Publicly Verifiable Secret Sharing Schemes Using Bilinear Pairings

Youliang Tian[1,2], Changgen Peng[2], and Jianfeng Ma[1]
(Corresponding author: Changgen Peng)

Key Laboratory of Computer Networks and Information Security[1]
Ministry of Education, Xidian University, Xi'an 710071, China
(Email: youliangtian@163.com)
College of Science, Guizhou University, Guiyang 550025, China[2]
(Received Sep. 4, 2010; revised and accepted DEc. 26, 2010)

Abstract

A non-interactive, simple and efficient publicly verifiable secret sharing (PVSS) is constructed based on the bilinear pairing on elliptic curves, which has all advantages of Schoenmakers' PVSS in [15]. Moreover, in the scheme's distribution of shares phase, only using bilinearity of bilinear paring, anybody can verify that the participants received whether correct shares without implementing interactive or the non-interactive protocol and without construction so called witness of shares applying Fiat-Shamir's technique. Subsequently, in the scheme's reconstruction of secret phase, the released shares may be verified by anybody with the same method. Since the PVSS need not to implement non-interactive protocol and construct witness in order to prevent malicious players, hence it reduces the overhead of communication. Finally, the PVSS has been extensions to the case without a dealer (or without a trusted center). A distributive publicly verifiable secret sharing (DPVSS) is proposed, which also reduces the overhead of communication. Analysis shows that these schemes are more secure and effective than others, and it can be more applicable in special situation.

Keywords: Bilinear pairing, cryptography, Diffie-Hellman assumption, publicly verifiable secret sharing, secret sharing

1 Introduction

Secret sharing schemes were introduced independently in [16] and [1] and since then much work has been put into the investigation of such schemes. In a Secret Sharing scheme, the dealer shares a secret among n participants such that only specified subsets of the whole participants' can later recover the secret. In the so called (k, n) threshold model for secret sharing, the sharing is done so that subsets of k or more participants can later reconstruct the secret, while subsets of at most $k-1$ participants have no information about it. The basic model for secret sharing distinguishes at least two protocols: (i) a distribution protocol in which the secret is distributed by a dealer among the participants, and (ii) a reconstruction protocol in which the secret is recovered by pooling the shares of a qualified subset of the participants. In the basic scheme (e.g., [1, 16] for threshold secret sharing) we assumed that the dealer and all participants is reliable, however, a misbehaving dealer or participants can deal inconsistent shares to the participants, from which they will not be able to reconstruct a secret. To prevent such malicious behavior of the dealer and players, one needs to implement a protocol through which a consistent dealing can be verified by the recipients of shares. Thus basic schemes solve the problem for the case that all players in the scheme are honest.

In verifiable secret sharing (VSS) [6, 7, 13] the object is to resist malicious players, such as (i) a dealer sending incorrect shares to some or all of the participants, and (ii) participants submitting incorrect shares during the reconstruction protocol. Verifiable secret sharing or the basic schemes such as [1, 14, 16] all require the availability of private channels from the dealer to each of the participants individually. However, communication over the private channels is clearly not publicly verifiable. In publicly verifiable secret sharing (PVSS), as introduced by Stadler [17], it is an explicit goal that not just the participants can verify their own shares, but that anybody can verify that the participants received correct shares. It is explicitly required that can be verified publicly. In fact, the VSS scheme of [6] already achieved this property. Hence, publicly verifiable secret sharing (PVSS) is a special kind of secret sharing, in which any-body, not just the participants, can verify whether the dealer distributed correct to each participants at the secret distribution phase and whether each participant releases the correct share at reconstruction phase. Furthermore, in an efficient PVSS, private channels are not assumed between the dealer and the participant. In the reference [10],

Fujisaki and Okamoto present a practical and provably secure PVSS scheme. At Crypto'99, Schoenmakers in [15] proposed a simple PVSS scheme based on the discrete log problem and its application to electronic voting, escrow cryptosystems, etc. In the reference [19], the security of Schoenmakers' scheme is proved theoretically; and a distributed publicly verifiable secret sharing scheme (without a dealer or trusted center) is proposed.

Earlier bilinear pairings, namely Weil pairing and Tate pairing of algebraic curves were used in cryptography for the MOV attack [12] using Weil pairing and FR attack [9] using Tate pairing. These attacks reduce the discrete logarithm problem on some elliptic or hyperelliptic curves to the discrete logarithm problem in a finite field. In recent years, bilinear pairings have found positive application in cryptography to construct new cryptographic primitives. Joux [11], in 2000, showed that the Weil pairing can be used for "good" by using it in a protocol to construct three-party one-round Diffie-Hellman key agreement. This was one of the breakthroughs in key agreement protocols. After this, Boneh and Franklin [2] presented in Crypto 2001 an IDbased encryption scheme based on properties of bilinear pairings on elliptic curves which is the first fully functional, efficient and provably secure identity-based encryption scheme. In Asiacrypt 2001, Boneh, Lynn and Shacham proposed a basic signature scheme using pairing, the BLS [3] scheme, that has the shortest length among signature schemes in classical cryptography. Subsequently numerous cryptographic schemes based on BLS signature scheme were proposed. Apart from the three fundamental cryptographic primitives: encryption, signature and key agreement, there are protocol designs for signcryption, threshold decryption, key sharing, identification scheme, chameleon hashes etc.

Although the bilinear pairing is an important tool for construction encryption and signature algorithms, it is almost not the secret sharing scheme based on bilinear parings. Consequently, it is extremely necessary and significant to construct the secret sharing scheme based on bilinear pairing on elliptic curves. In this paper, a non-interactive, simple and efficient PVSS is constructed based on the bilinear pairing on elliptic curves. In the scheme's distribution of shares phase, using bilinearity of bilinear paring on elliptic curves, anybody can verify that the participants received whether correct shares without implementing the non-interactive protocol such as $DLEQ(g_1, h_1; g_2, h_2)$ by Chaum and Pedersen in [5] and without construction so called witness of shares applying Fiat-Shamir's technique in [8]. Subsequently, in the scheme's reconstruction of secret phase, the released shares may be verified by anybody with the same method. Compared to [10, 15, 17], consequently, this scheme is simpler and more efficient.

Summarizing, our PVSS construction will be much simpler and more efficient than the above approaches, and since the PVSS need not to implement zero-knowledge proofs or non-interactive protocol and construct witness in order to prevent malicious players, the complexity of communication is lower than the above approaches. We only need techniques that work in any group for which the discrete log problem is intractable and there exist bilinear pairing in any group. The protocols consist of a few steps only, relying on simple primitives. The performance is not only asymptotically optimal, but also good in practice. And, finally, we are able to use this to construct the election scheme and other applications like [15].

The rest of the paper os organized in the following manner. In Section 2, we will describe the definition of bilinear pairings, the Diffie-Hellman assumption and bilinear Diffie-Hellman problem. In Section 3, we then present our main construction of a simple and efficient PVSS scheme based on bilinear pairings and proof the scheme security. In Section 4, we consider extensions to the case without a dealer (or without a trusted center). In Section 5, we analysis these performance and the comparison results. In Section 6, we introduce main conclusion in this paper and our next work.

2 Preliminaries

2.1 Bilinear Pairing

Definition 1. *Let G_1, G_2 be two groups of the same prime order q. We view G_1 as an additive group and G_2 as a multiplicative group. Let P be an arbitrary generator of G_1. (aP denotes P added to itself a times). Assume that discrete logarithm problem (DLP) is hard in both G_1 and G_2. A mapping $e : G_1 \times G_1 \to G_2$ satisfying the following properties is called a cryptographic bilinear map.*

Bilinearity. $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_a^*$. This can be restated in the following way. For $P, Q, R \in G_1, e(P + Q, R) = e(P, R)e(Q, R)$ and

$$e(P, Q + R) \quad = \quad e(P, R)e(Q, R).$$

Non-degeneracy. If P is a generator of G_1, then $e(P, P)$ is a generator of G_2. In other words, $e(P, P) \neq 1$.

Computable. There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

2.2 The Classical Diffie-Hellman Problems

The security of many modern cryptosystems is based on trap-door functions which are easy to compute in one way, but hard to invert except if we know a secret key. One of the main trap function was introduced by Diffie and Hellman in 1976.

The Diffie-Hellman problem is defined as follows: consider a cyclic additive group $G =< g >$ (which means that g is a generator of G or in other words that the elements of G are exactly the successive multiples of g). We take two elements of G namely $g_1 := a \cdot g$ and $g_2 := b \cdot g$, and

we suppose we know the generator g but we do not know a and b. The problem is then to compute $g_3 = (ab) \cdot g$. This problem is tightly linked with the Discrete Logarithm problem (DLP) which is defined below.

Definition 2. *Let us define formally the problems we are dealing with:*

1) *Let G be a finite cyclic group and let g be a generator of G. The discrete logarithm problem (DLP) in G is as follows: Given $(g, a \cdot g)$ with uniformly random choice of $a \in Z^*_{|G|}$, find a.*

2) *Let G be a finite cyclic group and let g be a generator of G. The computational Diffie-Hellman problem (CDHP) in G is as follows: Given $(a, a \cdot g, b \cdot g)$ with uniformly random choice of $a, b \in Z^*_{|G|}$, compute $(ab) \cdot g$.*

3) *Let G be a finite cyclic group and let g be a generator of G. The decisional Dif-fie-Hellman problem (DDHP) in G is as follows: Given $(g, a \cdot g, b \cdot g, c \cdot g)$ with uniformly random choice of $a, b \in Z^*_{|G|}$, decide $(ab) \cdot g = c \cdot g$.*

It is clear that the CDHP can easily be solved if the DLP can be solved. If the DLP can be solved, we can indeed find a from $a \cdot g$ and then we compute $(ab) \cdot g$ as $a \cdot (b \cdot g)$. We then say that $DLP \rightarrow CDHP$ but the reciprocity is not true. Yet, as far as we know, solving the DLP is the only known method to solve the CDHP, and for this reason the CDHP is believed to be as hard as the DLP, which is usually of exponential difficulty.

Concerning the DDHP it is another story. Suppose indeed that we can compute a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ (like the Weil pairing for example). We want to confirm that $cP = abP$ for the tuple (P, aP, bP, cP) where $a, b, c \in Z^*_q$. Then we just have to verify that the equality $e(aP, bP) = e(P, cP)$ holds: if it does then, in the tuple (P, aP, bP, cP), we must have $cP = abP$, and if does not then we know that $cP \neq abP$. When we can compute a map e the DDHP is easy, since it can be solved in polynomial time (if e is computed in polynomial time), and it is not linked to the difficulty of CDHP.

Groups for which the DDHP is easy and the CDHP is hard are called gap groups.

2.3 Bilinear Diffie-Hellman Problem

New applications like ID-based encryption base their security on a bilinear version of the Diffie-Hellman problems. The bilinear Diffie-Hellman definitions presented here, were first formally presented in [4]. We con-sider G_1 an additive group of prime order q, and P a generator of this group. We also consider a multiplicative group G_2 and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ which can be computed efficiently (in polynomial time).

Definition 3. *The bilinear Diffie-Hellman problem (BDHP) in (G_1, G_2, e) is as follows: Given (P, aP, bP, cP) with uniformly random choices of $a, b, c \in Z^*_q$, compute $e(P, P)^{abc} \in G_2$.*

An algorithm is said to have advantage ε in solving the BDHP in (G_1, G_2, e).

$$\text{If } Pr[A(P, aP, bP, cP) = e(P, P)^{abc}] \geq \varepsilon.$$

Here the probability is measured over the random choices of $a, b, c \in Z^*_q$, $P \in G_1$.

The BDH assumption states that no probabilistic polynomial time algorithm has non-negligible advantage (in k) in solving the BDHP for (G_1, G_2, e).

The BDHP for parameters (G_1, G_2, e) is no more difficult to solve than the CDHP in either G_1 or G_2. Given $(P, aP, bP, cP) \in G_1^4$, there exist two ways for solving the BDHP using the CDHP:

1) By solving the CDHP on (P, aP, bP) in G_1, we can find $abP \in G_1$. Given abP we then can compute $e(abP, cP) = e(P, P)^{abc} \in G_2$ which is the solution to the BDHP;

2) By solving the CDHP on $(e(P, P), e(aP, P))$, $e(bP, cP)$ (which is the same as $(e(P, P), e(P, P)^a)$, $e(P, P)^{bc}$ in G_2, we get $e(P, P)^{abc}$ which is the solution to the BDHP.

The conclusion of this is clear: if the BDHP is no harder than the CDHP which is no harder than the DLP, we should better ensure that this last one is really hard.

3 Special PVSS Scheme

Let G_1 denote an additive group of prime order q, P and Q denote independently selected generator of this group, hence no party knows the discrete log of P with respect to Q. At the same time, we also consider a multiplicative group G_2 of order q and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ which can be computed efficiently (in polynomial time). Moreover, the computing discrete logarithms in G_1 and G_2 are infeasible. The dealer will achieve this by first selecting $s \in_R Z_q$ and then distributing shares of secret $S = sQ$.

3.1 Protocols

The PVSS Scheme consists of three phases: Initialization, Distribution and Reconstruction.

Initialization.

The group G_1 and G_2 denote an additive group and multiplicative group of prime order q, among G_1 and G_2 exist a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, and the independently generators P, Q are selected using appropriate public procedure. Participants P_i generates a private key $x_i \in_R Z^*_q$ and registers $y_i = x_i Q$ as its public key.

Distribution.

The protocol consists of two steps:

1) Distribution of the shares. The dealer wishes to distribute a secret among participants $P_1, ldots, P_n$. The dealer picks a random polynomial f of degree at most $t-1$ with coefficients in Z_q : $f(x)\sum_{j=0}^{t-1} a_j x^j$, and sets $s = a_0$. The dealer keeps this polynomial secret but publishes the related commitments $c_j = a_j P$, for $0 \le j < t$. The dealer also publishes the encrypted shares $Y_i = f(i) \cdot y_i$, for $1 \le i \le n$.

2) Verification of the shares. Verifier computes $x_i = \sum_{i=0}^{t-1} i^j \cdot C_j$ from the C_j values. The verifier accepts if $e(P, Y) = e(X_i, y_i)$ and rejects otherwise.

Reconstruction.

The protocol consists of two steps:

1) Decryption of the shares and its verification. Using its private key x_i, each participants finds the shares $S_i = f(i) \cdot Q$ from Y_i by computing $S_i = x_i^{-1} \cdot y_i$. The correctness of the shares is easy to verify since $e(Y_i, Q) = e(y_i, S_i)$.

2) Pooling the shares. Participants Pi produce correct values for S_i, for $i = 1, 2, \cdots, t$. The secret $S = sQ$ is obtained by Lagrange interpolation:

$$\begin{aligned}\sum_{i=1}^{t} \lambda_i \cdot S_i &= \sum_{i=1}^{t} \lambda_i \cdot (f(i) \cdot Q) \\ &= \sum_{i=1}^{t} (\lambda_i f(i)) \cdot Q \\ &= f(0) \cdot Q = sQ,\end{aligned}$$

where $\lambda_i = \Pi_{j \ne i} \frac{j}{j-i}$ is a Lagrange coefficient.

3.2 Security

We first consider the security of the share-encryptions. We observe that directly breaking the encryption used in special PVSS scheme implies breaking the Diffie-Hellman assumption. Consequently, we have the following lemma.

Lemma 1. *The encryption used in special PVSS scheme is security if and only if the Diffie-Hellman assumption holds.*

Proof. (reduction to absurdity) Let the encryption of shares is security, but the Diffie-Hellman assumption not holds. Since the Diffie-Hellman not holds, then there exists a algorithm A: for two random elements $\alpha P, \beta P \in G_1$ $(\alpha, \beta \in Z_q^*)$, the algorithm output $\alpha\beta P$ with some success probability ε. Now we need to prove that breaks the encryption of shares with the same as success probability ε. Breaking the encryption of shares amounts to finding $S_i = f(i)Q$ given P, Q, X_i, Y_i, y_i, for the group G_1. To prove this, suppose $Q = \alpha P, \alpha \in_R Z_q^*$, and pick random elements $\alpha', \beta' \in_R Z_q^*$, subsequently $\alpha' Q$ $(= \alpha'\alpha P)$, $\beta' X_i$ $(= \beta f(i) P)$, as input of and run A. Then output with success probability ε. Thus we can obtain the value of S_i:

$$S_i = \alpha'^{-1}\beta'^{-1} (\alpha\alpha'\beta' f(i) P) (= \alpha f(i) P = f(i) Q).$$

It shows that the encryption of shares is security the Diffie-Hellman assumption must holds.

Suppose the Diffie-Hellman holds, then the encryption of shares is no security. Then there exists a algorithm: when any random elements $P, Q, X_i, Y_i, y_i \in_R G_1$ as input of the algorithm B, it can output $S_i (= f(i)Q)$ with some success probability ε. Writing $Q = \alpha P$, $X_i = \beta P$ and $y_i = \gamma P$.

Given $P, \alpha P, \beta P, \gamma P, \beta\gamma P$, running B can output $\alpha\beta P$ with the success probability ε. Now we need to prove that given $\alpha P, \beta P$, then $z = \alpha\beta P$ can be computed by B with the success probability ε. To prove this, select random $\alpha', \beta', \gamma \in_R Z_q^*$, and compute $P, \alpha\alpha' P, \beta\beta' P, \gamma P$, as input of B, running has computed $\alpha\alpha'\beta\beta' P$. Then we can obtain the value of z:

$$z = \alpha'^{-1}\beta'^{-1}(\alpha\alpha' P \beta\beta' P) (= \alpha\beta P).$$

It shows that the Diffie-Hellman holds $\Rightarrow$ then the encryption of shares is security. □

A strong result is that the secret can be reconstructed by at least $t-1$ participants. This is expressed by the following lemma.

Lemma 2. *If the Diffie-Hellman assumption holds, then that $t-1$ participants pool their shares and cannot obtain the secret.*

Proof. (reduction to absurdity) If $t-1$ participants pool their shares, the secret can be recovered. Suppose without loss of generality that participants $P_1, P_2, \cdots, P_{t-1}$ are able to pool their and recover the secret. Now need to prove that let be given, adversary II can compute with using $t-1$ participants as Oracle. Here are random; if not, it is trivial to adapt the proof, as in the previous lemma.

Now we will set up the system to simulate PVSS for adversary II such that this fact enables adversary II to compute with oracle. The Setup of system consists of six steps:

1) Adversary II sets $Q = \alpha P$, $C_0 = \beta P (= f(0)P)$.

2) Taking $t-1$ values: $f(1), \cdots, f(t-1) \in_R Z_q^*$; and previous fixed $f(0)$ such that function $f(x)$ can be decided.

3) Adversary II compute forward $t-1$ values of X_i and $Y - i$:

$$X_i = f(i) \cdot P, Y_i = f(i) \cdot y_i, i = 1, 2, \cdots, t-1.$$

4) $f(0)$ is hiding fixed, thus II is not able to compute the following values: $f(t), f(t+1), \cdots, f(n)$, but making use of Lagrange interpolation formula II can compute values of $X_t, \cdots, X_n$.

5) Compute $C_j(i = 1, \cdots, t-1)$. Since $f(x) = \sum_{i=n}^{t-1}$, then there is the following linear system of equations:

$$\begin{bmatrix} 1 & 0^1 & \cdots & 0^{t-1} \\ 1 & 1^1 & \cdots & 1^{t-1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & (t-1)^1 & \cdots & (t-1)^{t-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{bmatrix} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(t-1) \end{bmatrix}$$

In this linear system of equations, adversary II knows values of $(f(1), \cdots, f(t-1))$, but II do not know the value of $f(0)$. As a result, II cannot have solved values of $(a_0, a_1, \cdots, a_{t-1})$. However, so coefficient matrix of the linear system of equations is Van der Monde matrix, it have inverse matrix.

Then it is obvious that II can compute values of C_j $(= 1, \cdots, t-1)$ by the linear system of equations and values of $C_0, X_i(i = 1, \cdots, t-1)$.

6) Now, adversary II random picks up $\omega_i \in_R Z_q^*$ $(i = t, \cdots, n)$, sets $y_i = \omega_i P$, $Y_i = \omega_i X_i$, $(i = t, \cdots, n)$, then Y_i satisfied $Y_i = f(i) \cdot y_i$.

□

The complete view for the system is now defined. It is consistent with the private view of participants $P_1, \cdots, P_{t-1}$, and comes from the right distribution. Now, suppose that they are able to compute the secret $f(0) \cdot Q$. Since $Q = \alpha P$ and $f(0) = \beta$, we are thus able to compute $(\alpha\alpha) \cdot P$. This contradicts the Diffie-Hellman assumption.

Making use of above two lemmas, it is easy that this lead to follow theorem.

Theorem 1. *Under the Diffie-Hellman assumption, the special PVSS scheme is secure in the random oracle model. That is, (i) the reconstruction protocol results in the secret distributed by the dealer for any t participants, (ii) any $t-1$ participants is not able to recover the secret.*

4 Distributed PVSS Scheme

In the above section, special PVSS scheme is homomorphic. Thus we will consider extensions to the case without a dealer (or without a trusted center), which is called distributed publish verifiable secret sharing (DPVSS). The secret S is decided by all players.

For convenience, special PVSS scheme denotes $PVSS(S; C_j; Y_i; S_i; f(x))$, S: sharing secret, C_j and Y_i: publish information, S_i: participants P_i received shares, $f(x)$: the dealer select random function. Then homomorphic of special PVSS scheme shows the following theorem.

Theorem 2. *Given two implementing instances of the PVSS scheme I_1: $PVSS(S; C_j; Y_i; S_i; f(x))$ and I_2: $PVSS(S; C_j; Y_i; S_i; f(x)) \Leftrightarrow$ one implementing instance I_3: $PVSS(S + S; C_j + C_j; Y_i + Y_i; S_i + S_i; f(x) + f(x))$.*

Proof. Its proof is very easy, omit. □

Now we will design DPVSS scheme. Assumption the secret S will distribute among n participants. Other assumption is the same as the above section. In the DPVSS scheme, anybody can verify the shares whether correct or not, and any t participants can recover the secret but any $t-1$ participants are not able to gain any information for the secret.

The DPVSS Scheme consists of three phases: Distribution, Computation of the shares and Reconstruction.

Distribution. Participants P_i implements the protocol $PVSS(S; C_{ij}; Y_{ik}; S_{ik}; f_i(x))$, i.e. P_i pick a random number $S_i \in_R Z_q^*$ and a polynomial $f_i(x) = \sum_{i=n}^{t-1} a_{ij}x^j \in_R Z_q[x]$ such that $f_i(0) = a_{i0} = s_i$ and $S = s_i Q$. At the same time, P_i keeps this polynomial secret but publishes the related commitments $C_{ij} = \alpha_{ij} P$, for $0 \le j < t$. P_i also publishes the encrypted shares $Y_{ik} = f_i(k) \cdot y_i$, for $1 \le k \le n$. Consequently, anybody can verify that all participants implement the protocol whether success.

Computation of the shares. All participants have succeeded in distribution their shares. Then each participants computes encrypted shares Y_i:

$$\begin{aligned} Y_i &= \sum_{i=1}^{n} Y_{ij} (= \sum_{i=1}^{n} f_j(j) \cdot y_i \\ &= (x_i \sum_{i=1}^{n} f_j(i)) \cdot Q). \end{aligned}$$

Reconstruction. The protocol consists of two steps:

1) Decryption of the shares and its verification. Using its private key x_i, each participants finds the shares $S_i == (\sum_{i=1}^{n} f(i)) \cdot Q$ from $Y_i = \sum_{i=1}^{n} Y_{ji}$ by computing $S_i = X_i^{-1} \cdot Y_i$. The correctness of the shares is easy to verify since $e(Y_i, Q) = e(y_i, S_i)$.

2) Pooling the shares. Participants P_i produce correct values for S_i, for $i = 1, 2, \cdots, t$. The secret $S = sQ$ is obtained by Lagrange interpolation:

$$\begin{aligned} \sum_{i=1}^{t} \lambda_i \cdot S_i &= \sum_{i=1}^{t} \lambda_i \cdot (f(i) \cdot Q) \\ &= \sum_{i=1}^{t} (\lambda_i f(i)) \cdot Q = f(0) \cdot Q = sQ, \end{aligned}$$

where $\lambda_i = \prod_{j \ne 1} \frac{j}{j-i}$ is a Lagrange coefficient.

The above distributive protocol denotes $DPVSS$ $(S; C_{ij}; Y_{ik}; S_{ik}; f_i(x))$. Then we have the following theorem:

Theorem 3. *Let $S = \sum_{i=1}^{n} \bar{S}_i$, $S_i = X_i^{-1} \cdot Y_i$, $C_i = \sum_{ki}^{n} C_{ki}$, $Y_i = \sum_{ki}^{n} Y_{ki}$, $f(x) = \sum_{i=1}^{n} f_i(x)$. Then $DPVSS(S; C_{ij}; Y_{ik}; S_{ik}; f_i(x)) \Leftrightarrow PVSS(S; C_i; Y_i; S_k; f(x))$.*

Table 1: Performance comparison

Schemes	Secure channel	Verifiable/Public verifiable	Trusted Center	Forward Secrecy
The scheme of [17]	Not needed	Public verifiable	Required	No
The scheme of [10]	Not needed	Public verifiable	Required	No
The scheme of [15]	Not needed	Public verifiable	Required	No
The scheme of [18]	Required	Verifiable	Required	Yes
The PVSS of the paper	Not needed	Public verifiable	Required	No
The DPVSS of the paper	Not needed	Public verifiable	Not needed	No

5 Performance Analysis

In this section, the performance of the proposed schemes are analyzed and compared with those of previous schemes. Compared to Schoenmakers' PVSS in [15], the PVSS has all advantages of Schoenmakers' PVSS. At the same time, in this PVSS, the dealer only needs to post $t+n$ elements of G_1 (the numbers C_i and Y_i); in Schoenmakers' PVSS, the dealer not only needs to post $t+n$ elements of G_1 (the numbers C_i and Y_i), but also plus $n+1$ number of size $|q|$ (the responses r_i and the challenge c).

The number of exponentiations throughout the protocol is correspondingly lower than B. Schoenmakers' PVSS, and all of these exponentiations are with relatively small exponents from Z_q ($|q| = 160$ bits in practice).

In distribution of shares phase, using bilinearity of bilinear paring on elliptic curves, anybody can verify that the participants received whether correct shares without implementing the non-interactive protocol such as $DLEQ(g_1, h_1; g_2, h_2)$ by Chaum and Pedersen in [5] and without construction so called witness of shares applying Fiat-Shamir's technique in [8]. Subsequently, in reconstruction of secret phase, the released shares may be verified by anybody with the same method.

Since the PVSS need not to implement non-interactive protocol and construct witness in order to prevent malicious players, the complexity of communication is lower than PVSS in [10, 15, 17]. Compared to [10, 15, 17], consequently, this scheme is simpler and more efficient. Clearly, the PVSS is homomorphic. For example, given the dealer's output for secrets Q^{S1} and Q^{S2}, the combined secret Q^{S1+S2} can be obtained by applying the reconstruction protocol to the combined encrypted shares Y_{i1}, Y_{i2}. Thus we are able to use this to construct the election scheme and other applications like [15].

Moreover, the DPVSS also needs not implementing the non-interactive protocol such as zero-knowledge proofs to verify the shares, and also not construct so called witness. It needs only the technology that is bilinearity of the bilinear paring. Thus its efficiency is extremely high. The performance comparison with related schemes is shown in Table 1.

6 Conclusion

The publicly verifiable secret sharing (PVSS) is a special kind of secret sharing, in which anybody, not just the participants, can verify whether the dealer distributed correct shares to each participant at the secret distribution phase and whether each participant releases the correct share at secret reconstruction phase. Furthermore, in an efficient PVSS, private channels are not assumed between the dealer and the participants. The bilinear pairing is an important tool for construction encryption and signature algorithms. In this paper, a non-interactive, simple and efficient PVSS is constructed based on the bilinear pairing on elliptic curves, and extent to the case without a dealer (or without a trusted center), i.e. distributive publicly verifiable secret sharing (DPVSS). Using bilinearity of the bilinear paring anybody can verify the shares whether correct or not without implementing relative non-interactive protocol and without construction so called witness. It is obvious that the efficiency is extremely high. It is very significant that study secure multi-party computation problem based on bilinear parings. This will be our next work.

Acknowledgment

This work was supported by The National Natural Science Foundation of China under Grant No. 60872041, 61072066, 60963023, 60970143, 61100230 and 61100233, the Fundamental Research Funds for the Central Universities under Grant No. JY10000903001 and JY10000901034, the Doctor Foundation of the Guizhou University under Grant No.2007-040. The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] G. R. Blakley, "Safeguarding cryptographic keys," *Proceedings of the National Computer Conference*, vol. 48, pp. 313-317, 1979.

[2] D. Boneh, and M. Franklin, "Identity based encryption from the Weil Pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586-615, 2003. Extended Abstract in Crypto 2001

[3] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil Pairing," *Journal of Cryptology*, vol. 17, pp. 297-319, 2004.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Advances in Cryptology - Crypto '01*, LNCS 2139, pp. 213-229, Spriner-Verlag, 2001.

[5] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," R. A. Rueppel ed., *Advances in Cryptology - Eurocrypt'92 Proceedings*, LNCS 658, pp. 390-407, Springer-Verlag, Berlin, 1993.

[6] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," *Proceedings 26th IEEE Symposium on Foundations of Computer Sciences (FOCS '85)*, pp. 383-395, 1985.

[7] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," *Proceedings 28th IEEE Symposium on Foundations of Computer Science (FOCS '87)*, pp. 427V437, 1987.

[8] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," *Advances in Cryptology - Crypto '86*, LNCS 263, pp. 186-194, Springer-Verlag, 1987.

[9] G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Mathematics of Computation*, vol. 62, pp. 865-874, 1994.

[10] E. Fujisaki and T. Okamoto, "A practical and provably secure scheme for publicly verifiable secret sharing and its applications," *Advances in Cryptology-Eurocrypto '98*, LNCS 1403, pp. 32V46, Springer-Verlag, Berlin, 1998.

[11] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *Proceedings of ANTS 4*, LNCS 1838, pp. 385-394, Spriner-Verlag, 2000.

[12] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transaction of Information Theory*, vol. 39, pp. 1639-1646, 1993.

[13] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Advances in Cryptology - Crypto '91*, LNCS 576, pp. 129-140, Springer-Verlag, Berlin, 1992.

[14] T. P. Pedersen, *Distributed Provers and Verifiable Secret Sharing based on the Discrete Logarithm Problem*, PhD thesis, Aarhus University, Computer Science Department, Aarhus, Denmark, Mar. 1992.

[15] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to Electronic Voting," *Advances in Cryptology - Crypto' 99*, LNCS 1666, pp. 148-164, Springer-Verlag, Berlin, 1999.

[16] A. Shamir, "How to share a secret," *Communications of the ACM*, 1979, vol. 22, no. 11, pp. 612-613.

[17] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology - Eurocrypt '96*, LNCS 1070, pp. 190V199, Springer-Verlag, Berlin, 1996.

[18] 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.

[19] G. Wang, *The Design and Analysis of Threshold Signatures Schemes and Authentication Protocols*, PhD-DDdissertation, Institute of Software, the Chinese Academy of Sciences, China, 2001.

Tian Youliang, born in 1982, Ph.D candidate. His research interests focus on game theory, information security and cryptography.

Peng Changgen, born in 1963, PH.D, professor and M. Sc. supervisor. His current research interests include cryptography and information security.

Ma Jianfeng, born in 1963, PH.D, professor and Ph.D. supervisor. He mainly en-gaged in the research of channel coding, in-formation and network security.