A Joint Random Secret Sharing Scheme with Public Verifiability

Zhenhua Chen[1], Shundong Li[2], Qiong Huang[3], Jianhua Yan[4], Yong Ding[5]
(Corresponding author: Zhenhua Chen)
School of Computer Science and Technology, Xi'an University of Science and Technology[1]
Shaanxi, China
(Email: chenzhenhua@snnu.edu.cn)
School of Computer Science, Shaanxi Normal University[2]
College of Mathematics and Informatics, South China Agricultural University[3]
State Key Laboratory of Networking and Switching Technology, Beiyou University[4]
Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology[5]
(Received July 12, 2015; revised and accepted Oct. 15 & Nov. 2, 2015)

Abstract

In this paper, we propose a joint random secret sharing scheme with public verifiability. It is practical in distributed environment. Utilizing additive homomorphism, a random secret will be corporately constructed by some participants, which avoids the need for a mutually trusted dealer. In addition, we explore the technique of homomorphic verification and that of bilinear pairing to allow each participant to publicly verify whether the received shares are consistent. The verification process in our scheme is unconditionally secure and non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof, which is simple and higher efficient compared with previously known. Lastly, as an applied example of our work, we present how our techniques can be applied to handle dynamic node-join in mobile ad hoc network.

Keywords: Additive homomorphism, bilinear pairing, joint random secret sharing, public verifiability, unconditionally secure

1 Introduction

1.1 Background and Motivation

Secret sharing, invented independently by Shamir [24] and Blakley [2] in 1979, is a fundamental cryptographic primitive that has been found useful in numerous applications such as witness encryption [13], access control [15], secure group communication [10], secure information communication [16], and cloud storage [4]. In a (k, n) secret sharing scheme (SSS), a secret s is distributed to n participants by a dealer in such a way that any k participants or more can reconstruct the secret s but participants less than k learn nothing about s with their shares. However, there are several common drawbacks in the existing SSS [2, 24].

1) A dishonest dealer may distribute a fake share to a certain participant, and then the participant would never obtain the true secret subsequently.

2) A malicious participant may submit a fake share, which makes it the only one who gets to reconstruct the true secret after observing the shares of honest participants, whereas the honest cannot obtain the true secret.

3) A mutually trusted dealer must be needed for generating and distributing each share to the corresponding participant secretly.

4) It is required that there exists a private channel between the dealer and each participant during the share distribution phase.

To address the cheating problems in 1) and 2), a verifiable algorithm can be added to SSS, which is called verifiable secret sharing scheme (VSSS) [3] and can be used to verify whether the shares are consistent. VSSS further are investigated by many other researchers. Feldman [5] and Pederson [22] proposed a VSSS based on Shamir's scheme, respectively, which can effectively detect cheating of the participants or the dealer. The security of verifiability in VSSS can be classified into two types: computational security and unconditional security. The verification process in the former is unconditionally secure, whereas that in the latter is computationally secure. More specifically, it is based on the hardness of solving the discrete logarithm.

In many general schemes including those discussed above, there is a dealer whose function is to distribute the shares among participants. Nevertheless, it is difficult to

find a trustworthy person or organization as the dealer in the real world. In addition, the mutually trusted dealer has too much rights and is easy to suffer from the adversary's attacks. To deal with the drawbacks in 3), Ingemarsson and Simmons [11] introduced a new type of SSS without the assistance of a mutually trusted dealer, which is called joint random secret sharing scheme (JRSSS). The basic idea of JRSSS is that each participant also acts as a dealer to select a sub-secret and to generate sub-shares for others. The master secret is the summation of all sub-secrets. However, there is one potential problem in their design, that is, each participant needs to keep n sub-shares secretly and the number of shares kept by each participant is proportional to the number of participants. Therefore, the cost of storage and management of shares are expensive. Pederson [23] proposed a solution to overcome this problem. According to the property of additive homomorphism defined in [1], each participant only needs to keep one master share secretly.

Based on Pederson's approach [23], Harn and Lin [7] introduced a new notion of strong k-consistency of shares and proposed a verifiable JRSSS (VJRSSS), which enables participants to verify their shares whether to satisfy the security requirements of a (k, n) SSS. However, in their scheme, shareholders need to utilize 100 verification polynomials to verify the strong t-consistency of master shares, which makes the verification more complicated and spends too much time. After that, numerous VJRSSS were proposed to reduce the computational complexity. In 2012, Liu et al. [19] updated the scheme of Harn and Lin [7], in which shareholders utilize the sub-polynomials of master secret to construct a verification polynomial and use it to verify master shares. In 2013, Mahmoud [20] constructed a polynomial differential-based VJRSSS, in which they calculate the t-th derivative of polynomials and apply Shamir's SSS to generate and distribute the sub verification shares and use Pedersen's SSS to find the master verification shares. However, this scheme seems not to guarantee the strong t-consistency, i.e., it cannot detect the fact of cheating.

Despite the research results, we observe that participants in the existing VJRSSS can only verify their own shares rather than others and only detect the fact of cheating but not identify who are the cheaters. Though Kaya et al. [12] designed a VJRSSS which can identify who are cheaters, their scheme does not hold public vitrifiability, i.e., the shares cannot be verified by anyone. Stadler [25] introduced a publicly verifiable secret sharing (PVSS) scheme, in which not only the participants can verify the validity of their shares, but also any one can do it from the public information without revealing shares. Note that unconditional secrecy is not possible in a PVSS scheme since the encrypted shares are sent by public channels, namely, no private channels between the dealer and each participant are assumed. So, the PVSS schemes overcome the drawbacks in 4). Recently, Villar and Heidarvand [8] construct a PVSS scheme using pairing, the verification of which is unconditionally secure and efficient since the verification process does not depend on any computational assumption and is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof. Nevertheless, their scheme requires a mutually trusted dealer to generate and distribute shares, which is impractical in a distributed scenario.

1.2 Our Contribution

In this paper, to solve the aforementioned problems, we first provide a joint random secret sharing scheme (JRSSS) with public verifiability. Our scheme is based on the technique of Pedersen's (k, n) SSS [23] and that of Villar et al.'s scheme [8]. However, the techniques in [8, 23] cannot be used in building our JRSSS with public verifiability directly. It is because that the share in verification equality in [8] is just only one, while in our scheme, that in verification equality is the summation of n sub-shares. In order to take advantage of these techniques in [8], we make a modification on their scheme with homomorphic verification.

We employ additive homomorphism to avoid the use of a mutually trusted dealer who selects and distributes the private shares, and explore homomorphic verification and the bilinear pairing to allow anyone to publicly verify whether the shares are consistent. To the best of our knowledge, our scheme first provides a distributed secret sharing scenario with public verifiability.

The primary advantages of our scheme are summarized as follows.

- Cheater identification: our scheme can not only detect the fact of cheating but also identity who are the cheaters.
- Unconditionally secure verifiability: the verifiability of our scheme does not depend on any computational assumption.
- Non-interactive verification: our scheme is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof.
- *Homomorphic property*: our scheme enjoys homomorphic property compatible with public verifiability.

1.3 Roadmap

In Section 2, we briefly review the related preliminary. In Section 3, we present our scheme in detail. The homomorphic property of our scheme is discussed in Section 4. Scheme analysis is provided in Section 5, while in Section 6, a performance comparison of our scheme with previously known is illustrated. An application of our scheme is showed in Section 7. Finally, in Section 8, we summarize our works.

2 Preliminary

2.1 Review of Pedersen's JRSSS

In this section, we review a joint random secret sharing scheme (JRSSS) originally proposed by Pedersen [23], in which each participant also acts as a dealer. Each participant P_i selects a random sub-secret s_i independently and the master secret s can be constructed cooperatively with the help of the homomorphism property [1], where $s = \sum_{i=1}^{n} s_i$. We now describe this scheme as follows.

Share Generation.

1) Sub-secret generation:
Each participant P_i selects a random secret s_i called as sub-secret.
2) Sub-share generation:
For each sub-secret s_i, the participant P_i selects a random polynomial $f_i(x)$ of degree $k-1$ such that $s_i = f_i(0)$ and uses Shamir's (k, n) SSS to generate sub-shares s_{ij} such that $s_{ij} = f_i(j)$. Later, P_i sends each s_{ij} to other participant P_j secretly, for $j = 1, 2, \cdots, n$.

Secret Reconstruction.

1) Master share generation:
Each participant P_j with n sub-shares s_{ij} for $i = 1, 2, \cdots, n$, computes the master share s'_j as $s'_j = \sum_{i=1}^{n} s_{ij} = \sum_{i=1}^{n} f_i(j)$.
2) Master secret reconstruction:
With knowledge of any k master shares $s'_1, \cdots, s'_k$, the master secret s can be reconstructed using the Lagrange interpolating formula:

$$\begin{aligned} s &= \sum_{j=1}^{k} s'_j \lambda_j \\ &= \sum_{j=1}^{k} (\sum_{i=1}^{n} s_{ij} \lambda_j) \\ &= \sum_{i=1}^{n} s_i, \end{aligned}$$

where $\lambda_j = \prod_{j \neq i} \frac{i}{i-j}$.

2.2 The Homomorphic Property

The homomorphic property of the secret sharing scheme was introduced by Benaloh [1]. We say that a SSS has the homomorphic property if the sum of the shares of two secrets s_1 and s_2 sent to the participants are shares of the sum of secrets $s_1 + s_2$. Therefore, the participants are able to recover the sum of secrets only knowing the shares from s_1 and s_2.

Let S be the domain of a secret and T be the domain of the shares corresponding to the secret. $F_A : T^k \to S$ is an induced function of the (k, n) SSS for each $A \subset \{1, 2, \cdots, n\}$ with $|A| = k$. This function defines the sub-secret s_i based on k sub-shares $s_{i1}, s_{i2}, \cdots, s_{ik}$, namely, $s_i = F_A(s_{i1}, s_{i2}, \cdots, s_{ik})$. Following theorem proves that each participant only needs to keep one master share secretly and then the master secret can be reconstructed based on any k master shares or more according to the property of additive homomorphism.

Theorem 1. *With knowledge of any k master shares or more, participants can reconstruct the master secret using Shamir's secret reconstruction algorithm according to the property of additive homomorphism.*

Proof.

$$\begin{aligned} s_1 &= F_A(s_{11}, \cdots, s_{1k}) \\ s_2 &= F_A(s_{n1}, \cdots, s_{nk}) \\ &\vdots \\ s_n &= F_A(s_{n1}, \cdots, s_{nk}). \end{aligned}$$

Then, we have

$$\begin{aligned} s &= s_1 + \cdots + s_n \\ &= F_A(s_{11}, \cdots, s_{1k}) + \cdots + F_A(s_{n1}, \cdots, s_{nk}) \\ &= F_A((s_{11} + \cdots + s_{n1}), \cdots, (s_{1k} + \cdots + s_{nk})) \\ &= F_A(\sum_{i=1}^{n} s_{i1}, \cdots, \sum_{i=1}^{n} s_{ik}) \\ &= F_A(s'_1, \cdots, s'_k). \end{aligned}$$

Using Lagrange interpolation, the master secret s can be reconstructed with k master shares $s'_1, \cdots, s'_k$ where $s'_i = \sum_{j=1}^{n} s_{ji}$ for $i = 1, \cdots, k$.

In the case of a SSS with public verifiability, we say that such a scheme has homomorphic property when, beside all above, the verification of the shares of the new secret $s_1 + s_2$ can be done from the broadcasted public information about s_1 and s_2. □

2.3 Bilinear Pairing

Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with the same prime order q where g is a generator of group $\mathbb{G}_1$. A bilinear pairing e is a function defined by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. For all $a, b \in \mathbb{Z}_q^*$, $P, Q \in \mathbb{G}_1$, we say e is an admissible bilinear map if the function e satisfies the following three conditions:

1) Bilinear: $e(g^a, g^b) = e(g, g)^{ab}$.
2) Non-degenerate: $e(g, g) \neq \mathbf{1}$.
3) Computable: $e(P, Q)$ is efficiently computable.

2.4 Related Complexity Assumptions

For security analysis of our proposed scheme, we summarize some important security problems and assumptions as follows.

- **Computational Diffie-Hellman (CDH) problem**: Given $g, g^a, g^b \in \mathbb{G}_1$ for some $a, b \in \mathbb{Z}_q^*$, the CDH problem is to compute $g^{ab} \in \mathbb{G}_1$.
- **CDH assumption**: No probabilistic polynomial time (PPT) algorithm can solve the CDH problem with a non-negligible probability.
- **Bilinear Diffie-Hellman (BDH) problem**: Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some $a, b, c \in \mathbb{Z}_q^*$, the BDH problem is to compute $e(g,g)^{abc} \in \mathbb{G}_2$.
- **BDH assumption**: No PPT algorithm can solve the BDH problem with a non-negligible probability.

3 Proposed Scheme

In this section, we present a joint random secret sharing scheme with public verifiability. Let $s_i \in \mathbb{Z}_q^*$ be a random sub-secret selected by each participant P_i for $i = 1, 2, \cdots, n$. A random secret s is recovered cooperatively by any k participants or more where $s = e(h^{\sum_{p_i \in A} s_i}, h)$ and A is the set of participants whose shares all are verified correctly.

Our scheme consists of three algorithms: share generation, share verification, and secret reconstruction. The concrete construction is illustrated as follows.

Share Generation.

1) Setup: Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with the same prime order q where q g, h are two independently generators of group $\mathbb{G}_1$. A bilinear pairing e is a function defined by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The public parameters $Param = (\mathbb{G}_1, \mathbb{G}_2, q, g, h, e)$ was agreed and published by all participants. Every participant publishes his public key $y_i = h^{x_i}$ and withholds the corresponding secret key $x_i \in \mathbb{Z}_q^*$.
2) Sub-secret generation: Each participant P_i selects a random sub-secret $s_i \in \mathbb{Z}_q^*$ independently.
3) Sub-share generation: P_i chooses a random polynomial $f_i(x) = \sum_{l=0}^{k-1} a_{il} x^l \bmod q$ where $a_{i0} = f_i(0) = s_i$, and uses Shamir's (k, n) SSS to generate sub-shares s_{ij} for other participant P_j such that $s_{ij} = f_i(j)$ for $j = 1, 2, \cdots, n$. After that, P_i broadcasts the commitments $C_{il} = g^{a_{il}}$ for $0 \le l \le k-1$. Later, he computes and publishes the encryption $Y_{ij} = (y_j)^{s_{ij}}$ of each sub-share s_{ij} to other participant P_j for $j = 1, 2, \cdots, n$.

   Finally, each participant P_j receives n encryptions Y_{ij} for $i = 1, 2, \cdots, n$.

Share Verification.

Any verifier can check whether each encryption Y_{ij} received by participant P_j are consistent with sub-share s_{ij} by means of checking the equation $e(\prod_{l=0}^{k-1} C_{il}^{j^l}, y_j) = e(g, Y_{ij})$ for $i = 1, 2, \cdots, n$.

Secret Reconstruction.

1) Master share generation: Let A be the set of participants whose shares all are verified correctly. Using his own secret key x_j, every participant P_j in the set A decrypts Y_{ij} as $Y_{ij}^{x_j{}^{-1}} = h^{s_{ij}} = h^{f_i(j)}$. Then, P_j computes $s'_j = \prod_{p_i \in A} h^{f_i(j)} = h^{\sum_{p_i \in A} f_i(j)}$ and saves s'_j as his master share.
2) Master share verification: The master share of participant $P_j \in A$ can be verified by others with the following verification equation:
$$e(s'_j, y_j) = \prod_{p_i \in A} e(Y_{ij}, h).$$
3) Master secret reconstruction: After the verification, then for an arbitrary subset $B \subseteq A$ consisting of k participants whose correct master shares have pooled, every participant in B can get master secret s by the following Lagrange interpolation:
$$\begin{aligned} s &= \prod_{j=1, p_j \in B}^{k} = e(s'_j, h)^{\lambda_j} \\ &= \prod_{j=1, p_j \in B}^{k} e(h,h)^{\sum_{p_i \in A} f_i(j)\lambda_j} \\ &= e(h,h)^{\sum_{p_i \in A}(\sum_{j=1, p_j \in B}^{k} f_i(j)\lambda_j)} \\ &= e(h,h)^{\sum_{p_i \in A} f_i(0)} \\ &= e(h,h)^{\sum_{p_i \in A} s_i}, \end{aligned}$$
   where $\lambda_j = \prod_{p_j \in B, j \neq i} \frac{i}{i-j}$.

4 The Homomorphic Property

Let $f_1(j)$ and $f_2(j)$ be the sub-shares of sub-secrets s_1 and s_2 for participant P_j, respectively. Following the idea from [1], we say that our scheme has the homomorphic property since Shamir's scheme also has it. So we have $f_1(j) + f_2(j)$ be the master share of the sum sub-secret $s_1 + s_2$. In relation to the verification process, if the elements (g, h, y_j, Y_{1j}) and (g, h, y_j, Y_{2j}) are used in the verifications of the sub-shares of s_1 and s_2, respectively, namely, $e(h^{f_1(j)}, y_j) = e(Y_{1j}, h)$ and $e(h^{f_2(j)}, y_j) = e(Y_{2j}, h)$, then it is easy to prove the homomorphic verification of master share of the sum sub-secret $s_1 + s_2$ if the equality
$$e(h^{f_1(j)+f_2(j)}, y_j) = e(Y_{1j}, h) e(Y_{2j}, h)$$
is satisfied.

Note that the property of homomorphic verification is not achieved if the protocol makes use of a typical zero knowledge proof in the verification process.

5 Scheme Analysis

5.1 Correctness

The correctness of this scheme means that:

1) A honest participant can always pass the verification procedure in both share generation phase and secret reconstruction phase;

2) At least t honest participants are always able to recover an unique master secret.

It is straightforward to check these requirements for the above protocol.

5.2 Verification of Scheme

In this section, we show that the participants in the protocol must behave honestly or will be detected. More precisely, on the one hand, the participants must be honest in the share generation phase and, on the other hand, the participants must be honest in the secret reconstruction phase.

5.2.1 Verification of the Share Generation

In the share generation phase, if P_i passes the verification procedure, then any qualified sets of k honest participants will reconstruct the same sub-secret s_i.

Theorem 2. *If P_i passes the verification process in share generation phase, then there exists a unique polynomial $f_i(x)$ such that the encrypted share of participant P_j is $Y_j = y_j^{f_i(j)}$ for $1 \le j \le n$, i.e., P_i must be honest.*

Proof. Assume that the encrypted sub-share of the participant P_j sent by P_i is equal to $Y_{ij} = y_j^{\bar{s_{ij}}}$. If P_i passes the equation $e(\prod_{l=0}^{k-1} C_{il}^{j^l}, y_j) = e(g, Y_{ij})$, then by the definition $C_{il} = g^{a_{il}}$, we follow $e(g, y_j)^{f_i(j)} = e(g, y_j)^{\bar{s_{ij}}}$, which leads to $\bar{s_{ij}} = f_i(j)$ for $1 \le j \le n$. Hence, the uniqueness of $f_i(x)$ will be reconstructed by any qualified sets of k correct sub-shares and then the same sub-secret $s_i = f_i(0)$ will be recovered. □

5.2.2 Verification of the Secret Reconstruction

In the share reconstruction phase, if some participant gives a different master share, then it means one of the sub-shares will be decrypted incorrectly. Otherwise, the master share should derive from n sub-shares decrypted correctly.

Theorem 3. *If P_j passes the verification process in the secret reconstruction phase, then for any i, $h^{f_i(j)} = Y_{ij}^{x_j{}^{-1}}$, where x_j is the secret key of P_j, i.e., P_j must be honest.*

Proof. Suppose $P_j \in A$ sends a different master share $\bar{s_j'} = h^{\sum_{p_i \in A} f_i(j)}$ where one of sub-shares $h^{f_i(j)}$ is decrypted by another secret key $\bar{x_j}$, namely, $h^{f_i(j)} = Y_{ij}^{\bar{x_j}}$. If any other participant accepts P_j's master share, then the following verification equality holds in the share reconstruction phase:

$$\begin{aligned}
e(s_j', y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
e(h^{\sum_{p_i \in A} f_i(j)}, y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(h^{f_i(j)}, y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(Y_{ij}^{\bar{x_{ij}}}, h^{x_{ij}}) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(Y_{ij}, h)^{\bar{x_{ij}} x_{ij}} &= \prod_{p_i \in A} e(Y_{ij}, h).
\end{aligned}$$

Thus, the above equality results in $\bar{x_j} x_j = 1$ and then $\bar{x_j} = x_j^{-1}$, which means that the sub-share $h^{f_i(j)}$ is decrypted correctly by his secret key x_j^{-1}. It follows that if any other participant accepts the master share of P_j, then the master share should derive from n sub-shares decrypted correctly. □

Note that in our scheme, the validity of shares can be publicly verified without leaking the privacy of shares and secret in the share verification phase. Furthermore, the verification process does not depend on any computational assumption and is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof.

The results in this section are summarized in the following theorem.

Theorem 4. *The verification process of our scheme is unconditionally secure and non-interactive.*

5.3 Security of the Scheme

In this section, we present security analysis of the proposed scheme. We first consider the security of the sub-share $h^{f_i(j)}$. Given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$, the difficulty of computing the sub-share $h^{f_i(j)}$ is equivalent to breaking the CDH assumption.

Lemma 1. *Under the CDH assumption, it is infeasible to compute the sub-shares from public information.*

Proof. By contradiction, assume that there exists an algorithm $\mathcal{A}$ without knowing $f_i(j)$, which can compute the sub-share $h^{f_i(j)}$ with a non-negligible probability ϵ for the given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. Then, there exists an attacker can solve the CDH problem using the algorithm $\mathcal{A}$. Given $\alpha = g^a, \beta = g^b$ for some $a, b \in \mathbb{Z}_p^*$, we try to compute the value g^{ab} using the capacity of $\mathcal{A}$ in the following.

At random, we pick x, y, z and feed $h = \alpha^x, y_j = h^y, X_{il} = \beta^z, Y_{ij} = h^{yz}$ to $\mathcal{A}$. Since the input to $\mathcal{A}$ is uniformly distributed, we can obtain $h^{f_i(j)} = g^{axbz}$ with

success probability ϵ because of $X_{il} = \beta^z = g^{bz} = g^{f_i(j)}$. By taking $g^{axbz/xz}$, we are thus able to compute g^{ab} with the same success probability ϵ. It is a contradiction to the CDH assumption. □

In the following, we show that participants less than k learn nothing about the secret S. In other words, if no more than $k-1$ participants can recover the secret, it implies breaking the BDH assumption.

Lemma 2. *Under the BDH assumption, it is infeasible that any $k-1$ participants can cooperatively obtain the secret in the proposed scheme.*

Proof. Recalling that the BDH problem is to compute $e(g,g)^{abc}$ for given $g, g^a, g^b, g^c \in \mathbb{G}_1$ where $a, b, c \in \mathbb{Z}_q^*$. A natural variant of the standard BDH problem is to compute $e(g,g)^{aab}$ for given g, g^a, g^b where $a, b \in \mathbb{Z}_q^*$, which is called Computational Bilinear Square (CBS) assumption [18].

By contradiction, assume that there exists an algorithm $\mathcal{A}$ without knowing all $f_i(0) \in A$, which can compute the master secret $s = e(h^{\sum_{p_i \in A} s_i}, h) = e(h^{\sum_{p_i \in A} f_i(0)}, h)$ with a non-negligible probability ϵ for the given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. Equivalently, without knowing some $f_i(0) \in A$, the algorithm $\mathcal{A}$ can compute the $e(h^{f_i(0)}, h)$. Then, there exists an attacker can solve the variant of the BDH problem using the algorithm $\mathcal{A}$.

In the following, we show how to set up the system such that we can compute $e(g,g)^{aab}$. Suppose that participants $P_1, \cdots, P_{k-1}$ are able to break the scheme. At random, we pick some $x, y, x'_j \in \mathbb{Z}_q^*$ and set $h = (g^a)^x, C_{i0} = (g^b)^y, y_j = h^{x'_j}$ for $j = 1, 2, \cdots, n$, which implicitly defines $f_i(0)$ as it required that $C_{i0} = g^{f_i(0)}$. The values $f_i(1), f_i(2), \cdots, f_i(k-1)$ are chosen at random from $\mathbb{Z}_q^*$, which fixes a polynomial $f_i(x)$. This allows us to directly compute $Y_{ij} = y_j^{f_i(j)}$ and $X_{ij} = g^{f_i(j)}$ for $j = 1, 2, \cdots, k-1$. Since $f_i(0)$ is only given implicitly, we cannot compute the values $f_i(k), f_i(k+1), \cdots, f_i(n)$. However, we can use X_{ij} for $j = 1, 2, \cdots, k-1$ to obtain C_{il} for $l = 1, 2, \cdots, k-1$ by solving $k-1$ simultaneous equations $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. When we have computed these values C_{il}, we set $Y_{ij} = (C_{i0} \prod_{l=1}^{k-1} C_{il}^{j^l})^{x'_j}$ such that $Y_{ij} = y_j^{f_i(j)}$, as required for $j = k, k+1, \cdots, n$.

The complete view for the system is now defined. It is consistent with the private view of participants $P_1, \cdots, P_{k-1}$, and comes from the right distribution. Suppose that they are able to compute the master secret $s = e(h^{\sum_{p_i \in A}}, h) = \prod_{p_i \in A} e(h,h)^{f_i(0)}$, then can compute $e(h,h)^{f_i(0)}$. Since we put $h = g^{ax}$ and $C_{i0} = g^{by}$ which implies $f_i(0) = by$, thus we are able to compute $e(h,h)^{f_i(0)} = e(g,g)^{aaxxby}$ with success probability ϵ. By taking $e(g,g)^{aaxxby/xxy}$, we are thus able to compute $e(g,g)^{aab}$ with the same success probability ϵ. It is a contradiction to the variant of the BDH assumption, i.e., CBS assumption. □

By the two lemmas above, we can show that our proposed scheme is secure.

Theorem 5. *Under the CDH assumption and BDH assumption, the proposed scheme is secure, that is, 1) only qualified participants can compute the valid sub-shares; 2) any participants less than k can not recover the master secret.*

As the proof of Lemma 1 and Lemma 2, the correctness of Theorem 5 is straightforward.

6 Performance Analysis

6.1 Computational Complexity

Our scheme consists of three phases: share generation, share verification, and secret reconstruction. For the computation cost, we only consider the "time-consuming computation", which includes modular exponentiation, modular multiplication and pairing operation in each phase.

- Share generation: This phase outputs Y_{ij}, which is encryption of sub-share for each participant, and broadcasts the commitment C_{il}. These requires $n(k+n)$ modular exponentiation .
- Share verification: In this phase, the most time-consuming computation is to verify whether sub-share Y_{ij} is consistent, which needs approximately n^2 pairing operations.
- Secret reconstruction: In this phase, the most time-consuming computation is to verify whether master share s'_j is consistent, which requires k pairing operations.

The summation of operations required in our protocol is $n(k+n)$ modular exponentiation and n^2+k pairing operations.

6.2 Comparison

In this section, we give a comparison of our protocol with those in [7, 12] in terms of computation cost, security of verification, and related properties.

As we analyzed in Introduction, the verifiability of JRSSS [7] is unconditionally secure, however, the scheme can only detect the fact of cheating but not identify who are cheaters. On the contrary, the JRSSS [12] can identify who are cheaters whereas the verification of their scheme is based on the RSA assumption. In addition, the two schemes both do not achieve public verifiability.

In [12], the most time-consuming computation is to verify whether the shares are consistent, which requires $n(n^2+n+k)$ modular exponentiation. In [7], the most time-consuming computation is to reconstruct 100 verification polynomials to verify the strong t-consistency of master shares, which requires $n100k^3$ modular multiplicative.

We denote modular multiplicative, pairing operation, and modular exponentiation by M_m, M_p and M_e, respectively. The comparison of our protocol with those in [7, 12] is shown in Table 1.

We observe that our proposed scheme achieves better properties and stronger security compared with others. Although our efficiency is somewhat lower, but as a compensate for that we first provide JRSSS with public verifiability so far. Furthermore, the verification process in our proposed scheme is unconditionally secure and non-interactive without using the Fiat-Shamir's technique or any additional zero knowledge proof.

7 Application: Dynamic Node-Join in Mobile Ad Hoc Network

In a dynamic topology network, the new nodes need to join or depart it frequently. In this section, using the techniques in our scheme we describe how to add a new node in this environment such as mobile ad hoc network(MANET). In MANET, the secret often acts as a system key and there still exists a same requirement for security, that is, any k nodes or more can recover the system key but nodes less then k lean nothing about it with their shares. After the new node becomes a legitimate member of the MANET, it will possess a share whose format is like others and share the same system key.

There exists a specific dealer to redistribute shares for a new member-join in traditional secret sharing scheme. However, this approach is infeasible in MANET since it is a distributed environment.

If a new node wishes to join the MANET, it must obtain at least k or more nodes approving admission from current MANET and then a new share can be reconstructed cooperatively by k nodes. To maintain the essential security in this process, there are two types of methods. One is to shuffle the secret sharing polynomial by regenerating a random polynomial [9, 21]. The other is to shuffle the partial share by adding blind factor [14, 17]. Nevertheless, these methods lead to a higher computation and communication cost. Since MANET is composed of limited calculation ability, communication capacity and bandwidth, more communications and computation will consume longer time which leads to lower success rate to the generation of new shares.

Herein we employ a more straightforward mechanism to conduct it by combing the techniques in our scheme with Hamiltonian ring instead of the aforemention methods. The detailed is described in Section 7.2.

7.1 Security Requirement

There are security requirement which must be reached in our node-join protocol.

1) Any k nodes or more can recover the system key but nodes less then k learn nothing about it with their shares.
2) Any information about system key cannot be exposed.
3) None but the legal node can get its new share.
4) The shares of old nodes are secure.

7.2 Concrete Protocol

Assume that B is a collaboration of k nodes in MANET, a new node v_r wants to join the MANET. Cooperating parties $v_1, \cdots, v_k \in B$ are arranged in a unclosed Hamiltonian ring for computation the new share s'_r.

1) New node v_r broadcasts/multicasts an joining request among B.
2) Each node $v_j \in B$ calculates a partial share p_j for v_r as follows.

$$p_j = s_j'^{\lambda_j(r)} = h^{\sum_{P_i \in A} f_i(j)\lambda_j(r)},$$

where s'_j is v_j's own master share and $\lambda_j(r) = \sum_{v_i \in B, j \neq i, i=1}^{k} \frac{r-i}{j-i}$. Next, each node v_j masks its private value using v_r's public key $y_r = h^{x_r}$ as follows.

$$\begin{aligned} e(p_j, y_r) &= e(h^{\sum_{P_i \in A} f_i(j)\lambda_j(r)}, h^{x_r}) \\ &= e(h,h)^{x_r \sum_{P_i \in A} f_i(j)\lambda_j(r)}. \end{aligned}$$

3) After v_1 computes $e(p_1, y_r)$, it securely sends it to the next node v_2. Upon receiving $e(p_1, y_r)$, v_2 multiplies it by $e(p_2, y_r)$, to the product $e(p_1, y_r)e(p_2, y_r)$ before sending it to the next node v_3. At the end, last node v_k receives $\prod_{j=1}^{k} e(p_j, y_r)$ and send it to the new node v_r;
4) Using its private key x_r, new node v_r decrypts the last product and obtains master share s'_r as follows.

$$\begin{aligned} s'_r &= \prod_{j=1}^{k} e(p_j, y_r)^{1/x_r} \\ &= e(h^{\sum_{j=1}^{k}(\sum_{P_i \in A} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A}(\sum_{j=1}^{k} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A}(\sum_{j=1}^{k} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A} f_i(r)}, h). \end{aligned}$$

Note that the format of new share is different from that of old one slightly. The format is $e(h^{\sum_{P_i \in A} f_i(r)}, h)$ in the former, while that is $h^{\sum_{P_i \in A} f_i(j)}$ in the latter. This does not affect the reconstruction of the system key since there is still a need to calculate $e(h^{\sum_{P_i \in A} f_i(j)}, h)$ before reconstruction in the latter.

Figure 1 shows how this computation operates. Suppose that the communication channels between $v_j \in B$ are secure. For simplicity, we leave out the proof of security in this version.

Table 1: Comparison of three protocols

	[12]	[7]	Our scheme
Computational cost	$(n^3+n^2+nk)M_e$	$(100k^3n)M_m$	$(n^2+k)M_p$
Security of verification	computational	unconditional	unconditional
Public verifiability	no	no	yes
Cheater identification	yes	no	yes
Communication channels	private	private	public

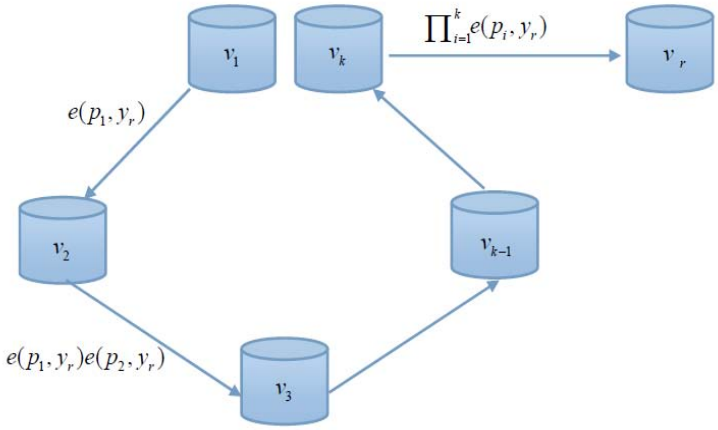

Figure 1: Dynamic Node-Join in MANET

8 Conclusions

In this paper, to the best of our knowledge, for the first time we provide a secret sharing scheme with public verifiability in distributed environment. Utilizing additive homomorphism, each participant acts as a dealer to choose the secret (sub secrets) and generate sub-shares for other participants, which avoids the need for a mutually trusted dealer. By this way, a random master secret will be constructed by some sub secrets corporately. In addition, we explore the technique of homomorphic verification and that of bilinear pairing to allow anyone to publicly verify whether the received shares are consistent. In the verification analysis, we show that the verification process is unconditional secure and non-interactive without using the Fiat-Shamir's technique or any additional zero knowledge proof, which makes it simple and efficient. Finally, we present how our techniques can be applied to handle dynamic node-join in MANET.

Acknowledgment

The authors would like to express their deep appreciation for the valuable comments provided by anonymous reviewers. This work was supported by the National Natural Science Foundation of China (Nos. 61272435, U1261114, 61472146), Guangdong Natural Science Foundation (No. S2013010011859), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2014A030306021), and Research Fund for the Doctoral Program of Xi'an University of Science and Technology (No. 2015QDJ008, 2013QDJ031).

References

[1] C. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret", in *Advances in Cryptology (CRYPTO'86)*, LNCS 263, pp. 251–260, Springer, 1987.

[2] G. Blakley, "Safeguarding cryptographic keys", *Proceedings in AFIPS National Computer Conference*, vo1. 48, pp. 313–317, 1979.

[3] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults", in *Proceedings of 26th IEEE Symposium on Foundation of Computer Science*, pp. 383–395, 1985.

[4] C. K. Chu, S. M. Chow, W. G. Tzeng, et al., "Key-aggregate cryptosystem for scalable data sharing in cloud storage", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 468–477, 2014.

[5] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing", in *Proceedings of 28th IEEE Symposium on Foundations of Computer Science*, pp. 427–437, 1987.

[6] A. Fiat, A. Shamir, "How to prove yourself: Practical solutions to identification and signature prob-

lems", in *Advances in Cryptology (CRYPTO'86)*, LNCS 263, pp. 186–194, Springer, 1986.

[7] L. Harn, C. Lin, "Strong (n, t, n) verifiable secret sharing scheme", *Information Sciences*, vol. 180, no. 16, pp. 3059–3064, 2010.

[8] S. Heidarvand, J. L. Villar, "Public verifiability from Pairings in Secret Sharing Schemes", in *Proceedings of SAC'08*, LNCS 5381, pp. 294–308, Springer, 2009.

[9] A. Herzberg, S. Jaracki, H. Krawczyk, M. Andyung, "Proactive secret sharing or: How to cope with perpetual", in *Advances in Cryptology (CRYPTO'95)*, LNCS 963, pp. 339–352, Springer, 1995.

[10] C. F. Hsu, S. Wu, L. Harn, "New results on ideal multipartite secret sharing and its applications to group communications", *Wireless Personal Communications*, vol. 82, no. 1, pp. 283–292, 2014.

[11] I. Ingemarsson, G. J. Simmons, "A protocol to set up shared secret schemes without the assistance of a mutually trusted party", in *Advances in Cryptology (EUROCRYPT'90)*, LNCS 473, pp. 266–282, Springer, 1991.

[12] K. Kaya, A. A. Selcuk, " A verifiable secret sharing scheme based on the chinese remainder theorem", in *Proceedings of Progress in Cryptology (INDOCRYPT'08)*, pp. 414–425, 2008.

[13] I. Komargodski, M. Naor, E. Yogev, "Secret-sharing for NP", in *Advances in Cryptology (ASIACRYPT'14)*, LNCS 8874, pp. 254–273, Springer, 2014.

[14] J. Kong, P. Zerfos, H. Luo, S. Lu, L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks", in *Proceedings of The Ninth International Conference on Network Protocols*, pp. 251–260, 2001.

[15] P. Kun, "Threshold distributed access control with public verification: A practical application of PVSS", *International Journal of Information Security*, vol. 11, no. 1, pp. 23–31, 2012.

[16] K. Kurosawa, "General error decodable sharing scheme and its application", *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6304–6309, 2011.

[17] L. C. Li, R. S. Liu, "Securing cluster-based ad hoc networks with distributed authorities", *IEEE Transactions on Wireless Communications*, vol. 9, no. 10, pp. 3072–3081, 2010.

[18] B. Libert, D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy reencryption", in *Proceeding sof Public Key Cryptography (PKC'08)*, LNCS 4939, pp. 360–379, Springer, 2008.

[19] Y. X. Liu, L. Harn, C. N. Yang, Y. Q. Zhang, "Efficient (n, t, n) secret sharing schemes", *Journal of Systems and Software*, vol. 85, no. 6, pp. 1325–1332, 2012.

[20] Q. Al Mahmoud, "Polynomial differential-based strong (n, t, n) verifiable secret sharing", *IET Information Security*, vol. 7, no. 4, 312–317, 2013.

[21] X. Y. Meng, Y. M. Li, "A verifiable dynamic threshold key management scheme based on bilinear pairing without a trusted party in mobile ad hoc network", in *Proceedings of IEEE International Conference on Automation and Logistics*, pp. 315–320, 2012.

[22] T. P. Pederson, "Non-interactive and information-theotetic secure verifiable secret sharing", in *Advances in Cryptology (CRYPTO'91)*, LNCS 576, pp. 129–140, Springer, 1991.

[23] T. P. Pedersen, "A threshold cryptosystem without a trusted party", in *Advances in Cryptology (EUROCRYPT'91)*, LNCS 547, pp. 522–526, Springer, 1991.

[24] A. Shamir, "How to share a secret", *Communications of ACM*, vol. 33, no. 3, pp. 612–613, 1979.

[25] M. Stadler, "Publicly verifiable secret sharing", in *Advances in Cryptology (EUROCRYPT'96)*, LNCS 1070, pp. 190–199, Springer, 1996.

Zhenhua Chen received her Ph. D. degree from Shaanxi Normal University in 2014. She is currently a associate professor at School of Computer Science and Technology, Xi'an University of Science and Technology. Her research interests include secure multi-party computation and secret sharing.

Shundong Li received his Ph. D. degree from Xi'an Jiaotong University in 2003, and had been studied in Tsinghua University as Postdoctor from 2003 to 2005. From 2005 to 2007, he had been a associate professor at Beijing Normal University. He is currently a professor and supervisor of Ph.D. at Shaanxi Normal University. His research interests focus on secure multi-party computation and confidential data mining.

Qiong Huang received his Ph.D. degree from City University of Hong Kong in 2010. He is now a professor with College of Informatics, South China Agricultural University. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis.

Jianhua Yan is now a PH.D. candidate of Beijing university of posts and telecommunications. His research interests focus on lattice-based cryptography and the security of cloud.

Yong Ding received his Ph. D. degree from Xidian University in 2005, and had been studied in City University of Hongkong as Research Fellow from 2008 to 2009. He is currently a professor and assistant dean at School of Mathematic and Computing Science, Guilin University of Electronic Technology. His research interests focus on cryptography and information security.