non-interactive and effective publicly verifiable secret sharing (PVSS) scheme based on multilinear Diffie-Hellman assumption (MDH). Up to now, the publicly verifiable of secret sharing is still an issue. In this paper, we set the sharing secret is a multiple linear pairing, we apply the multiple linear property of multilinear map for the shares authentication to achieve publicly verifiability of secret sharing. What's more, the batch verification technique is used to reduce the computational overhead at share verification phase. Compared with the existing programs, this scheme has improved communication efficiency under the same security level and it can meet those high efficiency and security of the communication requirements of the application scenarios. In addition, we apply our PVSS scheme to electronic voting skillfully. At last, the performance analysis results show the publicly verifiability, security and practicality of our scheme in the random oracle and under MDH assumption.

Keywords: Electronic voting, multilinear map, multilinear Diffie-Hellman assumption, publicly verifiable secret sharing

1 Introduction

Secret sharing is an important research content of modern cryptography, it is a method of increasing the security of cryptography system. The earliest secret sharing schemes were proposed by Shamir [20] and Blakey [2] in 1979. Shamir's (t, n) threshold secret sharing scheme is based on polynomial interpolation in a finite field. In Shamir's scheme, the secret is able to be reconstructed by t or more participants at secret reconstruction phase, while any subset of t-1 or less participants has no information about the secret. Threshold secret sharing [1, 10] has remarkable effect on cryptography due to its effective and applicability. However, it still has the following drawbacks:

- 1) Unable to identify the honesty of the dealer;
- 2) Unable to detect dishonest participants and prevent cheating behavior.

In order to prevent malicious behavior of the dealer and participants, a new type of secret sharing scheme was first proposed by Feldman [11], called Verifiable Secret Sharing (VSS) schemes, which solved the security issues mentioned above. However, it also pledges that the participants only verify their own shares but cannot verify the other participants received shares. VSS scheme such as all require the availability of private channel from the dealer to each of the participants individually, but the communication over the private channel of VSS scheme is obviously not publicly verifiable.

However, in publicly verifiable secret sharing (PVSS) scheme, the dealer broadcasts information to the participants without needing to maintain a private channel, which avoids the interaction between the dealer and the participants, as well as the interaction among the participants. The notion of PVSS was first introduced by Stadler [21] in public key setting. PVSS scheme with the objective that anyone, not just the participants, can verify whether the distributed shares are valid without revealing any information about the secret at the secret distribution phase and whether each participant releases the correct share at the secret reconstruction phase. Moreover, Stadler expressed the main goal of threshold secret sharing scheme was that each authorized subset of the access structure could reconstruct the secret. The PVSS scheme reduces the overhead of communication and safeguards the security of the scheme because it does not require private channels. In view of these advantages, Schoenmakers [19] proposed a simple PVSS scheme based on discrete logarithm problem and gave its applications in electronic voting systems and key escrow. Later, some publicly verifiable secret sharing schemes based on traditional publickey systems were proposed [6, 7, 26]. Although PVSS plays a powerful role in threshold cryptography, the security of this kind of scheme was either based on the integer factoring problem or the discrete logarithm problem. Until 1993, Menezes et al. [16] presented the Weil pairing, which was defined on an elliptic curve and could be used to solve the decision Diffie-Hellman (DDH) problem effectively. Subsequently, many pairing-based secret sharing schemes were proposed [3, 23, 24]. For example, WU and TSENG [25] proposed the first pairing-based PVSS scheme in 2011, they had showed the security of their PVSS scheme under the bilinear Diffie-Hellman assumption, but the computation overhead of their scheme was considerable, especially in the share verification phase.

Recently, multilinear map has received extensive attentions from cryptographic researchers, which has been applied to public key cryptography [8, 12] successfully. In [13], Garg et al. presented a public and secure attribute-based signcryption scheme based on multilinear map, this signcryption scheme gave the foundation method of carrying out secure communication in social network. In 2009, on the basis of multilinear map, by using multiple linear pairing, Ruckert et al. [18] have constructed efficient aggregate and verifiable encrypted signatures without random oracles. From the above references, we can easily know that multiple linear pairing has been an important tool for constructing encryption and signature algorithms, and the security of the signcryption schemes is guaranteed under the multilinear Diffie-Hellman assumption, but there is almost no secret sharing scheme based on multiple linear pairing presently.

Consequently, by using multiple linear pairing, in this paper, we propose a non-interactive, simple and effective PVSS scheme, whose security is based on multilinear Diffie-Hellman assumption. In our scheme, we assume that the secret is a multiple linear paring, by using multiple linear property of multilinear map and the batch verification technique to prevent cheating at secret distribution phase and reduce computational overhead at verification phase, respectively. Moreover, anyone can identify the process of distributing and recovering secret publicly without implementing the interactive protocol such as DLEQ(g1, h1; g2, h2) by Chaum and Pedersen in [5], it's an effective solution to prevent dishonest dealer and participants, thereby reducing the communication cost. Furthermore, we show that in the random oracle model and under multilinear Diffie-Hellman assumption, our proposed scheme is securely and effectively. In addition, the performance analysis shows that it is less communication overhead and more effective than the previous schemes [11, 17, 25], so it can be more applicable in those high efficiency of the communication requirements of the application scenarios.

The rest of the paper is organized as follows. We briefly describe the concept of multilinear map and the related security assumptions. At the same time, we review the model of PVSS scheme in Section 2. In Section 3, we present our new publicly verifiable secret sharing based on multiple linear pairing. And then in Section 4, we make the scheme analysis, which focus on the proof of the correctness and security, as well as the performance comparison. In Section 5, the application of our PVSS scheme in electronic voting is briefly presented. We introduce a conclusion and our next work in Section 6.

2 Preliminaries

In this section, we briefly describe the definition of multilinear map, the related security assumptions, and recall the publicly verifiable secret sharing (PVSS) scheme.

2.1 Multilinear Maps

Boneh and Silverberg (BS) [4] first proposed the concept of multilinear map and described many cryptographic applications in 2003. The definition of BS is that: Let G_1 and G_2 be two groups which have the same prime order q. In particular, G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group. A map $e : G_1^n \to G_2$ is an *n*-multilinear map if it satisfies the following three properties:

- 1) Multilinear: For all $g_1, g_2, \dots, g_n \in G_1$ and $a_1, a_2, \dots, a_n \in Z_q^*$, we have $e_n(a_1g_1, a_2g_2, \dots, a_ng_n) = e_n(g_1, g_2, \dots, g_n)^{a_1a_2\cdots a_n}$;
- 2) Non-degenerate: If $g \in G_1$ is a generator of G_1 , then $e_n(g, g, \dots, g)$ is a generator of G_2 ;
- 3) Computable: For all $g_1, g_2, \dots, g_n \in G_1$, there is an efficient algorithm to compute $e_n(g_1, g_2, \dots, g_n)$.

2.2 Security Problems and Assumptions

- Computational Diffie-Hellman (CDH) problem: Given $g, ag, bg \in G_1$ for some $a, b \in Z_q^*$, it is difficult to compute $abg \in G_1$.
- **Discrete logarithm (DL) problem:** Given $g, ag \in G_1$, it is hard to compute $a \in Z_q^*$.
- Multilinear discrete logarithm (MDL) problem: Let G be a finite cyclic group with prime order q, for all $k > 1, 1 \le i \le k$ and $g_i \in G$, given (i, g_i, ag_i) for some $a \in \mathbb{Z}_a^*$, it is hard to compute a.
- *n*-Multilinear computational Diffie-Hellman (*n*-MDH) problem: Given $g, a_1g, a_2g, ..., a_ng \in G_1$ for some random selective $a_1, a_2, \cdots, a_n \in Z_p$, where g is a generator of group G_1 , it is hard to compute $e_n(g, g, \cdots, g)^{a_1a_2\cdots a_n} \in G_2$.
- **MDH assumption:** No PPT algorithm can solve the MDH problem with a non-negligible advantage.

2.3 Model of PVSS

In this section, the model of (t, n) threshold publicly verifiable secret sharing (PVSS) scheme is presented. Let t and n be two positive integers such that $1 \leq t \leq n$. Let $U_1, ..., U_n$ denote the participants, and D denotes the dealer. An access structure can be a (t, n) threshold scheme for $1 \leq t \leq n$, it means that any subset of t or more participants is able to reconstruct the secret, while the subset of at most t-1 participants cannot recover the secret and has no information about it. The system of a PVSS scheme consists of three phases are described below.

- 1) Initialization phase: On input the number n of participants, a threshold t, it outputs all public parameters as well as participants' private keys and the corresponding public keys as part of the system parameters.
- 2) **Distribution phase:** On input a secret *s*, the distribution phase consists of two steps as follows.
 - a. Share distribution: The dealer D distributes a secret s among n participants, the dealer uses the participants' private keys and public parameters to encrypt secret and then publishes some specific value Y_i (the shares are embedded into these specific values Y_i) to the participants U_i , where for $i = 1, 2, \dots, n$.
 - b. **Public verification:** This step can be executed by a third party and determines whether the distributed shares are valid. Anyone not just the participants can verify these specific values Y_i by checking some equations. If all the checking equations hold, then these specific values Y_i are believed to be correctly published by the dealer, and the shares included in Y_i are valid. Once the equations do not hold, we say that the dealer fails to distribute a secret, and then break the scheme.
- 3) **Reconstruction phase:** The reconstruction phase contains decryption of the shares and reconstruction of the secret:
 - a. Decryption of the shares: Each participant uses his/her own private key to obtain the corresponding share s_i from the specific value Y_i , respectively.
 - b. Reconstruction of the secret: When the qualified participants offered at least t correct shares s_i , then the secret s can be recovered from these shares s_i by threshold technique such as Lagrange interpolation.

3 Proposed PVSS Scheme

In this section, we present our non-interactive and effective PVSS scheme based on multiple linear pairing. First, the key generation center (KGC) generates mpublic parameters $P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)}, m \in {}_{R}Z_{q}^{*}$. We assume that the secret $S = e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{a}$ will be distributed by the dealer D among n participants, where $a \in Z_{q}^{*}$. Let $U = \{U_{1}, U_{2}, \cdots, U_{n}\}$ be a set of n qualified participants. The PVSS scheme consists of three phases: Initialization phase, Distribute phase and Reconstruct phase.

1) Initialization phase

Let G_1 and G_2 be two groups, separately denote additive cyclic group and multiplicative cyclic group which have the same prime order q. Assuming that there exists a multilinear map $e: G_1^n \to G_2$ among G_1 and G_2 . The independently generators P, Q of groups G_1 and G_2 are selected using appropriate public procedure. Each participant U_i chooses a private key $d_i \in Z_q^*$ and compute the corresponding public key $P_i = d_i P_{pub}^{(i)}$ for i = 1, 2, ..., n.

2) Distribute phase

The distribution phase consists of two steps as following:

a. Distribution of the shares: The dealer D wishes to distribute a secret among n participants. The dealer D first chooses a random polynomial $f(x) = \sum_{j=0}^{t-1} a_j x^j$ of degree at most t-1 with coefficients in Z_q . Here $f(0) = a_0 = a$. And then the dealer keeps this polynomial secretly but computes and publishes the following values: the related commitments $C_j = a_j \cdot P$, for j = 0, 1, ..., t-1, $X_i = f(i) \cdot P$ and $\gamma_i = f(i) \cdot P_{pub}^{(i)}$. The dealer also publishes the encrypted shares $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \dots, P_{pub}^{(m)})^{f(i)}$ for i = 1, 2, ..., n.

Each X_i can be constructed by all public values C_i as follows:

$$X_i = f(i) \cdot P$$

= $\sum_{j=0}^{t-1} a_j \cdot (i^j) \cdot P$
= $\sum_{j=0}^{t-1} (i^j) \cdot a_j \cdot P$
= $\sum_{j=0}^{t-1} (i^j) \cdot C_j$

b. Verification of shares: Anyone first can recover $X_i = \sum_{j=0}^{t-1} (i^j) \cdot C_j$ from the value C_j and then checks equation (1) by public values C_j and X_i, γ_i , for j = 0, 1, ..., t - 1, i = 1, 2, ..., n. Equation (1):

$$e_m(\gamma_1, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, X_j) = e_{m+1}(\gamma_1, \gamma_2, ..., \gamma_{m-1}, P)$$
(1)

If the Equation (1) holds, then the verifier believes that these specific values Y_i correctly published by the dealer D and the verifier can confirm that each Y_i holds for i = 1, 2, ..., n. The proof is as follows:

$$Y_{i} = e_{m}(P_{i}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{f(i)}$$

$$= e_{m}(d_{i} \cdot P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{f(i)}$$

$$= e_{m}(P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{d_{i} \cdot f(i)}$$

$$= e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{d_{i} \cdot f(i)}.$$

3) **Reconstruct** phase

This phase is divided into decryption of the shares and the reconstruction of the secret:

a. Decryption of the shares: Each participant U_i uses his/her own private key d_i to compute the corresponding share $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)}$ by computing the following equation:

$$\begin{split} Y_{i}^{d_{i}^{-1}} &= e_{m}(P_{i},P_{pub}^{(1)},...,P_{pub}^{(i-1)},P_{pub}^{(i+1)},\\ &\cdots,P_{pub}^{(m)})^{f(i)\cdot d_{i}^{-1}}\\ &= e_{m}(d_{i}\cdot P_{pub}^{(i)},P_{pub}^{(1)},...,P_{pub}^{(i-1)},P_{pub}^{(i+1)},\\ &\cdots,P_{pub}^{(m)})^{f(i)\cdot d_{i}^{-1}}\\ &= e_{m}(P_{pub}^{(i)},P_{pub}^{(1)},...,P_{pub}^{(i-1)},P_{pub}^{(i+1)},\\ &\cdots,P_{pub}^{(m)})^{f(i)}\\ &= e_{m}(P_{pub}^{(1)},P_{pub}^{(2)},...,P_{pub}^{(m)})^{f(i)}\\ &= S_{i} \end{split}$$

b. Reconstruct of the secret:

Any t shareholders U_i with the correct shares S_i can reconstruct the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$, for i = 1, 2, ..., t. The secret S is obtained by Lagrange interpolation as Equation (2):

$$S = \prod_{i=1}^{t} S_i^{\lambda_i} = e_m (P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a \quad (2)$$

Where $\lambda_i = \prod_{j \neq i} \frac{i}{j-i}$ is Lagrange coefficient.

4 Scheme Analysis

This section introduced the proof of the correctness and security of the proposed scheme, and we make performance analysis mainly in the computation and communication aspects.

4.1 Correctness Analysis

Lemma 1. First, we verify the equation $e_m(\gamma_1, ..., \gamma_{j-1}, P_{pub}, \gamma_{j+1}, ..., \gamma_{m-1}, X_j) = e_m(\gamma_1, \gamma_2, ..., \gamma_{m-1}, P)$ (1).

Proof. From the public values $X_i = f(i) \cdot P$, $\gamma_i = f(i) \cdot P_{pub}^{(i)}$. we can gain that

$$e_{m}(\gamma_{1}, \gamma_{2}, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, X_{j})$$

$$= e_{m}(\gamma_{1}, \gamma_{2}, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, f(j) \cdot P)$$

$$= e_{m}(\gamma_{1}, \gamma_{2}, ..., \gamma_{j-1}, f(j) \cdot P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, P)$$

$$= e_{m}(\gamma_{1}, \gamma_{2}, ..., \gamma_{j-1}, \gamma_{j}, \gamma_{j+1}, ..., \gamma_{m-1}, P)$$

$$= e_{m}(\gamma_{1}, \gamma_{2}, ..., \gamma_{m-1}, P).$$

Hence, Equation (1) holds, the shares distributed by the dealer are valid. $\hfill \Box$

Lemma 2. And then verify that the method of reconstructing the secret is correct. In other words, it is need to verify equation $S = \prod_{i=1}^{t} S_i^{\lambda_i} = e_m (P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$.

Proof. From the known share value $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)}$, which is computed from private key d_i and specific public value Y_i , we can get that

$$\begin{split} \prod_{i=1}^{t} S_{i}^{\lambda_{i}} &= \prod_{i=1}^{t} \left(e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)} \right)^{\lambda_{i}} \\ &= e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{\sum_{i=1}^{t} f(i) \cdot \lambda_{i}} \\ &= e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)} \\ &= e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{a} \\ &= S \end{split}$$

The Equation (2) holds, so the method of secret reconstruction is correct. $\hfill \Box$

4.2 Security Analysis

In this section, we present security analysis of our proposed scheme under the multilinear Diffie-Hellman (MDH) assumption.

We first consider the security of the shares $S_i = e_m (P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$. Given the public values $P_{pub}^{(i)}, P_i, X_i$ and Y_i for i = 1, 2, ..., n, we observe that the difficulty of computing the share S_i is equivalent to solve the multilinear Diffie-Hellman (MDH) problem as described in Section 2. Consequently, we have the following lemma.

Lemma 3. The encryption of shares is security in the proposed PVSS scheme if and only if the MDH assumption holds.

Proof. ⇐ By contradiction proof. Assuming that the MDH assumption holds but the encryption of shares is not security. Since the method of share encryption does not hold, then there exists an Algorithm A can compute the shares $S_i = e_m (P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$ with a

non-negligible probability ε for the given public value $P_{pub}^{(i)}, P_i, X_i$ and Y_i . Then we want to prove that an attacker can solve the MDH problem with the same probability using the Algorithm A.

The MDH problem is that given $a_1P, a_2P, ..., a_mP$ for some $a_1, a_2, \cdots, a_m \in Z_q^*$, it is hard to compute $e_m(P, P, \cdots, P)^{a_1 a_2 \cdots a_m}$. Hence, we try to compute the value $e_m(P, P, \dots, P)^{a_1 a_2 \dots a_m}$ using A in the following. The attacker chooses random elements $a_1, a_2, \cdots, a_m, b \in$ Z_q^* and $a_1^{'}, a_2^{'}, \cdots, a_m^{'}, b^{\prime} \in Z_q^*$. For the given values $Q_1 = a_1 P, Q_2 = a_2 P, ..., Q_m = a_m P, Q = bP$, the attacker first computes and feeds the values $P_{pub}^{(i)} =$ $a'_i \cdot Q_i, P_i = a_i \cdot P^{(i)}_{pub}, \quad X_i = b' \cdot Q \text{ and } Y_i = e_m(P^{(1)}_{pub}, P^{(2)}_{pub}, ..., P^{(m)}_{pub})^{a_i \cdot bb'}$ to A, where i = 1, 2, ..., n. Since the input of A is uniformly distributed and $X_i =$ $b' \cdot Q = b' \cdot bP = f(i)P$ is known, we obtain that $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)} = e_m(a_1a_1^{'} \cdot P, \ a_2a_2^{'})^{f(i)} = e_m(a_1a_1^{'} \cdot P, \ a_2a_2^{'})^{f(i)}$ $\cdot P, ..., a_m a'_m \cdot P)^{f(i)} = e_m(P, P, ..., P)^{a_1 a'_1 \cdot a_2 a'_2 \cdot a_m a'_m \cdot f(i)}$ $= e_m(P, P, ..., P)^{a_1a'_1 \cdot a_2a'_2 \cdot a_ma'_m \cdot bb'}$ with the same nonnegligible probability ε . By taking $(S_i)^{1/(a'_1a'_2\cdots a'_mbb')}$ $= e_m(P, P, ..., P)^{a_1 a_2 \cdots a_m}$, we know that the attacker is able to compute $e_m(P, P, \cdots, P)^{a_1 a_2 \cdots a_m}$ with the probability ε . It is a contradiction to the above MDH assumption.

It shows that the MDH assumption holds, then the encryption of shares is secure.

⇒ By contradiction proof. Assuming that the encryption of shares is secure but the MDH assumption does not hold. Because the MDH assumption does not hold, then there exists an algorithm *B* can compute $e_m(P,P,\cdots,P)^{a_1a_2\cdots a_m}$ with a non-negligible probability ε for *m* random elements $a_1P, a_2P, \ldots, a_mP \in G_1$, where $a_1, a_2, \cdots, a_m \in Z_q^*$. The attacker chooses random elements $\beta_1, \beta_2, \cdots, \beta_m, b \in Z_q^*$ and $\beta'_1, \beta'_2, \cdots, \beta'_m, b' \in Z_q^*$. When feeding $Q = bP, X_i = b'Q$ to *B*, the attacker computes and inputs $Q'_1 = \beta_1 \cdot P, Q'_2 = \beta_2 \cdot P, \ldots, Q'_m = \beta_m \cdot P$, $P_{pub}^{(i)} = \beta'_i P$ for i = 1, 2, ..., n. Then the share S_i must satisfy that $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)} = e_m(Q'_1, Q'_2, \cdots, Q'_m) = e_m(\beta_1 P, \beta_2 P..., \beta_m P)$.

Since the input of *B* is uniformly distributed, we can compute $X_i = b'Q = b'bP = f(i)P$ with the same probability ε because of $Q = bP, X_i = b'Q$. Therefore, we can obtain that $e_m(\beta_1'P, \beta_2'P, ..., \beta_m'P)^{f(i)} = e_m(\beta_1P, \beta_2P..., \beta_mP)$. Which produces

$$e_m(P, P..., P)^{\beta'_1\beta'_2\cdots\beta'_mbb'} = e_m(P, P..., P)^{\beta_1\beta_2\cdots\beta_m}.$$

Due to the MDH assumption does not hold. So the algorithm B can compute $e_m(P, P..., P)^{\beta_1\beta_2\cdots\beta_m}$ with the same non-negligible probability ε , and then the share S_i can be computed by algorithm B. Hence, the encryption of shares is not secure.

It shows that the encryption of shares is secure, the MDH assumption must hold. $\hfill \Box$

Lemma 4. If only t-1 participants can work together to

recover the secret in the proposed scheme, then the Multilinear Diffie-Hellman (MDH) problem can be solved.

Proof. At first, we recall that the MDH problem is to compute $e_m(P, P, \dots, P)^{a_1 a_2 \dots a_m}$ for given $P, a_1 P, a_2 P, \dots, a_n P \in G_1$ for some random choices $a_1, a_2, \dots, a_n \in Z_p$. As in Section 2, solving the MDH problem is to compute $e_m(P, P, \dots, P)^{a_1 a_2 \dots a_m}$ with the non-negligible probability ε .

Without loss of generality, we assume that t-1 participants $U_1, U_2, ..., U_{t-1}$ are able to pool their valid shares and recover the secret.

Now we need to prove that adversary Λ can compute $e_m(P, P, \dots, P)^{a_1 a_2 \dots a_m}$ by using t-1 participants as oracle. In the following, we will set up the system to simulate PVSS for adversary Λ such that this system enables the adversary Λ to compute $e_m(P, P, \dots, P)^{a_1 a_2 \dots a_m}$ when t-1 participants are seen as oracle. The Setup system consists of six steps as follows:

- 1) Adversary Λ sets $P_{pub}^{(i)} = a_i P, C_0 = bP(=f(0)P)$ for i = 1, 2, ..., n, where $a_i \in Z_q^*, b \in Z_q^*$.
- 2) Taking t-1 values: The values f(1), f(2), ..., f(t-1) are chosen at random from Z_q^* , and previous fixed f(0) such that a polynomial f(x) can be fixed.
- 3) Adversary Λ compute forward t-1 values of X_i and Y_i as follows: $X_i = f(i)P$, $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \dots, P_{pub}^{(m)})^{f(i)},$ i = 1, 2, ..., t - 1.
- 4) f(0) is hiding fixed, so Λ is not able to compute the following values: $f(t), f(t+1), \dots, f(n)$. However, we can use X_i for $i = 1, 2, \dots, t-1$ to obtain C_j by solving t-1 simultaneous equations $X_i = \sum_{j=0}^{t-1} (i^j) \cdot C_j$ for $j = 1, 2, \dots, t-1$. When we have computed these values C_j , we can obtain X_i for $i = t, t-1, \dots, n$ by Lagrange interpolation formula.
- 5) First compute $C_j(i = 1, ..., t 1)$. Since $f(x) = \sum_{i=1}^{t-1} a_i \cdot x^i$, then there is the following linear system of equations:

$$\begin{cases} f(0) = a_0 \\ f(1) = a_0 + a_1 \cdot 1 + \dots + a_{t-1} \cdot 1^{t-1} \\ \vdots \\ f(t-1) = a_0 + a_1 \cdot (t-1)^1 + \dots + a_{t-1} \cdot (t-1)^{t-1} \end{cases}$$

In this linear system of equations, adversary Λ knows the values of $f(1), f(2), \dots, f(t-1)$, while f(0) is unknown, so it is unable to compute the coefficient a_i of the polynomial f(x). However, adversary Λ can compute values of C_j by the linear system of equations and public values C_0, X_j for $i = 0, 1, \dots, t-1, j =$ $1, 2, \dots, t-1$. 6) Now, adversary Λ computes the public keys P_i TG_{mul} : The time of executing a scalar multiplication opof participants U_i as $P_i = v_i \cdot P_{pub}^{(i)}$ for i = 0, 1, ..., n, where $v_i \in Z_q^*$. In particular, we set $Y_i = e_m(X_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(m)})^{a_i \cdot v_i}$ such that $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \dots, P_{pub}^{(m)})^{f(i)}$, as

Now, the complete view for the system is defined. Which is consistent with the private view of participants U_1, U_2, \dots, U_{t-1} , and the view comes from the right distribution. Supposing that they are able to obtain the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)}$. Since $P_{pub}^{(i)} = a_i P$ and f(0) = b for $i = 0, 1, \dots, n$, we can compute $e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)} = e_m(P, P, \cdots, P)^{a_1 a_2 \cdots a_m b}.$ It contradicts the MDH assumption.

So far we have ignored the proofs that are required at several points in the protocol. However, in the random oracle model, these proofs can easily be simulated. By the above two lemmas, we can draw the following theorem.

Theorem 1. Under the MDH assumption, the proposed scheme is a secure PVSS scheme in the random oracle model. That is, (1) only qualified participants can compute the valid shares; (2) any subset of t-1 participants is unable to recover the secret. (3) The proposed PVSS scheme must provide publicly verifiable property.

- 1) From Lemma 3 and the scheme's construction method in Section 3, we know that $Y_i = S_i^{d_i}$, then any attacker is unable to compute the corresponding shares S_i from these specific values Y_i because of the hardness of the MDH and discrete logarithm.
- 2) By Lemma 4, any t participants with shares S_i can obtain the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$ by Lagrange interpolation method. And any subset of t-1 or less participants is unable to recover the secret unless the MDH problem is solved.
- 3) From Section 3, it is easy to know that anyone not just the participants can verify each Y_i whether it is equal to $e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{d_if(i)}$ with the dealer's secret f(i) for $i = 0, 1, \dots, n$. In this section, we also have verified that each qualified participant U_i can use his/her private key d_i to compute the share $S_i = Y_i^{d_i^{-1}} = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$. Each S_i also contains the factor f(i). So the proposed scheme must provide publicly verifiable property.

4.3**Performance Analysis**

In this subsection, we mainly analyze the computation overhead and communication overhead. The performance analysis shows that our scheme is effective when comparing with previous schemes. For convenience to evaluate the computational cost, we define the following notations:

 T_{e_m} : The time of executing a multiple linear pairing operation $e_m: G_1^m \to G_2$.

- eration of points in G_1 .
- T_{exp} : The time of executing a modular exponent operation of points in Z_a .
- T_{mul} : The time of executing a modular multiplication operation of points in Z_q .
- T_{Lag} : The time of using the Lagrange interpolating method to construct the secret.
- T_{pol} : The time of computing the polynomial value f(x) = $\sum_{i=0}^{t-1} a_i x^i \text{ in } Z_q.$
 - 1) From the computation aspect. As we all know, the most time consuming is power modular operation in the scheme based on Discrete Logarithm. The most time consuming is a modular multiplication operation of points in the scheme based on ECDLP. While the most time consuming mainly contains T_{e_m} , TG_{mul} in the scheme based on multiple linear pairing.

Hence, we only consider these time-consuming operations T_{e_m} , TG_{mul} and T_{exp} in the performance analysis of the proposed PVSS scheme. In our scheme, there is no need for the dealer to compute the corresponding shares for the participants, compared with the references [23] and [14], our scheme solves the overhead at the secret distribution phase. Especially in verification phase of the shares, we use the tool of multiple linear paring and the batch verification technique to reduce the computational overhead. In Table 1, we list the performance comparison, which are concentrated on the publicly verifiability and computation cost of all phases in secret sharing schemes.

From Table 1, we know that the computation overhead of our scheme is lower in share verification phase, and the main operation cost is a linear relationship with the number of participants. In addition, some calculations can be done preprocessing in secret distribution phase, which can greatly improve the efficiency of secret distribution.

2) From the communication aspect. Since there is no need for our scheme to implement interactive protocol to prevent malicious players, which greatly saves the communication overhead. The communication complexity of our scheme is lower than PVSS in [19]. The communication of the proposed scheme mainly reflects in secret distribution phase and reconstruction phase. Namely the process of the dealer distributes the secret and publishes the public information at secret sharing phase, as well as the overhead of t shareholders pool shares to the secret restorer. Other phases do not need interaction between participants. Consequently, the total communication cost of our scheme is 4nq + tq, which is almost the

Authors'	Publicly	The computation cost of all phases		
schemes	verifiable	Distribution phase	Verification phase	Construction phase
<i>Tian et al. [22]</i>	No	$\begin{array}{c} t(n+1)TG_e + nT_{pol} \\ + 2tTG_{mul} \end{array}$	$TG_e + TG_{mul} + tT_{exp}$	$T_{Lag} + tT_{mul}$
Wu et al. [25]	Yes	$TG_e + (4n+t)TG_{mul} + nT_{exp} + nT_{pol}$	$\begin{array}{l} (n+3)TG_e + n(t+1)TG_{mul} \\ + nT_{\exp} + ntT_{pol} \end{array}$	$T_{Lag} + tT_{\exp}$
<i>Tian et al. [24]</i>	Yes	$nTG_e + 2nTG_{mul} + nT_{mul} + nT_{pol}$	$nTG_e + nTG_{mul} + nT_{exp} + ntT_{pol}$	$T_{Lag} + tT_{mul}$
Our PVSS	Yes	$\frac{(2n+1)TG_e + 3nTG_{mul}}{+nT_{\exp} + nT_{pol}}$	$2nTG_e + nT_{mul}$	$T_{Lag} + tT_{\exp}$

Table 1: Performance comparison

same with reference [25]. Moreover, using multiple linear paring and the technique of batch verification, our communication overhead has great advantage at the share verification phase. So the proposed scheme is less communication overhead and more effective.

5 Discussion

In this section, the application of our publicly verifiable secret sharing scheme is presented in electronic voting. By using our PVSS scheme as a basic tool, we get a simple and efficient voting scheme. At last, we analyze the advantages of this electronic voting scheme.

From the model for universally verifiable elections as introduced by Hwang et al. [15], it is easily to know that all of the players will post their messages in electronic voting schemes. We assume that the players are composed by a set of tallying authorities (tallliers) T_1, \dots, T_n , which act as the participants in our PVSS scheme, a set of Voters V_1, \dots, V_l , and each of them acts as a dealer in our PVSS scheme, as well as a set of passive observers. These sets need not be disjoint, each player may be both a voter and a tallier. Assuming that each tallier T_i has registered a public key $P_i = a_i P_{pub}^{(i)}$ for the randomly selected private key $a_i \in RZ_q$, where $i = 1, 2, \dots, n$.

The designed electronic voting scheme consists of two phases: Ballot casting and Tallying.

1) **Ballot casting.** A voter V casts a vote $v \in \{0, 1\}$ by running the distribution protocol for our PVSS scheme from Section 3, using a random secret value $a \in {}_{R}Z_{q}$, the voter can compute the value $U = e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{a+v}$. In addition, the voter constructs a proof $PROOF_{U}$ showing that indeed $v \in \{0, 1\}$ without revealing any information on v. $PROOF_{U}$ refers to the commitment value of $C_{0} = a_{0}P = aP$ which is published as part of the PVSS distribution protocol. And then each voter proves that: $e_{m}(P_{i}, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)}) = e_{m}(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{a_{i}+v}$.

Due to the publicly verifiability of the proposed

PVSS scheme and the known value of $PROOF_U$, the ballots can be checked by using the above equation by the bulletin board when the voters submit their ballots. What's more, the ballot for voter V consists of the output values U and $PROOF_U$ of the PVSS distribution protocol.

2) **Tallying.** Supposing that voters V_j have all cast valid ballots, where $j = 1, \dots, k$ and $k \leq l$. The tallying protocol uses the reconstruction protocol of our PVSS scheme. We first accumulate all the respective encrypted shares, that is, we compute the values Y_i^* , where

$$Y_{i}^{*} = \prod_{j=1}^{k} Y_{ij}$$

= $e_{m}(P_{i}, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} f_{j}(i)}$.

And then each tallier T_i applies the reconstruction protocol to the value Y_i^* , which will produce

$$e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} f_j(0)}$$

= $e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{a_j}$

Next, by combining with the equation

$$\prod_{j=1}^{k} U_{j}$$

$$= e_{m}(P_{i}, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \dots, P_{pub}^{(m)}) \sum_{j=1}^{k} a_{j} + v_{j}.$$

We obtain

=

$$e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{\sum_{j=1}^k v_j},$$

from which the tally $T = \sum_{j=1}^k v_j, 0 \le T \le k$ can be computed efficiently.

The advantages of this electronic protocol:

- 1) In the ballot casting phase, the voters' ballots contain the votes in encrypted form and the voters need not be anonymous in this protocol. In tallying phase, the talliers use their private keys to collectively compute the final tally corresponding with the accumulation of all the valid ballots.
- 2) The above electronic voting scheme achieves the same level of security with regard to publicly verifiability, privacy, and robustness.
- 3) Our scheme does not require a shared-key generation protocol for a threshold decryption scheme, which avoids the interaction between the voters and the interaction among the talliers.
- 4) Compared with [9], which requires a private channel by public key encryption, our protocol does not need a shared-key generation protocol, so the informationtheoretic privacy for the voters is not lost.

Analysis results show that our PVSS scheme can be used in elections for computational privacy without needing a private channel.

6 Conclusion

In this paper, we proposed a non-interactive, simple and effective publicly verifiable secret sharing based on multiple linear pairing. In our PVSS scheme, not just the participants, anyone is able to verify whether the shares distributed by the dealer are correctly at the secret distribution phase and whether each participant releases valid shares at the reconstruction phase. We use multiple linear property of multilinear map and the batch verification technique to reduce the computational overhead at verification phase. The computation cost and communication overhead are lower than the previous PVSS schemes which are based on bilinear paring or discrete logarithm. In addition, under the multilinear Diffie-Hellman assumption, we have shown our PVSS scheme is security in the random oracle model. In the discussion section, we present the application of our PVSS scheme in electronic voting and analyze the advantages of this protocol. Our next work is to apply the proposed PVSS scheme in secure multi-party computation and other practical protocols.

Acknowledgments

This study was supported by The National Natural Science Foundation of China under Grant (No. 61363068, 61262073), the China Postdoctoral Science Foundation under Grant (No. 2013M530705), the National Natural Science Foundation of Guizhou under Grant (No. 20132112), the Doctor Foundation of the Guizhou University under Grant (No. 2012-024). The authors would like to thank the anonymous reviewers for their helpful comments.

References

- C. Bhagvati, "CRT based threshold multi secret sharing scheme," *International Journal of Network* Security, vol. 16, no. 4, pp. 249–255, 2014.
- [2] G. R. Blakley et al., "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, vol. 48, pp. 313–317, 1979.
- [3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology (Crypto'01), pp. 213–229. Springer, 2001.
- [4] D. Boneh and A. Silverberg, "Applications of multilinear forms to cryptography," *Contemporary Mathematics*, vol. 324, no. 1, pp. 71–90, 2003.
- [5] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," in Advances in Cryptology (Eurocrypt'92), pp. 390–407, Springer, 1993.
- [6] H. Y. Chien, J. K. Jan and Y. M. Tseng, "A 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.
- [7] H. Y. Chien, J. A. N. Jinn-Ke and Y. M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 12, pp. 2762–2765, 2000.
- [8] J. S. Coron, T. Lepoint and M. Tibouchi, "Practical multilinear maps over the integers," in Advances in Cryptology (Crypto'13), pp. 476–493, Springer, 2013.
- [9] R. Cramer, M. Franklin, B. Schoenmakers and M. Yung, "Multi-authority secret-ballot elections with linear work," in *Advances in Cryptology (Eurocrypt'96)*, pp. 72–83, Springer, 1996.
- [10] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.
- [11] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in 28th IEEE Annual Symposium on Foundations of Computer Science, pp. 427–438, 1987.
- [12] S. Garg, C. Gentry and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Eurocrypt*, vol. 7881, pp. 1–17, Springer, 2013.
- [13] S. Garg, C. Gentry, S. Halevi, A. Sahai and B. Waters, "Attribute-based encryption for circuits from multilinear maps," in *Advances in Cryptology* (Crypto'13), pp. 479–499, Springer, 2013.
- [14] L. Harn, "Efficient sharing (broadcasting) of multiple secrets," *IEE Proceedings-Computers and Digital Techniques*, vol. 142, no. 3, pp. 237–240, 1995.
- [15] C. T. Li, M. S. Hwang and Y. C. Lai, "A verifiable electronic voting scheme over the internet," in *IEEE* Sixth International Conference on Information Technology: New Generations, pp. 449–454, 2009.
- [16] A. J. Menezes, T. Okamoto, S. Vanstone, et al., "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transactions on Information The*ory, vol. 39, no. 5, pp. 1639–1646, 1993.

- [17] J. Pieprzyk and X. M. Zhang, "Ideal secret sharing schemes from permutations," *International Journal* of Network Security, vol. 2, no. 3, pp. 238–244, 2006.
- [18] M. Rückert and D. Schröder, "Aggregate and verifiably encrypted signatures from multilinear maps without random oracles," in *Advances in Information Security and Assurance*, pp. 750–759, Springer, 2009.
- [19] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology (Crypto'99)*, pp. 148–164, Springer, 1999.
- [20] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.
- [21] M. Stadler, "Publicly verifiable secret sharing," in Advances in Cryptology (Eurocrypt'96), pp. 190–199, Springer, 1996.
- [22] Y. L. Tian, J. F. Ma, C. G. Peng and X. Chen, "Information-theoretic secure verifiable secret sharing scheme on elliptic curve group," *Journal on Communications*, vol. 12, pp. 014, 2011.
- [23] Y. Tian and C. Peng, "Verifiable secret sharing scheme and applications based on bilinear pairing," *Computer Engineering*, vol. 35, no. 10, pp. 158–161, 2009.

- [24] Y. Tian, C. Peng and J. Ma, "Publicly verifiable secret sharing schemes using bilinear pairings," *International Journal Network Security*, vol. 14, no. 3, pp. 142–148, 2012.
- [25] T. Y. Wu and Y. M. Tseng, "A pairing-based publicly verifiable secret sharing scheme," *Journal of Systems Science and Complexity*, vol. 24, no. 1, pp. 186–194, 2011.
- [26] J. Yu, F. Kong and R. Hao, "Publicly verifiable secret sharing with enrollment ability," in *IEEE Eighth* ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 3, pp. 194–199, 2007.

Qiao Peng born in 1991, M.S candidate. Her research interests include information security and cryptography.

Youliang Tian born in 1982, the corresponding author, PH.D, associate professor and M.Sc supervisor. Now he serves as deputy director of the institute of information and data security of guizhou university. And his research interests focus on game theory, information security and cryptography.