A Strong PVSS Scheme

Fuyou Miao, Xianchang Du, Wenjing Ruan and Suwan Wang

Abstract A verifiable secret sharing allows the dealer to share a share with some participants and can be verified the correctness. Stadler extended the idea and presented a notion of Public Verifiable Secret Scheme (PVSS), which has the property that any one can verify the validity of the share from the sender, but neglects the deceit of receiver. In this paper, we first give a formal definition of strong PVSS (SPVSS) based on PVSS and propose a specific SPVSS scheme. The scheme solves the cheating problems on both sides, which has strong verification requirements. Finally, we show our scheme satisfies the definition of strong PVSS.

Keywords Secret sharing · Secret security · Publicly verifiable · Diffie-Hellman

1 Introduction

The Secret Sharing (SS) scheme was first introduced by Blakley [1] and Shamir [2] independently in 1979. The SS shows that a secret is divided into n shares shared among n participants by a dealer. In this way, any legal subset of the shareholders (access structure) can recover the secret. SS is a method for the storage of secrets, which can be used to preserve key and avoid attacks from the adversary.

Shamir's [2] scheme is based on the threshold method implemented by interpolating polynomial. The (t, n) threshold scheme has been widely applied in secret sharing. A (t, n) threshold scheme has two properties, that is, (1) any t or more than t shareholders can reconstruct the secret in cooperation; (2) but fewer than t can't

F. Miao (✉) · X. Du · W. Ruan
School of Computer Science, University of Science and Technology,
Hefei, People's Republic of China
e-mail: mfy@ustc.edu.cn

S. Wang
School of Computer Science and Technology, University of Anhui, Hefei, China

N. Chaki et al. (eds.), *Computer Networks & Communications (NetCom)*,
Lecture Notes in Electrical Engineering 131, DOI: 10.1007/978-1-4614-6154-8_51,

reconstruct the secret. The (t, n) threshold scheme with polynomial is simple and efficient, whose computational complexity is $o(n \log^2 n)$.

In 1985, Chor [3] extended the notion of secret sharing and proposed a definition of verifiable secret sharing (VSS). VSS was introduced to protect against the cheating of dishonest participants. In contrast with traditional secret sharing, a verifiable protocol was added, which was used to verify the validity of shares. That is because that there are probably some malicious participants or dishonest dealers to send incorrect shares mainly in the process of secret distribution and reconstruction. Feldman [4] and Pederson [5] respectively proposed a VSS scheme based on Shamir's [2] scheme and they can effectively detect cheating of participants or dealers. If the VSS scheme can be verified by any party, it has public verifiability, which is called Public Verifiable Secret Sharing (PVSS) introduced by Stadler [6]. Stadler expressed that not only the participants, but everyone else is also able to verify whether the shares have been correctly distributed. Schoenmakers [7] extended the scheme, he required that in the reconstruction process the participants can provide a proof of correctness for each share. The approach is simple and secure computationally. Subsequently Young-Yung [8] proposed an improvement on [7] and presented a PVSS for discrete logs based on the difficulty in computing discrete logs, compared to [7], which is based on the difficulty in deciding Diffie-Hellman. Later Behnad and Eghlidos [9] introduced a new PVSS. They added two different processes: One is disputation used in the case of a complaint against the dealer. The other one called membership proof is used to authenticate the membership of shareholders. The scheme is simpler and can solve the complaint from the participants and verify its validity.

PVSS has a wide range of applications, such as, electronic voting, software key escrow and revocable electronic cash. In previous PVSS, each participant can verify the correctness of shares from the dealer in the process of distribution. Also in the reconstruction process, a shareholder can verify the validity of a share from another shareholder. But the identity of the share provider is unknowable. The share provided to the third verifier is probably incorrect. For example, assuming that a participant distributes one correct share to the participant, but the participant is a cheater, and he announces that the share from sender is invalid. So it's important to guarantee the correctness of information from the dealer or participant, as to detect the deceiving of both sides. In the electronic voting protocol, a voter is regarded as the dealer and talliers are participants. With the PVSS scheme, a voter must provide public proof, but the tallier is probably a cheater, and he is likely to make a false complaint and provide an invalid share to the third party.

In this paper, we extend the idea of PVSS and give a formal definition of strong PVSS (SPVSS), in which both participants must provide the proof of correctness to the third party. This scheme involves Behnad's [9] scheme as the basis for both sharing and reconstruction. In such a way, we add two processes to the traditional PVSS. One in the distribution process, called Distribution-Check, is used to check the validity of shares from the dealer and cheating by the participant. We will show that by this process, a dealer can not only prove to the third party the correctness of a share sent to the participant who complained, but the participant can also prove to the third party that his complaint is reasonable and the share is not modified. The other

one, called Reconstruction-Check, is used to check who is the cheater. We prove that our proposed scheme satisfies the definition of SPVSS and meet higher security requirements.

The security of our scheme is based on the Diffie-Hellman problem which is difficult computationally. In our scheme, once a complaint happens, the cheater can be detected. Compared to Behnad's [9] scheme, this scheme does not make use of membership proof and is simpler and more secure [10, 11].

The rest of the paper is organized as follows: In Sect. 2, we briefly describe the concept and characteristics of PVSS. In Sect. 3, we define new notion of strong PVSS and present our SPVSS scheme. Finally, we discuss the security and performance of the new scheme.

2 Review of PVSS

2.1 The Model of PVSS

We will present an informal description of PVSS in Staler [6]

A PVSS scheme consists of a dealer, n participants, $P_1, \ldots P_n$ and an access structure $A \subseteq 2^{\{1,2\ldots,n\}}$, The access structure is monotonous, which means that if A $\in A$ and A $\subseteq$ B then B $\in A$. S is the secret being shared.

A public encryption function E_i is assigned to each participant P_i, such that only he has the corresponding decryption function D_i. The dealer sends S_i to P_i by calculating $S_i = E_i(s_i), i = 1, 2 \ldots n$ and publishing the encrypted share S_i in the share distribution process. Then the algorithm PubVerify is assigned to verify the validity of the encrypted shares. The algorithm has the property that $\exists u \forall A \in 2^{\{1,2\cdots n\}}$:

$$(PubVerify(\{S_i | i \in A\}) = 1) \Rightarrow \text{Recover}(\{D_i(S_i) | i \in A\}) = u$$

if the dealer is honest, $u = s$. The algorithm can be run by any party. If the secret needs to be recovered, an algorithm Recover will be run, which has the feature that

$$\forall \text{A} \in A : \text{Recover}(\{S_i | i \in \text{A}\}) = s$$

and that for all A$\notin A$ it is computationally infeasible to calculate s from $\{\text{S}_\text{i}|\text{i} \in \text{A}\}$.

2.2 The Property of PVSS

Firstly, the PVSS scheme has the general property of a VSS scheme, that is, Verification and Secrecy. The receiver can verify the validity of their received shares. The secrecy means that any group in the access structure should not receive a share and recover the secret.

Secondly, another unique property of PVSS is Publicity. The participant or any other party can verify the validity of other participants (with whom they might be able to recover the secret).

3 Definition of Strong PVSS and a Specific Scheme

A PVSS scheme enables a judge to deal with the conflict. But the identities of participants both are uncertain. In other words, the participants both are not credible. Both sides have the probability of behaving fraudulently. We improve the notion and propose new strong publicly verifiable secret sharing that can ensure a higher level of security.

3.1 The Definition of Strong PVSS

Definition 1 Both participants in a strong PVSS scheme can be verified to demonstrate the validity of data provided by them.

In the SPVSS scheme, we add the verification process, S_i' is share provided by each participant P_i, the SPVSS is described as follow: $PubVerify(\{S_i \&\& S_i' | i \in A\} = 1)$

$$\text{Then}(PubVerify(\{S_i' | i \in A\}) = 1) \Rightarrow \text{Recover}(\{D_i(S_i') | i \in A\}) = u$$

3.2 Strong PVSS Scheme

Notation
Throughout this paper the chosen parameters p and q denote large primes such that q divides $p - 1$. Z_p and Z_q are two different fields. G_p is the unique subgroup of Z_p^* of prime order p, and g denotes a generator of G_p.

SPVSS scheme
This SPVSS scheme is based on Shamir's [2] (t, n) threshold scheme. In our scheme, if the share from the dealer is not approved. In the distribution process, then the Distribution-Check protocol is run by other participants or any other party to check the conflict. Similarly, in the Reconstruction process, if a complaint occurs, the reconstruction-check protocol is used to investigate the complaint and check the cheater.

3.2.1 Distribution

The process consists of two steps:

(1) **Distribution of shares**:

The dealer randomly chooses F polynomial of degree at most $t-1$ with coefficients in Z_q.

$$F(x) = F_0 + F_1 x + \cdots + F_{t-1} x^{t-1}$$

And sets $F_0 = s$ which is the secret. The dealer publishes $C_i = g^{F_i}, i = 0, 1, \cdots, t-1$. The participant j, $j = 1, \cdots, n$ registers g^{a_j} as his public key to the dealer, where $a_j \in Z_q$. Then the dealer also publishes g^d as his public key, where $d \in Z_q$. The dealer publishes the encrypted shares $Y_i = s_i (g^{a_i})^d, i = 1, \cdots, n$, where $s_i = F(i)$, using the public keys of the participants.

(2) **Verification of shares**:

The shareholder i decrypts the Y_i using a_i by computing $s_i = Y_i ((g^d)^{a_i})^{-1}$, then, verifies the share s_i by computing $g^{s_i} = \prod_{j=0}^{t-1} C_j^{i^j}$. If the equation does not hold, then the participant complains against the dealer.

3.2.2 Distribution-Check

If the shareholder A complains against the dealer D, the third party R will run to verify the validity of the complaint and vote against the dealer D or the shareholder A by the following protocol. Firstly, we assume that the public secret of A and D is g^a and g^d and the published encrypted share is $Y_A = s_A (g^a)^d$.

(1) **Verification of shareholder**:

1. R chooses randomly $r \in Z_q$, calculates g^r and sends to shareholder A;
2. A calculates $\lambda = (g^r)^{g^{ad}}$ and sends back to R;
3. R then sends λ to the dealer D;
4. D calculates $(g^{ad})^{-1}$ and $\eta = \lambda^{(g^{ad})^{-1}}$, then publish η.
5. Both R and A verify if $\eta = g^r$. If it holds, the protocol is continued. Otherwise, A and D respectively provide the values a and d to R, R verifies the dishonest party by calculating g^a, g^d and λ.
6. R randomly chooses another value $t \in Z_q$, calculates $\sigma = \lambda^{r^{-1}t}$ and sends to A;
7. A calculates σ^{s_A} and sends back to R;
8. R verifies if $(\sigma^{s_A})^{t^{-1}} = g^{Y_A}$, if it holds, it says that the share provided by shareholder is true and it is not modified. Then the protocol checks the dealer's identity.

(2) **Verification of dealer**

9. R chooses $s \in Z_q$, and publishes $\rho = g^{s s_A} g^{ab}$;
10. A and D independently calculate $\delta = \rho^{(g^{ab})^{-1}}$ and send the results to R;
11. R verifies if the two values received are equal, if they are, the protocol is continued. Otherwise, A and D send the values a and d to R to check the cheat;

12. R checks if $\delta^{s^{-1}} = \prod_{i=0}^{t-1} C_i^{j_A{}^i}$, if the equality holds, then s_A is correct, else it is not valid and the dealer is a cheater.

3.3 Reconstruction

The process consists of two steps;

3.3.1 Verification of Share

When at least t shareholders will cooperate to recover the secret, there will be a shareholder sending share to another shareholder,this protocol can be run to verify the correctness of the share.

a and b respectively are shareholder A and shareholder B's private key. Shareholder A encrypts the share s_A by calculating $Y_{AB} = s_A g^{ab}$, and publishes the value.

Shareholder B decrypts the encrypted share by calculating $s_A = Y_{AB}(g^{ab})^{-1}$, and verifies if $g^{s_A} = \prod_{i=0}^{t-1} C_i^{j_A^i}$, if it holds, the share provided by shareholder A is correct. Else the following protocol reconstruction-check is run.

Reconstruction-check

1. The shareholder B sends g^{s_A} calculated to R;
2. R sends g^{s_A} to the shareholder A, and A calculates $\alpha = (g^{s_A})^{g^{ab}}$ sends back to R;
3. B then sends $\beta = (g^{s_A})^{g^{ab}}$ to R, R verifies if $\alpha = \beta$, if it holds then the protocol is continued, else A and send a and to R;
4. R verifies if $\alpha = \beta = (g^{s_A})^{g^{ab}}$ if the equation does not hold, then we can be sure B is a cheater, else the protocol is continued;
5. R verifies if $g^{s_A} = \prod_{i=0}^{t-1} C_i^{j_A^i}$, if it does not hold, then A is a cheater;

3.3.2 Recovering the Secret

When all the shares received are correct, then the secret is reconstructed as follows:

$$s = \sum_{i=1}^{t} w_i s_i$$

where $w_i = \prod_{j \neq i} \frac{i}{j-1}$ is a lagrange coefficient.

4 Security

Our scheme adds two new processes to the protocol, Distribution-Check and Reconstruction-Check. The two processes are both used to check which party is the cheater once a complaint happens. This mainly reflects two aspects:

- Participant A is a cheater and sends party B an invalid share, then B reports an error;
- A sends B the correct share , but B is a cheater, then B also reports an error;

Hence, the security of the protocol is based on the security of the two stages. Distribution-check is run when there is a complaint against the dealer.

Lemma 1 The shareholder A cannot commit an invalid and modified share.

Proof By verifying $(\sigma^{s_A})^{t^{-1}} = g^{Y_A}$ in step 8 of the Distribution-Check stage, R is able to believe the share provided by the shareholder is consistent with what was sent by the dealer, and the shareholder does not modify the share. Firstly, if the shareholder provides the invalid $(g^{ab})'$ to replace λ with λ' in step 2. Then during step 5, the value η' received will be $(g^r)^{(g^{ab^{-1}})^{g^{ab'}}}$ and $\eta' \neq \eta$. At this time, A and D have to provide the value a and d to verify the fact. If A modifies the share s_A to s'_A, and it must guarantees $(g^t)^{s_A g^{ab}} = (g^t)^{s'_A g^{ab}}$. Even if A can solve the discrete logarithm, he must compute $ts_A g^{ab}$ from $ts'_A g^{ab}$, which is impossible, because the value t is unknown.

Lemma 2 the dealer cannot send an invalid share to the participant

Proof By verifying $\delta_A = \delta_D$ in step 10, R is convinced that A and D agree on the common key. Due to the random value s and the difficulty of solving the discrete logarithm, the dealer D cannot forge an invalid δ_D to make the equation $\delta_D^{s^{-1}} = \delta^{s^{-1}}$.

Lemma 3 1. The shareholder B cannot send an invalid share from shareholder A; 2. The share received by shareholder B cannot be invalid;

Proof

1. Firstly, the shareholder B cannot send a false value $(g^{ab})'$ such that the equation $(g^{sA})^{g^{ab}} = (g^{sA})^{(g^{ab})'}$ holds. This is a discrete logarithm problem. Similarly, B also cannot forge a value s'_A to deceive R to make $(g^{s_A})^{g^{ab}} = (g^{s'_A})^{g^{ab}}$, this is computationally impossible. Therefore, the shareholder must provide a correct share except that he can solve the discrete logarithm problem.
2. By verifying the equality in step 5, R is convinced that B sent an encrypted correct share. On the other hand, from step 2, R was convinced that A would have the same share value.

5 Conclusion

In this paper, we extend the basic definition of PVSS and give a formal definition of SPVSS. We show that traditional PVSS scheme can't ensure the cheat of both sides. We develope a SPVSS scheme, in which we stress that each party's behave can be tested. If one party is malicious, our scheme can detect it and determine the complaint. Compared to previous schemes, our scheme adds two stages: (1) Distribution-Check, where the dealer and every shareholder can both prove the correctness of the share to other parties; (2) Reconstruction-Check phase, where each shareholder can prove to other parties the correctness of a share to reconstruct the secret. We prove that our scheme satisfy the security of PVSS and have a higher level of security.

Acknowledgments The paper is supported by NSF of China (granted no. 60970128, No. 61170233, No. 61272472), Youth Innovation Foundation of USTC of 2010 and China Postdoctoral Science Foundation (No. 2011M501397).

References

1. Blakley GR (1979) Safeguarding cryptographic keys. In: Proceedings of the national computer conference 1979, vol 48. American Federation of Information Processing Societies, pp 313–317
2. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
3. Chor B, Goldwasser S, Micali S, Awerbuch B (1985) Verifiable secret sharing and achieving simultaneity in the presence of faults. In: 26th annual symposium on foundation of computer, science, pp 383–395
4. Feldman P (1987) A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th IEEE symposium on foundations of computer science, pp 427–437
5. Pederson TP (1992) Non-interactive and information-theotetic secure verifiable secret sharing. In: Advances in cryptology-CRYPTO'91, LNVS576, pp 129–140
6. Stadler M (1996) Publicly verifiable secret sharing. In: Advance in cryptology-EUROCRYPT'96. LNCS 1070:190–199
7. Schoenmakers B (1999) A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: CRYPTO'99. LNCS 1666:148–164
8. Young A, Yung M (2001) A PVSS as hard as discrete log and shareholder separability. In: PKC 2001, LNCS 1992, 2001, pp 287–299
9. Behnad A, Eghlidos T (2008) A new publicly verifiable secret sharing scheme. Scientia Iranica 15(2):246–251
10. Fujisaki E, Okamoto T (1998) A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In: Eurocrypt'98, LNCS 1403, 1998, pp 32–46
11. Chor B, Goldwasser S, Micali S, Awerbuch B (1985) Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Proceedings of 26th annual symposium on foundations of computer science 1985, pp 383–395