Research Article

Publicly Verifiable Secret Sharing Scheme with Provable Security against Chosen Secret Attacks

Yuanju Gan,1,2 Lihua Wang,3 Licheng Wang,1 Ping Pan,1 and Yixian Yang1

1 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2 School of Information, Guangdong Ocean University, Zhanjiang, Guangdong 524088, China
3 Network Security Research Institute, National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

Correspondence should be addressed to Yuanju Gan; yj.gan@163.com

Received 29 August 2012; Accepted 8 January 2013

Academic Editor: Wensheng Zhang

Copyright © 2013 Yuanju Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Secret sharing is an important aspect of key management in wireless ad hoc and sensor networks. In this paper, we define a new secure model of secret sharing, use the Lagrange interpolation and the bilinear cyclic groups to construct an efficient publicly verifiable secret sharing scheme on the basis of this model, and show that this scheme is provably secure against adaptively chosen secret attacks (CSAs) based on the decisional bilinear Diffie-Hellman (DBDH) problem. We find that this scheme has the following properties: (a) point-to-point secure channels are not required in both the secret distribution phase and the secret reconstruction phase; (b) it is a noninteractive secret sharing system in that the participants need not communicate with each other during subshadow verification; and (c) each participant is able to share many secrets with other participants despite holding only one shadow.

1. Introduction

A secret sharing scheme [1–8] allows the splitting of a secret into different pieces, called shares or shadows, which are given to a group of participants (or shareholders). Only a certain specified subset of the participants can reconstruct the secret easily by providing their shadows, while any unqualified subsets cannot obtain any knowledge about the secret. Secret sharing is useful for any important action whose initiation requires the collective decision of several designated participants, such as the launch of a missile, opening of a bank vault, or opening of a safety deposit box. Research on secret sharing is important for the key distribution of wireless ad hoc and sensor networks [9], both in theory and in practice.

In 1979, two basic secret sharing schemes were independently proposed by Shamir [1] and Blakley [10]. They used two different methods to construct threshold secret sharing schemes. In Shamir’s scheme, a secret $s$ is divided into $n$ shadows by a dealer and shared among $n$ participants in such a way that it is possible to reconstruct the secret with any $t$ or more shadows but impossible to reconstruct the secret with fewer than $t$ shadows. This scheme is called a $(t, n)$ threshold secret sharing scheme. Early secret sharing schemes [1, 10] did not include the verification of the correctness of shadows; hence, if one or more participants are dishonest, the secret cannot be recovered correctly.

Verifiable secret sharing (VSS) was proposed in [11] to solve the problem of dishonest participants who want to deceive other honest participants or the problem of a dishonest dealer who distributes incorrect shadows to some participants. VSS has been an important area of cryptography research for the last two decades [5, 7, 8, 12–15]. Feldman [12] proposed a very practical VSS scheme in which the security is based on a discrete logarithm problem. In this scheme, a deterministic function of the secret is published; hence, it
achieves only one-way security. Pedersen [13] proposed a VSS scheme that can withstand an unbounded passive adversary.

Stadler [16] proposed a publicly verifiable secret sharing (PVSS) scheme in which the validity of the shadows can be verified by anyone without knowledge of the shadows. In some PVSS schemes [5, 14], the verification procedure involves interactive proofs of knowledge. If these proofs are made noninteractive by means of the Fiat-Shamir technique [17], the security of the verification process would only be carried out in the random oracle model (ROM) [18].

Transferring security analysis of cryptographical primitives from the random oracle model to the standard model (SM) [19] has always been a theoretically important task.

1.1. Related Work. In 2005, Ruiz and Villar [15] proposed a new PVSS scheme that has a higher level of secrecy, called indistinguishability (IND) of secrets based on the decisional composite residuosity assumption. In 2009, Heidarvand and Villar [3] gave two new secure definitions of publicly verifiable secret sharing, which capture the notion of indistinguishability of shared secrets. Then they proposed a noninteractive PVSS scheme against the attacks of indistinguishability of secrets in the standard model based on the decisional bilinear square assumption (DBS) which is a natural variant of the standard decisional bilinear Diffie-Hellman (DBDH) assumption. In 2010, Jhanwar [20] proposed a PVSS scheme whose level of security is called semantic security based on the (t,n)-multi-sequence of exponents Diffie-Hellman problem. In 2011, Wu and Tseng [2] proposed a pairing-based PVSS scheme. For deducing the computational cost, they used the batch verification technique. They also showed that their scheme is a secure PVSS scheme under the bilinear Diffie-Hellman (BDH) assumption in the random oracle model. In fact, semantic security does not guarantee any level of secrecy if an adversary mounts an active attack. Therefore, it is very important to design a PVSS scheme against adaptively chosen secret attacks (CSAs) in the standard model.

Another important aspect of secret sharing is the problem of making the size of shadows of each participant as small as of making the size of shadows of each participant as small as. A secret sharing scheme is ideal if the length of every shadow is the same as the length of the secret. This is the best possible situation. However, we would like to emphasize that it is also very important to reduce the number of secure channels used in a secret sharing scheme, especially in wireless ad hoc and sensor networks.

A secret sharing scheme contains at least two essential phases: a share distribution phase and a secret reconstruction phase. In the share distribution phase, a dealer chooses a secret, executes a secret distribution algorithm to generate shadows, and then sends the generated shadows to the participants through point-to-point secure channels. In the secret reconstruction phase, the participants belonging to a qualified subset of participants exchange shadows amongst themselves through point-to-point secure channels to reconstruct the secret. In a (t,n) threshold secret scheme, there are n secure channels in the share distribution phase and at least $O(t^2)$ secure channels in the secret reconstruction phase. To reduce the number of secure channels to $O(t)$, Huaxiong and wong [21] constructed a $(t,n)$ threshold secret sharing scheme using partial broadcast channels.

1.2. Our Contributions. In this paper, we use the Lagrange interpolation and bilinear cyclic groups to construct a $(t,n)$ threshold PVSS scheme with IND-CSA security. Our scheme has the following features.

(i) **Public Verifiability**: a dishonest dealer or participant is detected unconditionally.

(ii) **Security**: the scheme has provable security against an IND-CSA (see the security model present in this paper) adversary in the standard model. The security relies on the hardness of the decisional bilinear Diffie-Hellman (DBDH) problem.

(iii) **Needless security channels**: in both the setup and share distribution phases, these are no secure point-to-point communication channels between the dealer and the participants. Moreover, no secure point-to-point communication channels are used in the reconstruction phase of the extended scheme.

(iv) **Noninteractivity**: the participants need not talk to each other during the secret reconstruction phase.

An overview comparison of the major technique differences and the corresponding security level those of WT11’s [2] and HV09’s [3] PVSS schemes is given in Table 1.

|                      | WT11 [2] | HV09 [3] | Ours          |
|----------------------|----------|----------|---------------|
| Needless secure channels | No       | No       | Yes           |
| Verify subshare in reconstruction | No       | No       | Yes           |
| Assumption | BDH      | DBS      | DBDH          |
| Security model | ROM      | SM       | SM            |
| Security level | IND      | IND      | IND-CSA       |

1.3. Paper Organization. This paper is organized as follows. In Section 2, we describe the definition of bilinear maps and the decisional bilinear Diffie-Hellman problem. In Section 3, we describe the model of our PVSS scheme and the security model. In Section 4, we present our pairing-based PVSS scheme, and in Section 5, we prove its security. In Section 6, we analyze the performance of our scheme. In Section 7, we present an extended scheme that allows reconstruction of the secret through public broadcast channels. Finally, we give a conclusion in Section 8.

2. Preliminaries

If $S$ is a set, $|S|$ denotes its size. The symbol “⊥” denotes failure.

2.1. Bilinear Map. Let $G$ and $G_1$ be two cyclic groups of prime order $p$. Here, we assume that $G$ is an additive cyclic group, and $G_1$ is a multiplicative cyclic group. A bilinear map $e(\cdot,\cdot)$

|                      | WT11 [2] | HV09 [3] | Ours          |
|----------------------|----------|----------|---------------|
| Needless secure channels | No       | No       | Yes           |
| Verify subshare in reconstruction | No       | No       | Yes           |
| Assumption | BDH      | DBS      | DBDH          |
| Security model | ROM      | SM       | SM            |
| Security level | IND      | IND      | IND-CSA       |
is a map $G \times G \to G_1$ such that for $P \neq 0 \in G$ and $\alpha, \beta \in Z_p \setminus \{0\}$, it satisfies the following properties [22, 23].

(i) Bilinearity: $e(\alpha P, \beta P) = e(P, P)^{\alpha \beta}$.
(ii) Nondegeneracy: $e(P, P) \neq 1_{G_1}$, unless $P = 1_G$.
(iii) Computability: there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G$.

The algorithm $\mathcal{BB}(\lambda)$ is a bilinear group generator that takes a secure parameter $\lambda \in Z$ as input and outputs the descriptions of the groups $G, G_1$, and a bilinear map $e: G \times G \to G_1$, where all group operations in $G$ and $G_1$ as well as map $e$ can be computed in polynomial time with $\lambda$. We posit that $B = (p, P, G, G_1, e(\cdot, \cdot))$ is the output of $\mathcal{BB}(\lambda)$.

2.2. Decisional Bilinear Diffie-Hellman Assumption. Given a tuple $P, aP, bP, cP \in G$ for some uniformly chosen $a, b, c \in Z_p \setminus \{0\}$ and $T \in G_1$ as input to decide whether or not $T = e(P, P)^{abc}$. The advantage of an algorithm $\mathcal{A}$ solving the DBDH problem is defined as

\[ \mathcal{Adv}(\mathcal{A}) = \mathcal{Pr}[\mathcal{A}(P, aP, bP, cP, e(P, P)^{abc}) = 1] - \mathcal{Pr}[\mathcal{A}(P, aP, bP, cP, e(P, P)) = 1]. \]

The DBDH problem is said to be $(t, \epsilon)$-hard if there is no algorithm that can solve the DBDH problem within time $t$ with an advantage equal to at least $\epsilon$.

2.3. Lagrange Interpolation. Let $f(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ be a polynomial over $\mathbb{Z}_p$ with degree $t - 1$, and let $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_t, f(x_t))$ be $t$ distinct points of $f(x)$. Then, given $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_t, f(x_t))$, one can reconstruct $f(x)$ as

\[ f(x) = f(x_1) \lambda^x_{x_1} + f(x_2) \lambda^x_{x_2} + \cdots + f(x_t) \lambda^x_{x_t}, \]

where

\[ \lambda^x_{x_j} = \frac{(x-x_1) \cdots (x-x_{j-1})(x-x_{j+1}) \cdots (x-x_t)}{(x_j-x_1) \cdots (x_j-x_{j-1})(x_j-x_{j+1}) \cdots (x_j-x_t)}, \]

for any $1 \leq j \leq t$.

3. Definitions

This section is dedicated to the definition of a $(t, n)$ threshold PVSS scheme and its security model.

3.1. Threshold PVSS Scheme without Secure Channels. Let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be a set of $n$ participants. A dealer wants to share a secret $S$ among the participants of $\mathcal{U}$ in such a way that any $t$ or more participants can recover the secret, while no $t - 1$ participants can obtain any information about the secret.

A PVSS scheme is described by the following algorithms.

(1) Setup ($\lambda$): takes as input a secure parameter $\lambda$.

(i) The dealer generates all public parameters of the scheme.
(ii) Furthermore, every participant selects its channel protection key $d_i$ and publishes the corresponding public key $D_i$.
(iii) The dealer randomly picks a number as the main secret of the system and uses $D_i$ ($i = 1, 2, \ldots, n$) and the main secret to generate a main shadow $Y_i$ ($i = 1, 2, \ldots, n$) for every participant and the system shadow verification key (SVK).
(iv) For each $U_i$, the dealer sends $U_i$’s main shadow $Y_i$ through public channels.

(2) Secret distribution: the dealer randomly selects a secret $S$ that will be distributed to the participants. It calculates and publishes the secret commitment value (SCV) $s_1$ and the secret deriving value (SDV) $s_2$ of the secret $S$. It then outputs $(s_1, s_2)$. A participant can use $s_1$ and its main-shadow $Y_i$ to obtain its subshadow of the secret $S$ by itself.

(3) Verification $(s_1, s_2)$: takes as inputs $s_1$ and $s_2$ of a secret. It is required that $s_1$ be public verifiable. Knowing only the publicly parameter, anyone may verify that $s_2$ is consistent with $s_1$. If the verification fails, the verifier broadcasts a complaint about the dealer.

(4) Reconstruction: this algorithm is composed of three subalgorithms.

(a) Subshadow generation $(s_1, s_2, U_1, Y_i, d_i)$: takes as inputs $s_1$ and $s_2$ of a secret, a participant $U_i$, $U_i$’s main shadow $Y_i$, and $U_i$’s channel protection key $d_i$. To generate its sub-shadow, $U_i$ executes verification $(s_1, s_2)$. If the verification fails, $(i, 1)$ is output. Otherwise, $U_i$ generates the sub-shadow $S_i$ from $s_1$ and $s_2$ using its main shadow $Y_i$ and channel protection key $d_i$. $(i, S_i)$ is then output.

(b) Sub-shadow verification $(s_1, U_i, SVK_i, S_i)$: takes as inputs $s_1$ of a secret, a participant $U_i$, and $U_i$’s verification key SVK$_i$, and $U_i$’s sub-shadow $S_i$. This algorithm checks whether $S_i$ is a valid sub-shadow with respect to SVK$_i$ and $s_1$. If the verification fails, a complaint about the participant $U_i$ is broadcast.

(c) Combine $(s_1, s_2, \Gamma, \Omega)$: takes as inputs $s_1$ and $s_2$ of a secret, a qualified set $\Gamma \subseteq \mathcal{U}$ of $t$ participants, and a list $\Omega = \{S_1, \ldots, S_t\}$ of $t$ valid sub-shadows. Outputs a secret $S$.

3.2. Security Model. The PVSS scheme described above must satisfy the following properties.

(i) Correctness: if the dealer and the participants act honestly, any $t$ or more participants can reconstruct the secret correctly during the execution of the reconstruction algorithm.
(ii) **Verifiability**: a successful verification of the SCV and SDV of a secret implies that the SCV and SDV are consistent.

(iii) **Privacy**: the basic requirement is that it is impossible for any collusion of less than \( t \) participants to obtain any information about a secret.

Hereafter, we will use the notion of a CSA to define the security of the PVSS scheme. We mostly follow the notation from \([19,23]\), using a game between an adversary \( \mathcal{A} \) and a challenger \( \mathcal{C} \).

(i) **Init.** \( \mathcal{C} \) executes \( \text{Setup}(\lambda) \) to obtain the public parameters and sends the public parameters to \( \mathcal{A} \) along with all of the shadow verification keys SVK.

(ii) **Phase 1.** The adversary adaptively selects a secret and generates \((s_1,s_2)\) about the secret using the public parameters just as the dealer does. Moreover, the adversary \( \mathcal{A} \) is permitted to query a sub-shadow of a participant using \((s_1,s_2)\).

(iii) **Sub-shadow query.** On being input a participant \( U_i \), as well as \( s_1 \) and \( s_2 \), \( \mathcal{C} \) executes the Sub-shadow generation sub-algorithm using \((s_1,s_2)\), \( Y_i \) and \( d_i \) then forwards the resulting \((i,S_i)\) or \((i,\perp)\) to the adversary \( \mathcal{A} \).

(iv) **Challenge.** The adversary \( \mathcal{A} \) outputs a target set of participants \( U^* \), where \(|U^*| < t \). The challenger \( \mathcal{C} \) picks two random secrets \( T_0 \) and \( T_1 \) as well as a random bit \( \delta \in \{0,1\} \). Then, \( \mathcal{C} \) executes the secret distribution algorithm to obtain \((s_1^*,s_2^*)\) for the secret \( T_\delta \) and sends \((s_1^*,s_2^*)\) and all the sub-shadows of each \( U_i^* \in U^* \) to \( \mathcal{A} \) along with \( T_0 \) and \( T_1 \).

(v) **Phase 2.** The adversary continues to adaptively issue the subshadow query as in phase 1, but with the constraint that \((s_1,s_2) \neq (s_1^*,s_2^*)\), and challenger \( \mathcal{C} \) responds similarly in phase 1.

(vi) **Guess.** Finally, the adversary \( \mathcal{A} \) outputs a guess \( \delta' \in \{0,1\} \) and wins the game if \( \delta = \delta' \).

**Definition 1 (IND-CSA security).** A \((t,n)\) PVSS scheme has indistinguishability against adaptive CSA if, for any probabilistic polynomial-time (PPT) adversary \( \mathcal{A} \), the advantage

\[
\text{ADV}_{\mathcal{A},t,n}^{\text{IND-CSA}}(\lambda) = \Pr[\delta = \delta'] - \frac{1}{2}
\]

is negligible with respect to \( \lambda \).

### 4. Construction

In this section, we present a concrete \((t,n)\) PVSS scheme and prove its security against CSA in the next section.

**Setup.** The dealer obtains the group parameters \((p,P,G,G_1,e(\cdot,\cdot))\) by executing the group generator algorithm \( \mathcal{G}(\lambda) \). It then selects a random integer \( \gamma \in \mathbb{Z}_p \) \( \setminus \{0\} \) and publishes \((p,G,G_1,e(\cdot,\cdot),P,P_2 = \gamma P)\) on the system bulletin board (BB). Supposing that \( U = \{U_1,\ldots,U_n\} \) is the set of participants of the system, each participant may be uniquely identified by means of an index \( i \in \{1,\ldots,n\} \). After the dealer has announced the public parameters, each participant \( U_i \) randomly selects an integer \( d_i \in \mathbb{Z}_p \setminus \{0\} \) (\( d_i \) is \( U_i \)'s channel protection key) and calculates \( D_i = d_i P_2 \). Each participant keeps \( d_i \) confidentially and sends \( D_i \) to the dealer over public channels.

Having received all the \( D_i \), the dealer performs the following operations.

(1) The dealer selects a random number \( \alpha \in \mathbb{Z}_p \setminus \{0\} \) and a random polynomial \( f(x) = \alpha + \sum_{j=1}^{t-1} \alpha_j x^j \) with degree \( t - 1 \), where \( t \) is the threshold value and, for \( j = 1,\ldots,t-1, \alpha_j \in \mathbb{Z}_p \setminus \{0\} \).

(2) The dealer computes \( P_i = \alpha P \) and \( Y_i = f(i) D_i \) (for \( i = 1,\ldots,n \)), where \( Y_i \) is the main shadow of the participant \( U_i \).

(3) The dealer selects a collision-resistant hash function \( H : G \rightarrow \{0,1\}^l \), where \( l \) is the output length of \( H \).

(4) The dealer sets the SVK = \((f(1)P,f(2)P,\ldots,f(n)P)\) as the shadow verification key.

(5) The dealer randomly selects \( u',u_1,u_2,\ldots,u_l \in \mathbb{G} \).

Finally, the dealer sends \( Y_i \) to the participant \( U_i \) \((i = 1,\ldots,n)\) through public channels and publishes \((P_i,H,u',u_1,u_2,\ldots,u_l,\text{SVK})\) on the bulletin board.

**Secret Distribution.** The dealer wants to share a secret, which is a random element in \( G_1 \). The form of the secret is \( \delta' = e(P_i,P_2)\delta \), where \( k \) is selected randomly from \( \mathbb{Z}_p \setminus \{0\} \). Let \( v \) be a bit string of length \( l \), let \( v_i \) denote the \( i \)th bit of \( v \), and let \( \mathcal{F} \subseteq \{1,\ldots,l\} \) be the set of all \( i \) for which \( v_i = 1 \). The dealer calculates and publishes the SCV \( s_1 \) and SDV \( s_2 \) as follows:

\[
\begin{align*}
    s_1 &= k P, \\
    v &= H(s_1), \\
    s_2 &= k \left( u' + \sum_{i \in \mathcal{F}} u_i \right).
\end{align*}
\]

The dealer either broadcasts \((s_1,s_2)\) to all participants or publishes \((s_1,s_2)\) on the BB. (\(U_i\)'s real sub-shadow \( S_i \) for the secret \( \delta' \) is \( e(s_1,-d_i f(i)D_i) \). In order to achieve CSA security, in the reconstruction algorithm, no participant \( U_i \) directly sends \( S_i \) to other participants.) If the dealer wants to share a new secret, it just executes the secret distribution algorithm again and publishes appropriate information on the BB. However, the main shadow \( Y_i \) of \( U_i \) need not be changed.

**Verification.** Given \((s_1,s_2)\), this algorithm first computes \( v = H(s_1) \) and outputs “valid” or “invalid” according to the following:

\[
e(s_2,P) = e\left( u' + \sum_{i \in \mathcal{F}} u_i, s_1 \right).
\]
Reconstruction. Without loss of generality, let us assume that $Γ = \{1, \ldots, t\}$ is a qualified subset of the set of participants, that is, it consists of at least $t$ participants who want to collectively reconstruct the secret $s'$. Each participant in $Γ$ executes the following algorithms:

1. **Sub-shadow generation** $(s_1, s_2, U_1, Y_1, d_i)$: takes as inputs $s_1$ and $s_2$, a participant $U_1$, $U_1$’s main shadow $Y_1$, and $U_1$’s channel protection key $d_i$. To generate its sub-shadow, $U_1$ executes the verification $(s_1, s_2)$ algorithm. If the verification fails, $U_1$ outputs $(i, \bot)$ and exits. Otherwise, $U_1$ randomly selects $β_i \in \mathbb{Z}_p \setminus \{0\}$ and performs the following calculations:

\[ v = H(s_1), \]
\[ μ_{i1} = (-d_i)Y_i + β_i \left( u' + \sum_{j \in V} u_j \right), \quad (7) \]
\[ μ_{i2} = β_iP. \]

At this point, $U_1$ sends $S_i = \{μ_{i1}, μ_{i2}\}$ to the other participants in $Γ$ through secure channels. ($U_1$ may use the method described in Section 7 to send $S_i = \{μ_{i1}, μ_{i2}\}$ through public channels.)

2. **Sub-shadow verification** $(s_1, U_1, SVK_i, S_i)$: takes as inputs $s_1$ for a secret, a participant $U_i$, and $U_i$’s verification key $SVK_i$, $U_i$’s sub-shadow $S_i = \{μ_{i1}, μ_{i2}\}$. Another participant $U_j \in Γ$ computes $v = H(s_1)$ and then checks

\[ e(μ_{i1}, P) = e(P_{ij}, SVK_i)e \left( u' + \sum_{j \in V} u_j, μ_{i2} \right). \quad (8) \]

If the checked equality does not hold, $U_j$ demands that $U_i$ sends $(μ_{i1}, μ_{i2})$ again or declares that $U_i$ is a cheater.

3. **Combine** $(s_1, s_2, Γ, Ω)$: takes as inputs $s_1$ and $s_2$ for a secret, a qualified subset $Γ \subseteq \mathcal{U}$ of the set of participants that contains at least $t$ participants, and a list $Ω = (S_1, \ldots, S_t)$ (where $S_i = \{μ_{i1}, μ_{i2}\}$) of $t$ valid subshadows. Each participant $U_i \in Γ$ first computes the Lagrange coefficients $λ_1, λ_2, \ldots, λ_t \in \mathbb{Z}_p$, where $\lambda_i = \prod_{j=1, j \neq i}^{t} \frac{(0 - j)/(i - j)}{(0 - j)/(i - j))$, then calculates

\[ μ_{i1} = \sum_{i=1}^{t} λ_i \mu_{i1}, \quad (9) \]
\[ μ_{i2} = \sum_{i=1}^{t} λ_i \mu_{i2}. \]

At this point, every participant in $Γ$ uses $(s_1, s_2, μ_{i1}, μ_{i2})$ to reconstruct the secret $s'$ as follows:

\[ s' = \frac{e(s_1, μ_{i1})}{e(s_2, μ_{i2})}. \quad (10) \]

5. **Security and Correctness**

5.1. Correctness. If the dealer and the participants are honest, any $t$ or more participants can reconstruct the secret during the execution of the reconstruction algorithm. The correctness of equalities (6), (9), and (10) is as follows.

\[ e(μ_{i1}, P) = e(−d_i Y_i + β_i \left( u' + \sum_{j \in V} u_j \right), P) \]
\[ = e(P, f(i)P) e \left( u' + \sum_{j \in V} u_j, β_i P \right) \quad (11) \]
\[ = e(P, SVK_i) e \left( u' + \sum_{j \in V} u_j, μ_{i2} \right), \]
\[ μ_{i1} = \sum_{i=1}^{t} λ_i μ_{i1} \]
\[ = \sum_{i=1}^{t} \left( λ_i (-d_i Y_i) + λ_i β_i \left( u' + \sum_{j \in V} u_j \right) \right) \quad (12) \]
\[ = \sum_{i=1}^{t} f(i) P + \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right) \]
\[ = αP + \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right), \]
\[ μ_{i2} = \sum_{i=1}^{t} λ_i μ_{i2} \quad (13) \]
\[ = \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right) \]
\[ = e(s_1, μ_{i1}) \]
\[ = e(s_2, μ_{i2}) \]

\[ s' = \frac{e(s_1, μ_{i1})}{e(s_2, μ_{i2})} \]
\[ = e(kP, αP) + \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right) \]
\[ = e(kP, αP) e(kP, \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right)) \]
\[ = e(kP, \sum_{i=1}^{t} β_i λ_i \left( u' + \sum_{j \in V} u_j \right)) \]
\[ = e(P, P)^k. \quad (14) \]

5.2. Security

Theorem 2 (IND-CSA of PVSS). Suppose the hash function $H$ is a universal collision-resistant one-way family. Then, the proposed PVSS scheme is secure against adaptive CSA under the intractability assumption of the DBDH problem. More specifically, if there is an adversary that can break the PVSS scheme within time $T$ with probability at least $e$, then there
exists an algorithm that can solve the DBDH problem within time $T'$ with probability at least $e'$, where

$$T' = T + T'_{\text{Q}}, \quad e' \geq \frac{3}{4}e. \quad (15)$$

Here, $T'_{\text{Q}}$ denotes the time taken to answer all queries.

**Proof.** Suppose an adversary $\mathcal{A}$ breaks the PVSS scheme with advantage $\text{ADV}_{\text{IND-CMA}}^{\mathcal{A}} > \epsilon$. Then we can devise an algorithm $\mathcal{R}$ that solves a random DBDH problem instance with advantage $e' \geq (3/4)e$. Algorithm $\mathcal{R}$ is given as input a group parameter $G = (P, G, G_1, \ell(\cdots))$ and a random tuple $(G, P, aP, bP, cP, T)$, where $T$ is a random element of $G_1$ or $T = e(P, P)^{abc}$. The goal of algorithm $\mathcal{R}$ is to output 1 ("true") if $T = e(P, P)^{abc}$ and 0 ("false") otherwise. Set $P_1 = aP, P_2 = bP, P_3 = cP$. Algorithm $\mathcal{R}$ works by interacting with $\mathcal{A}$ in a game as follows:

**Init.** Algorithm $\mathcal{R}$ does the following.

1. Algorithm $\mathcal{R}$ chooses a set $U$ containing $t - 1$ participants. Without loss of generality, let $U = \{U_1, \ldots, U_{t-1}\} \subset \mathcal{U} = \{U_1, \ldots, U_n\}$.
2. Algorithm $\mathcal{R}$ selects a collision-resistant hash function $H : G \rightarrow \{0, 1\}^t$. It then loses the public keys for all participants in $\mathcal{U}$ as follows: $D_i = d_iP_i$ for all $i \in \mathcal{R}$ and $d_i \overset{\$}{\leftarrow} Z_p \setminus \{0\}$, where $d_i$ is the channel protection key of $U_i$.
3. Algorithm $\mathcal{R}$ selects $t - 1$ random integers $a_i \in Z_p$, where $i = 1, \ldots, t - 1$, and $a_{t-1} \neq 0$. There exists a polynomial $f(x) \in Z_p[X]$ of degree $t - 1$ such that for all $U_j \in U$, $f(0) = a$ and $f(i) = a_i$. However, $\mathcal{R}$ does not know the polynomial $f$, because it does not know $a$.
4. Algorithm $\mathcal{R}$ constructs the shadow verification key $\text{SVK}_i$ as follows.
   - (i) If $U_j \in U$, since $\mathcal{R}$ knows $f(i) = a_i$, he can compute the shadow verification key $\text{SVK}_i = f(i)P = a_iP$.
   - (ii) If $U_j \notin U$, $\mathcal{R}$ computes the Lagrange coefficients $\lambda_0^i, \lambda_1^i, \ldots, \lambda_{t-1}^i \in Z_p$, such that $f(i) = \lambda_i^if(0) + \sum_{j=1}^{t-1} \lambda_j^if(j)$. Algorithm $\mathcal{R}$ then sets

$$\text{SVK}_i = \lambda_0^iaP + \lambda_1^ia_1P + \cdots + \lambda_{t-1}^ia_{t-1}P_i \quad (16)$$

which entails that $\text{SVK}_i = f(i)P$, as required.
5. Algorithm $\mathcal{R}$ follows Waters’ [24] method to simulate it as follows.
   - (i) For $u$, it lets $\mathcal{U}' \subseteq \{1, \ldots, t\}$ be the set of all $i$ for which $u_i = 1$.
   - (ii) It sets an integer $m = 4q$ (where $q$ is the maximum number of sub-shadow queries) and randomly chooses an integer $k$ between 0 and $l$. algorithm $\mathcal{R}$ chooses a random vector $\vec{x} = (x_i)$ of length $l$, where $x_i \in [0, m - 1]$.
   - (iv) It lets $v' = H(cP)$, and computes $x' = km - \sum_{i \in \mathcal{U}'} x_i$.
   - (v) It chooses a random integer $h' \in Z_p$ and a vector $\vec{h} = (h_i)$ of length $l$, where $h_i \in Z_p$.
   - (vi) It sets $u' = (p - km + x')P + h'P, u_i = x_iP + h_iP$.
   - (vii) It defines $F(v) = (p - mk) + x' + \sum_{i \in \mathcal{U}'} x_i, J(v) = 2\sum_{i \in \mathcal{U}'} h_i$ and

$$K(v) = \begin{cases} 0, & \text{if } x' + \sum_{i \in \mathcal{U}'} x_i = 0 \pmod{m}, \vspace{2mm} \cr 1, & \text{otherwise}. \end{cases} \quad (17)$$

6. Algorithm $\mathcal{R}$ sends $(B, P, P_1, P_2, H, e(\cdot, \cdot), \text{SVK}_i (i = 1, 2, \ldots, n), u'_i, u_i (i = 1, \ldots, l))$ to the adversary $\mathcal{A}$.

**Phase I.** The adversary $\mathcal{A}$ adaptively selects a secret $\delta' = e(P_j, P_j)^{\delta} \in Z_p$ and generates $(s_1, s_2)$ for the secret $\delta'$ using the public parameters $(P_1, P_2, e(\cdot, \cdot))$ just as the dealer does. Then, $\mathcal{A}$ adaptively issues a sub-shadow query of the form $(i, s_1, s_2)$, where $i \in \{1, \ldots, n\}$ for each such sub-shadow query the following applies.

- (i) The algorithm $\mathcal{R}$ computes $v = H(s_1)$ and checks $e(s_1, P) = e(u' + \sum_{i \in \mathcal{U}'} u_i, s_1)$. If the equality does not hold, $\mathcal{R}$ responds to $\mathcal{A}$’s query with $(i, \bot)$.
- (ii) Otherwise, $\mathcal{R}$ continues to check whether $K(\mathcal{U}') = 0$ holds. If it does hold, $\mathcal{R}$ aborts the game and randomly selects a bit as the answer to the DBDH problem.

- (iii) Otherwise, there are two different cases as follows.
  - (i) If $i \in \{1, \ldots, t-1\}$, computing $\mu_1, \mu_2$ is easy, because $f(i)$ is equal to one of the $\alpha_1, \ldots, \alpha_{t-1}$, which are known to $\mathcal{R}$. Thus, $\mathcal{R}$ randomly selects $\beta_i \in Z_p \setminus \{0\}$ and performs the following calculations:

$$\mu_1 = (-d_i)Y_i + \beta_i \left( u' + \sum_{j \in \mathcal{U}'} u_j \right)$$
$$\mu_2 = \beta_iP. \quad \text{(18)}$$

$\mathcal{R}$ sends the sub-shadow $S_i = (\mu_1, \mu_2)$ of $U_i$ to $\mathcal{A}$.
  - (ii) If $i \notin \{1, \ldots, t-1\}$, $\mathcal{R}$ randomly selects $\beta_i \in Z_p \setminus \{0\}$ and performs the following calculations:

$$\mu_1 = \frac{-I(\mathcal{U}')}{F(\mathcal{U}')} \lambda_0^iP_1 + \beta_i \left( u' + \sum_{j \in \mathcal{U}'} u_j \right) + \chi, \quad \text{(19)}$$

$$\mu_2 = \beta_iP + \frac{-\lambda_0^i}{F(\mathcal{U}')} P_1,$$

where $\chi = \sum_{j=1}^{t-1} \lambda_j^i \alpha_jP_i$. 

International Journal of Distributed Sensor Networks
We claim that $S_i = \{\mu_1, \mu_2\}$ is a valid sub-shadow for $\{i, (s_1, s_2)\}$. To see this, let $\beta_i = \beta_i - (a \lambda_0^i/F(\mathcal{V}))$. Then we have

$$
\mu_1 = -J(\mathcal{V}) \lambda_0^i P_1 + \beta_i \left( u' + \sum_{j \in \mathcal{V}} u_j \right) + \chi
$$

and

$$
\mu_2 = -J(\mathcal{V}) \lambda_0^i P_1 + \beta_i \left( u' + \sum_{j \in \mathcal{V}} u_j \right) + \chi
$$

where $\chi = J(\mathcal{V}) P + \lambda_0^i P_1$.

(ii) If $U_i^* \notin U$, the sub-shadow of $U_i^*$ is

$$
\mu_1^* = -J(\mathcal{V}) \lambda_0^i P_1 + \beta_i \left( u' + \sum_{j \in \mathcal{V}} u_j \right) + \sum_{j=1}^{t-1} \lambda_j^i \alpha_j P_2,
$$

and

$$
\mu_2^* = \beta_i P + -\lambda_0^i P_1,
$$

where $\chi = J(\mathcal{V}) P + \lambda_0^i P_1$.

At this point, algorithm $\mathcal{A}$ randomly selects a bit $\delta \in \{0, 1\}$, sets $T_\delta = T$, and assigns a random value in the secret space $G_1$ to $T_{1-\delta}$. $\mathcal{A}$ then sends $(s_1^*, s_2^*, T_0, T_1)$ to the adversary $\mathcal{A}'$, where $i \in \Phi$.

Claim 1. Our simulation does not abort with probability greater than $\frac{3}{4}$.

Proof. Without loss of generality, let us assume that the adversary $\mathcal{A}$ makes the maximum number $q$ of sub-shadow queries. For any sub-shadow query of a participant and the $v_1, v_2, \ldots, v_3 (\neq v^*)$, we have

$$
\Pr \left[ \text{abort} \right] = \Pr \left[ \bigwedge_{i=1}^{q} K((v_i)) = 1 \right]
$$

$$
= 1 - \Pr \left[ \bigvee_{i=1}^{q} K((v_i)) = 0 \right]
$$

$$
\geq 1 - \sum_{i=1}^{q} \Pr [K((v_i)) = 0]
$$

$$
= 1 - \frac{q}{m}
$$

$$
= \frac{3}{4} \quad (\text{for } m = 4q).
$$

This completes the proof of Claim 1.

Phase 2. The adversary $\mathcal{A}$ continues to issue queries about a sub-shadow of the form $[i, (s_1, s_2)]$, where $i \in \{1, \ldots, n\}$ and $(s_1, s_2) \neq (s_1^*, s_2^*)$. Algorithm $\mathcal{A}$ responds as in phase 1.

Guess. Eventually, $\mathcal{A}$ outputs a guess bit $\delta' \in \{0, 1\}$ for $\delta$. Based on the value of $\delta'$, $\mathcal{A}$ concludes its own game by outputting a guess as follows.

(i) If $\delta' = \delta$, $\mathcal{A}$ answers 1, meaning that $T = e(P, P)^{abc}$.

(ii) Otherwise, $\mathcal{A}$ answers 0, meaning that $T$ is a random element of $G_1$.

If the input $(P, aP, bP, cP, T)$ satisfies $T = e(P, P)^{abc}$, then $\mathcal{A}'$'s view is identical to its view in a real attack game, and therefore $\mathcal{A}'$ must satisfy $|\Pr[\delta = \delta'] - 1/2| > \epsilon$. On the other hand, if the input of $(P, aP, bP, cP, T)$ satisfies $T \neq e(P, P)^{abc}$, then $\mathcal{A}'$'s view is different from its view in a real attack game, and therefore $\mathcal{A}'$ must be incorrect.
According to Claim 1, we have that
\[
\Pr\left(\delta = \delta'\right) = 1/2. \text{ Therefore, with } P \in \mathbb{G}, a, b, c \in \mathbb{Z}_p \text{ and } T \in \mathbb{G}, \text{ we have}
\]
\[
\left|\Pr\left(\mathcal{R} \left( G, P, aP, bP, cP, e(P, P)^{abc}\right) = 1\right) \right|
\]
\[
- \Pr\left[\mathcal{R} \left( G, P, aP, bP, cP, T\right) = 1\right] \right| \geq \left|\frac{1}{2} \pm \epsilon\right| - \frac{1}{2} = \epsilon.
\]
According to Claim 1, we have that \(e' \geq (3/4)e\). This completes the proof of Theorem 2. \(\square\)

6. Comparison

Now, let us compare our scheme to WT11 [2] and HV09 [3] in terms of computational cost and security. We firstly define the following notations.

(i) \(T_p\): The time taken to execute a bilinear pairing operation \(e(\cdot, \cdot)\).

(ii) \(T_m\): The time taken to execute a scalar multiplication operation of point in \(G\).

(iii) \(T_\gamma\): The time taken to execute a modular exponent operation in \(G_1\).

(iv) \(|p|\): The binary length of order \(p\).

(v) \(|H|\): The output length of the hash function \(H\).

As is well known, the time taken to execute \(T_p, T_m,\) and \(T_\gamma\) is much greater than the other operations, so we will ignore the time consumption of the other operations, such as executing an addition operation of points in \(G\). The details of the comparison are given in Table 2. In Table 3, we compare the communication cost of the dealer distributing a secret to the participants, and a participant sends its sub-share to other participants.

From the comparison in Tables 2 and 3, one can see that our scheme achieves a higher level of security without significantly increasing the overall computational complexity and the communication cost.

7. Extension Scheme

In the basic scheme described previously, the secret reconstruction requires the presence of point-to-point secure channels among the participants. In this section, we remove this limitation without sacrificing any good property of the scheme.

Suppose that a participant \(U_j\) wants to send its sub-shadow through a public channel to a participant \(U_i\). For this purpose, \(U_j\) randomly selects \(\beta_j, r \in \mathbb{Z}_p \cup \{0\}\), uses \(U_i\)'s public key \(D_j\), and the following calculations are performed:

\[
v = H(\mathcal{S}_1),
\]

\[
v_1 = (-d_i)Y_i + \beta_i\left(u'_i + \sum_{j \in F} u_j\right) + rD_j,
\]

\[
v_2 = \beta_i P,
\]

\[
v_3 = rP_2.
\]

Then, \(U_i\) sends \(\{v_1, v_2, v_3\}\) to \(U_j\), \(U_j\) computes \(\mu_i = v_i - d_i v_3\), \(\mu_j = v_2\), and then checks whether or not

\[
e(\mu_i, P) = e(P_2, SVK_i)e\left(u'_i + \sum_{j \in F} u_j, \mu_j\right).
\]

Having collected \(t\) valid sub-shadows, \(U_i\) first computes \(\mu_1 = \sum_{j=1}^t \lambda^j_1 \mu_{j0}, \mu_2 = \sum_{j=1}^t \lambda^j_2 \mu_{j1}\) and then reconstructs the secret \(\mathcal{S}\) by computing \(e(s_1, \mu_1)/e(s_2, \mu_2)\) just as it does in the basic scheme.

8. Conclusion

In this paper, we proposed a \((t,n)\) threshold PVSS scheme. Under the decisional bilinear Diffie-Hellman assumption, we proved that our scheme has indistinguishability against adaptively chosen secret attacks in the standard model. In the secret distribution phase, the dealer can send the main shadow to a participant through public channels. When the participants exchange their sub-shadows in the secret reconstruction phase, point-to-point secure channels need not be established in the extended scheme. This scheme is fairly interesting for practical applications.
Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) Programs (nos. 61070251, 61003285, 6103198, and 61272534), the NSFC A3 Foresight Program (no. 6116140320), and the JSPS KAKENHI program (23500031).

References

[1] A. Shamir, “How to share a secret,” Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.
[2] 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.
[3] S. Heidarvand and J. Villar, “Public verifiability from pairings in secret sharing schemes,” in Selected Areas in Cryptography, R. Avanzi, L. Keliher, and F. Sica, Eds., vol. 5381 of Lecture Notes in Computer Science, pp. 294–310, Springer, Berlin, Germany, 2009.
[4] A. Beimel and M. Franklin, “Weakly-private secret sharing schemes,” in Theory of Cryptography, S. Vadhan, Ed., vol. 4392 of Lecture Notes in Computer Science, pp. 253–269, Springer, Berlin, Germany, 2007.
[5] E. Fujisaki and T. Okamoto, “A practical and provably secure scheme for publicly verifiable secret sharing and its applications,” in Advances in Cryptology-EUROCRYPT ’98, K. Nyberg, Ed., vol. 1403 of Lecture Notes in Computer Science, pp. 32–48, Springer, Berlin, Germany, 1998.
[6] R. J. Hwang and C. C. Chang, “An on-line secret sharing scheme for multi-secrets,” Computer Communications, vol. 21, no. 13, pp. 1170–1176, 1998.
[7] A. Patra, A. Choudhary, T. Rabin, and C. Rangan, “The round complexity of verifiable secret sharing revisited,” in Advances in Cryptology-CRYPTO 2009, S. Halevi, Ed., vol. 5677 of Lecture Notes in Computer Science, pp. 487–504, Springer, Berlin, Germany, 2009.
[8] R. Kumaresan, A. Patra, and C. Rangan, “The round complexity of verifiable secret sharing: the statistical case,” in Advances in Cryptology-ASIACRYPT 2010, M. Abe, Ed., vol. 6477 of Lecture Notes in Computer Science, pp. 431–447, Springer, Berlin, Germany, 2010.
[9] Hong Yu, Jingsha He, Ting Zhang, and Peng Xiao, “A group key distribution scheme for wireless sensor networks in the internet of things scenario,” International Journal of Distributed Sensor Networks, vol. 2012, 12 pages, 2012.
[10] G. R. Blakley, “Safeguarding cryptographic keys,” in Proceedings of the National Computer Conference, vol. 1 of Managing Requirements Knowledge, pp. 313–329, 1979.
[11] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, “Verifiable secret sharing and achieving simultaneity in the presence of faults,” in Proceedings of the 26th Annual Symposium on Foundations of Computer Science, pp. 383–395, 1985.
[12] P. Feldman, “A practical scheme for non-interactive verifiable secret sharing,” in Proceedings of the 28th Annual Symposium on Foundations of Computer Science, pp. 427–438, Los Angeles, Calif, USA, 1987.
[13] T. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” in Advances in Cryptology-CRYPTO ’91, J. Feigenbaum, Ed., vol. 576 of Lecture Notes in Computer Science, pp. 129–145, Springer, Berlin, Germany, 1992.
[14] B. Schoenmakers, “A simple publicly verifiable secret sharing scheme and its application to electronic voting,” in Advances in Cryptology-CRYPTO ’99, M. Wiener, Ed., vol. 1666 of Lecture Notes in Computer Science, pp. 784–799, Springer, Berlin, Germany, 1999.
[15] A. Ruiz and J. L. Villar, “Publicly verifiable secret sharing from paillier’s cryptosystems,” in Proceedings of the Western European Workshop on Research in Cryptology (WEWoRC ’05), vol. 74 of Lecture Notes in Informatics, pp. 98–108, 2005.
[16] M. Stadler, “Publicly verifiable secret sharing,” in Advances in Cryptology-EUROCRYPT ’96, U. Maurer, Ed., vol. 1070 of Lecture Notes in Computer Science, pp. 190–208, Springer, Berlin, Germany, 1996.
[17] A. Fiat and A. Shamir, “How to prove yourself: practical solutions to identification and signature problems,” in Advances in Cryptology-CRYPTO ’86, A. Odlyzko, Ed., vol. 263 of Lecture Notes in Computer Science, pp. 186–204, Springer, Berlin, Germany, 1987.
[18] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” in Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73, ACM, November 1993.
[19] R. Cramer and V. Shoup, “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack,” in Advances in Cryptology-CRYPTO ’98, H. Krawczyk, Ed., vol. 1462 of Lecture Notes in Computer Science, pp. 13–29, Springer, Berlin, Germany, 1998.
[20] M. Jhanwar, “A practical (non-interactive) publicly verifiable secret sharing scheme,” in Information Security Practice and Experience, F. Bao and J. Weng, Eds., vol. 6672 of Lecture Notes in Computer Science, pp. 273–287, Springer, Berlin, Germany, 2011.
[21] W. Huaxiong and D. S. Wong, “On secret reconstruction in secret sharing schemes,” Information Theory, IEEE Transactions on, vol. 54, no. 1, pp. 473–480, 2008.
[22] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in Advances in Cryptology-CRYPTO 2001, J. Kilian, Ed., vol. 2139 of Lecture Notes in Computer Science, pp. 213–230, Springer, Berlin, Germany, 2001.
[23] D. Boneh and X. Boyen, “Efficient selective-ID secure identity-based encryption without random oracles,” in Advances in Cryptology-EUROCRYPT 2004, C. Cachin and J. Camenisch, Eds., vol. 3027 of Lecture Notes in Computer Science, pp. 223–239, Springer, Berlin, Germany, 2004.
[24] B. Waters, “Efficient identity-based encryption without random oracles,” in Advances in Cryptology-EUROCRYPT 2005, R. Cramer, Ed., vol. 3494 of Lecture Notes in Computer Science, pp. 557–557, Springer, Berlin, Germany, 2005.