Non-Interactive and Information-Theoretic Secure Publicly Verifiable Secret Sharing

Chunming Tang[1,2,*] Dingyi Pei[1,3] Zhuojun Liu[2] Yong He[4]
1 Dept of Mathematics and Information Science, Guangzhou University, P.R.China
2 Academy of Mathematics and systems science, Chinese Academy of Sciences, P.R.China
3 State Key Laboratory of Information Security, Chinese Academy of Sciences, P.R.China
4 Computer Science and Engineering Academy, Hunan Science and Technology University, P.R.China

Abstract

A publicly verifiable secret sharing scheme is more applicable than a verifiable secret sharing because of the property that the validity of the shares distributed by the dealer can be verified by any party. In this paper, we construct a non-interactive and information-theoretic publicly verifiable secret sharing by a computationally binding and unconditionally hiding commitment scheme and zero-knowledge proof of knowledge.

Keywords: secret sharing, publicly verifiable secret sharing, information-theoretic, commitment, proof of knowledge.

1 Introduction

Secret sharing, which is a network of $n+1$ players consisting of n participants with identities $1, 2, ..., n$ and a dealer(in some cases, the dealer is also a participant), is an important primitive in cryptology. It usually consists of two basic protocols: (i) a *distribution* protocol in which the secret is distributed by the dealer among participants, and (ii) a *reconstruction* protocol in which the secret is recovered by pooling the shares of a qualified subset of the participants. Basic schemes(threshold secret sharing) solve the problem for the case that all players in the scheme are honest[1, 2].

However, if a player or some players are dishonest, basic secret sharing schemes will not be in practice because of the following reasons: (i) all participants can not verify validity of their shares from the dealer in the *distribution* protocol, and (ii) participant can not check validity of their shares from other participants in the *reconstruction* protocol. In order to resist malicious players, Feldman[8] and Pedersen[9] constructed a new type secret sharing scheme respectively. They are called verifiable secret sharing(VSS) schemes in which the following cases are ture (i)the dealer cannot send invalid shares to some or all of the participants during the distribution protocol, and (ii)participants cannot submit invalid shares during the reconstruction protocol.

Pedersen's VSS scheme is more applicable than Feldman's, because the former is information-theoretic secure, however, the latter is only computationally secure.

For all VSS schemes, they are two drawbacks in some cases(such as, electronic voting): (i) the participants can only verify their own share, but anybody can not verify that the participants received correct shares. (ii) the participants simply release their shares in the *reconstruction* protocol, subsequently the released shares may be verified by anybody against the output of the distribution protocol. In order to deal with them, publicly verifiable secret sharing(PVSS) schemes were proposed by Stadler[7] and Schoenmakers[3] respectively. Stadler's PVSS schemes

*Email: ctang@mmrc.iss.ac.cn

can only conquer the first drawback, that is, the participants can not only verify their own share, but also anybody can verify that the participants received correct shares. However, anybody else is not able to verify the validity of the participant's share if the participant does not provide a proof of correctness for share released by himself in Schoenmakers' PVSS scheme(in fact, the proof is the proof of knowledge which two discrete logarithm is equal), so his scheme can avoid two drawbacks.

In whole paper, Let G_q denote a group of prime order q, such that computing discrete logarithms in this group is infeasible. Let $g, g_1, g_2, h, h_1, h_2, G, H$ denote independently selected generators of G_q, hence no party knows the discrete logarithm of any two generators. In addition, we denote a secure $hash$ function by $H(\cdot)$.

In Schoenmakers' scheme, the following commitment scheme is used: $c_a := g^a$ where $a \in_R \mathbb{Z}_q$, subsequently his scheme is only computationally secure[3, 8]. In this paper, we will construct an information-theoretic secure PVSS scheme by Pedersen's commitment scheme[9] and proof of knowledge that expressions of X and Y to base g_1, g_2 and h_1, h_2 respectively are equal, furthermore, our PVSS scheme can also avoid two above drawbacks.

The structure of this paper is following, we introduce Pedersen's commitment scheme and PVSS model in section 2 and in section 3 respectively. In section 4, we construct several zero-knowledge proof of knowledge. In section 5, we propose a PVSS scheme and prove its security.

2 Commitment Scheme

Pederson[9] proposed a computationally binding and unconditionally hiding scheme based on the discrete logarithm problem. A value $\alpha \in \mathbb{Z}_q$ is committed to as $C_\alpha := g^\alpha h^r$, where r is randomly chosen from $\mathbb{Z}_q$. We will use this commitment scheme for our construction and hence they will be statistical zero-knowledge proof of knowledge.

3 Model for Non-interactive PVSS

We note that a distinctive feature of PVSS is that no private channels between the dealer and the participants are assumed.

In a PVSS scheme, a dealer D wishes to distribute shares of a secret value $s \in \Sigma$ among n participants $P_1, ..., P_n$. A monotone access structure describes which subsets of participants are qualified to recover the secret. For example, the access structure may be a (t, n)-threshold schemes, $1 \leq t \leq n$, which means that any subset of t or more participants will able to recover the secret, unless commitment is broken.

As a common structure for PVSS schemes we consider the following protocols. Note that initialization is done without any interaction between the dealer and the participants. In fact, participants may enter or leaver the system dynamically; the only requirement is that a participant holds a registered public key.

Initialization All system parameters are generated as part of the initialization. Furthermore, each participant P_i registers a public key key to be used with a public key encryption method E_i. The actual set of participants taking part in a run of the PVSS scheme must be a subset of the registered participants. We assume w.l.o.g that participants $P_1, ..., P_n$ are the actual participants in the run described below.

Distribution The protocol consists of two steps:

1. *Distribution of the shares.* The distribution of a secret $s \in \Sigma$ is performed by the dealer D. The dealer first generates the respective shares s_i for participant P_i, for $i = 1, ..., n$. For each participant P_i the dealer publishes the encrypted share $E_i(s_i)$. The dealer also publishes a string $PROOF_D$ to show that each E_i encrypts a share s_i. Furthermore, the

string $PROOF_D$ commits the dealer to the value of secret s, and it guarantees that the reconstruction protocol will result in the same value s.

2. *Verification of the shares.* Any party knowing the public keys for the encryption methods E_i may verify the shares. For each participant P_i a non-interactive verification algorithm can be run on $PROOF_D$ to verify that $E_i(s_i)$ is a correct encryption of a share for P_i. Since anyone may verify a share, it may be ruled out that a participant complains while it received a correct share. In case one or more verifications fail, we therefore say that the dealer fails, and the protocol is aborted. (If some level of fault-tolerance is desired one may continue and think of it as a $(t, n-c)$-threshold scheme, where c is the number of verifications that failed.)

Reconstruction The protocol consists of two steps:

1. *Decryption of the shares.* The participants decrypt their shares s_i from $E_i(s_i)$. It is not required that all participants succeed in doing so, as long as a qualified set of participants is successful. These participants release s_i plus a string $PROOF_{P_i}$ that shows that the released share is correct.

2. *Pooling the shares.* The strings $PROOF_{P_i}$ are used to exclude the participants which are dishonest or fail to reproduce their share s_i correctly. Reconstruction of the secret s can be done from the shares of any qualified set of participants.

Compared to [7], we have added the requirement for the reconstruction protocol that the participants must provide a proof of correct decryption of their shares. The proof is also non-interactive so that any party is able to sort out the correct shares and pool then together.

We have limited the description to non-interactive PVSS schemes by requiring that all $PROOF$s can be verified non-interactively. In fact, it is natural to reduce the amount of interaction between the players even more than for VSS schemes. Non-interactive VSS schemes, such as [8, 9], still include a stage in which participants file complaints if they received an incorrect share. Subsequently these complaints must be resolved to decide whether the distribution of the secret was successful. In non-interactive PVSS we have eliminated even this round of interaction: since any party can verify the output of the dealer, there is no need for the individual participants to check their own shares!

4 Proof of Knowledge

Chaum and Pedersen have ever proposed a protocol for proving that $log_{g_1} h_1 = log_{g_2} h_2$. We denote this protocol by $DLEQ(g_1, h_1, g_2, h_2)$, and it consists of the following steps, where the prover knows x such that $h_1 = g_1^x$ and $h_2 = g_2^x$:

1. The prover chooses a random $r \in \mathbb{Z}_q^*$, and sends $\alpha_1 = g_1^r$ and $\alpha_2 = g_2^r$ to the verifier.
2. The verifier sends a random challenge $c \in_R \mathbb{Z}_q$ to the prover.
3. The prover responds with $s := r - cx (mod\ q)$.
4. The verifier checks $\alpha_1 = g_1^s h_1^c$ and $\alpha_2 = g_2^s h_2^c$.

Now, we will generalize Chaum and Pedersen's result and present a protocol for proving that $X = g_1^{x_1} g_2^{x_2}$ and $Y = h_1^{x_1} h_2^{x_2}$. This protocol is denoted by $DLEQ(X, Y, g_1, g_2, h_1, h_2)$ and consists of the following steps, where the prover knows x_1, x_2 such that $X = g_1^{x_1} g_2^{x_2}$ and $Y = h_1^{x_1} h_2^{x_2}$.

1. The prover chooses a random $r_1, r_2 \in \mathbb{Z}_q^*$, and sends $t_1 = g_1^{r_1} g_2^{r_2}$ and $t_2 = h_1^{r_1} h_2^{r_2}$ to the verifier.

2. The verifier sends a random challenge $c \in_R \mathbb{Z}_q$ to the prover.
3. The prover responds with $s_i := r_i - cx_i (mod\ q)$, where i=1,2.
4. The verifier checks $t_1 = X^c g_1^{s_1} g_2^{s_2}$ and $t_2 = Y^c h_1^{s_1} h_2^{s_2}$.

In [5], Chaum and Pedersen have proven their protocol $DLEQ(g_1, h_1, g_2, h_2)$ satisfying zero-knowledge, and in [4], Chaum, Evertse and Graaf proposed a zero-knowledge protocol for $y = \prod_{i=1}^{l} g_i^{x_i}$. So, Protocol $DLEQ(X, Y, g_1, g_2, h_1, h_2)$ is a zero-knowledge protocol also.

Protocols $DLEQ(g_1, h_1, g_2, h_2)$ and $DLEQ(X, Y, g_1, g_2, h_1, h_2)$ are interactive, so they cannot be directly used in our non-interactive secret sharing scheme. However, If the random challenge c in above two protocols is replaced by a secure *hash* function value with message m as input, a signature of non-interactive zero-knowledge proof of knowledge can be constructed, and this signature will be used in our non-interactive and information-theoretic publicly verifiable secret sharing scheme.

In [6], Fiat and Shamir constructed a signature of non-interactive zero-knowledge proof of knowledge for $DLEQ(g_1, h_1, g_2, h_2)$. We denote the signature by $Sign(g_1, h_1, g_2, h_2)$.

1. The prover chooses a random $s \in \mathbb{Z}_q$, computes $h_1' := g_1^s$, $h_2' := g_2^s$, $c = H(m||g_1||g_2||h_1||h_2||h_1'||h_2')$, $r = cx + s(mod\ q)$, then sends h_1', h_2', r to the verifier.
2. The verifier computes $c := H(m||g_1||g_2||h_1||h_2||h_1'||h_2')$, and check $g_1^r := h_1^c h_1'$, $g_2^r = h_2^c h_2'$.

The following signature is a signature of non-interactive zero-knowledge proof of knowledge for $DLEQ(X, Y, g_1, g_2, h_1, h_2)$, and it is denoted by $Sign(X, Y, g_1, g_2, h_1, h_2)$.

1. The prover chooses two random $s, t \in \mathbb{Z}_q$, computes $X' := g_1^s g_2^t$, $Y' := h_1^s h_2^t$, $c = H(m||g_1||g_2||h_1||h_2||X||X'||Y||Y')$, $r_1 = cx_1 + s(mod\ q)$, $r_2 = cx_2 + t(mod\ q)$, then sends X', Y', r_1, r_2 to the verifier.
2. The verifier computes $c := H(m||g_1||g_2||h_1||h_2||X||X'||Y||Y')$, and check $g_1^{r_1} g_2^{r_2} := X^c X'$, $h_1^{r_1} h_2^{r_2} = Y^c Y'$.

5 Special PVSS Scheme

Basing on signature $Sign(g_1, h_1, g_2, h_2)$, Schoenmakers proposed a simple PVSS which it has some distinguishing features, however, his scheme is only computational secure[3]. In this section, we will propose an information-theoretic secure PVSS basing on $Sign(X, Y, g_1, g_2, h_1, h_2)$.

5.1 Protocols

Just like Schoenmakers, we also solve the problem of efficiently sharing a random value from G_q. The dealer will achieve this by first selecting $s_1, s_2 \in_R \mathbb{Z}_q$ and then distributing shares of the secret $S = G^{s_1} H^{s_2}$. This approach allows us to keep the required proofs simple and efficient.

Initialization The group G_q and the generators g, h, G, H are selected using an appropriate public procedure. Participant P_i generates a private key $x_i \in_R \mathbb{Z}_q^*$ and registers $y_{i1} = G^{x_i}$, $y_{i2} = H^{x_i}$ as its public key.

Distribution The protocol consists of two steps:

1. *Distribution of the shares.* The dealer picks two random polynomials $f(x)$ and $g(x)$ of degree at most $t-1$ with coefficients in $\mathbb{Z}_q$:

$$f(x) = \sum_{j=0}^{t-1} \alpha_j x^j,$$

$$g(x) = \sum_{j=0}^{t-1} \beta_j x^j$$

and sets $s_1 = \alpha_0, s_2 = \beta_0$. The dealer keeps these polynomials secret but publishes the related commitments $C_j = g^{\alpha_j} h^{\beta_j}$, for $0 \leq j < t$. The dealer also publishes the encrypted shares $Y_i = y_{i1}^{f(i)} y_{i2}^{g(i)}$, for $0 \leq j \leq n$, using the public keys of the participants. Finally, let $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$. The dealer shows that the encrypted shares are consistent by producing a proof of knowledge of the unique $f(i), g(i)$, $1 \leq i \leq n$, satisfying:

$$X_i = g^{f(i)} h^{g(i)}, \quad Y_i = y_{i1}^{f(i)} y_{i2}^{g(i)}.$$

The non-interactive proof is the n-fold parallel composition of the protocols for $DLEQ(X_i, Y_i, g, h, y_{i1}, y_{i2})$. Applying $Sign(X_i, Y_i, g, h, y_{i1}, y_{i2})$, the challenge c of non-interactive proof is computed as a cryptographic hash of $g, h, G, H, y_{i1}, y_{i2}, X_i, Y_i, a_{1i}, a_{2i}, 1 \leq i \leq n$. The proof consists of the common challenge c and $2n$ responses $r_{i1}, r_{i2}, 1 \leq i \leq n$.

2. *Verification of the shares.* The verifier computes $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$ from the C_j values. Using $y_{i1}, y_{i2}, X_i, Y_i, r_{i1}, r_{i2}, 1 \leq i \leq n$ and c as input, the verifier computes a_{1i}, a_{2i} as

$$a_{1i} = g^{r_{i1}} h^{r_{i2}} X_i^c, \quad a_{2i} = y_{i1}^{r_{i1}} y_{i2}^{r_{i2}} Y_i^c,$$

and checks that the hash of $g, h, G, H, y_{i1}, y_{i2}, X_i, Y_i, a_{1i}, a_{2i}, 1 \leq i \leq n$, matches c.

Reconstruction The protocol consists of two steps:

1. *Decryption of the shares.* Using its private key x_i, each participant finds the share $S_i = G^{f(i)} H^{g(i)}$ from Y_i by computing $S_i = Y_i^{1/x_i}$. They publish S_i plus a proof that the value S_i is a correct decryption of Y_i. To this end it suffices to prove knowledge of an α such that $y_i = y_{i1} y_{i2} = (GH)^\alpha$ and $Y_i = S_i^\alpha$, which is accomplished by the non-interactive version of the protocol $DELQ(GH, y_i, S_i, Y_i)$.

2. *Pooling the shares.* Suppose w.l.o.g that participants P_i produce correct values for S_i, for $i = 1, ..., t$. The secret $G^{s_1} H^{s_2}$ is obtained by Language interpolation:

$$\prod_{i=1}^{t} S_i^{\lambda_i} = \prod_{i=1}^{t} (G^{f(i)} H^{g(i)})^{\lambda_i} = G^{\sum_{i=1}^{t} f(i)\lambda_i} H^{\sum_{i=1}^{t} g(i)\lambda_i} = G^{f(0)} H^{g(0)} = G^{s_1} H^{s_2},$$

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

Note that the participants do not need nor learn the values of the exponents $f(i), g(i)$. Only the related values $S_i = G^{f(i)} H^{g(i)}$ are required to complete the reconstruction of the secret values $S = G^{s_1} H^{s_2}$. Also, note that participant P_i does not expose its private key x_i; consequently participant P_i can use its key pair in several runs of the PVSS schemes.

5.2 Efficiency Analysis

The dealer only needs to post $t+n$ elements of G_q(the numbers C_j and Y_i) plus $2n+1$ number of size $|q|$(the responses r_{i1}, r_{i2} and the challenge c). The number of exponentiations throughout the protocol is correspondingly low, and all of these exponentiations are relatively small exponents from $\mathbb{Z}_q(|q| = 160$ bits in pratice.

5.3 Security Analysis

We first recall *Diffie-Hellman* problem and *Diffie-Hellman* assumption.

Definition 1 *It is recalled* **Diffie-Hellman problem** *to find g^{ab} given g^a and g^b in group G_q, where $a, b \in_R \mathbb{Z}_q$.*

If the inputs to algorithm A are g^a and g^b which are randomly generated, and the output of it is g^{ab} with non-negligible success probability, we think that algorithm A solves *Diffie-Hellman* problem about g.

Assumption 1 *(***Diffie-Hellman assumption***) For any positive polynomial $p(\cdot)$ and probabilistic polynomial-time algorithm A, it exists an integer k_0 such that the following inequality holds for any $k > k_0$:*

$$Prob[A(G_q, g, g^a, g^b) = g^{ab} | a, b \in \mathbb{Z}_q] < \frac{1}{p(k)}$$

i.e., it does not exist polynomial-time algorithm A which can successfully resolve Diffie-Hellman problem with non-negligible probability.

It is necessary to simplify our scheme in order to evaluate security of our PVSS scheme. In the following proof, we will assume in our PVSS scheme that 1) $g = h$; 2) $G = H$; 3)$f(x) = g(x)$ hold. If our simplified PVSS scheme is secure, our original PVSS scheme will also be secure.

Now, we consider the security of the share-encryptions. We observe that directly breaking the encryptions used in our PVSS scheme implies breaking Assumption 1. This can be seen as follows. Breaking the encryption of the shares amounts to finding $G^{f(i)}H^{g(i)}$ given $g, h, G, H, X_i, y_{i1}, y_{i2}, Y_i$, for the group G_q. In the simplified model, Breaking the encryption of the shares amounts to finding $G^{2f(i)}$ given g, G, X_i, y_{i1} *or* y_{i2}, Y_i, for the group G_q. Writing $G = g^{\alpha}, X_i = g^{2\beta}, y_i = g^{\lambda}$, breaking the encryption of the shares is equivalent to computing $g^{2\alpha\beta}$, given g^{α}, $g^{2\beta}$, g^{λ}, and $g^{2\beta\lambda}$, for $\alpha, \beta, \lambda \in_R \mathbb{Z}_q$. Recalling that Assumption 1 states that it is infeasible to compute $g^{2\alpha\beta}$, given g^{α} and $g^{2\beta}$, we have the following lemma.

Lemma 1 *Under the Diffie-Hellman assumption, it is infeasible to break the encryption of the shares in our simplified PVSS model.*

Proof: Given $x = g^{\alpha}$ and $y = g^{\beta}$, we want to obtain $z = g^{\alpha\beta}$ by using an algorithm $\mathbb{A}$ that breaks the encryption of the shares. Pick random α', β', γ, and feed $x^{\alpha'}, y^{2\beta'}, g^{\gamma}, g^{2\beta'\gamma}$ to $\mathbb{A}$. Since the input to $\mathbb{A}$ is uniformly distributed, we then obtain $Z' = g^{2\alpha\alpha'\beta\beta'}$ with some success probability ϵ. By taking $z'^{1/(2\alpha'\beta')} = g^{\alpha\beta}$, we are thus able to compute z with same success probability ϵ. $\blacksquare$

A stronger result is that the secret is protected unless t or more participants cooperate. This is expressed by the following lemma.

Lemma 2 *Suppose that $t-1$ participants pool their shares and obtain the secret in our simplified PVSS model. Then we can break the Diffie-Hellman assumption.*

Proof: Let g^{α} and g^{β} be given, so we want to obtain $g^{\alpha\beta}$. We assume that α and β are random; if not, it is trivial to adapt the proof, as in the previous lemma. Suppose w.l.o.g. that participants $P_1, ..., P_{t-1}$ are able to break the scheme. We will show how to set up the system such that this fact enables us to compute $g^{\alpha\beta}$.

We put $G = g^{\alpha}$ and $C_0 = g^{2\beta}$, which implicitly defines $f(0)$ as it is required that $C_0 = g^{2f(0)}$. The points $f(1), ..., f(t-1)$ are chosen at random from $\mathbb{Z}_q$, which fixes polynomial $f(x)$. This allows us to directly compute $X_i = g^{2f(i)}$ and $Y_i = y_{i1}^{2p(i)}$, for $i = 1, ..., t-1$. Since $f(0)$ is only given implicitly, we cannot compute the points $f(t), ..., f(n)$. It suffices, however, that we can compute $X_i = g^{2f(i)}$ by Lagrange interpolation, which also yields the remaining C_j's. We now deviate from the protocol by computing the public keys y_{i1} *or* y_{i2} of participants $P_i, i = t, ..., n$, as $y_i = g^{w_i}$ for random $w_i \in \mathbb{Z}_q$, and we set $Y_i = X_i^{w_i}$ such that $Y_i = y_i^{2f(i)}$, as required.

The complete view for the system is now defined. It is consistent with the private view of participants $P_1, ..., P_{t-1}$, and comes from the right distribution. Now, suppose that they are able to compute the secret $G^{2f(0)}$. Since $G = g^{\alpha}$ and $f(0) = \beta$, we are thus able to compute $g^{2\alpha\beta}$. This contradicts the *Diffie-Hellman* assumption. ∎

Note that we are assuming a static adversary. The above argument may be extended to the case where the static adversary is allowed to take part in the PVSS protocols K times, i.e., before breaking it. In that case we follow the protocol for the first K runs except that for participants $P_t, ..., P_n$ we will set $S_i = G^{2f(i)}$ directly instead of decrypting Y_i.

So far we have ignored the proofs that are required at several points in the protocol. However, in the random oracle model these proofs can easily be simulated. This leads to the following summary.

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

Proof: It follows from the soundness of the Chaum-Pedersen[**?**] proof and the fact that the X_i's are obtained from the C_j's as $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$ that the shares of the participants are consistent with the secret. It follows from Lemma 2 and the fact that the Chaum-Pedersen proof is honest-verifier zero-knowledge that no set of less that t participants can recover the secret. ∎

Theorem 1 does not claim that the participants cannot get any partial information on the secret G^{2s}. This stronger result holds under the assumption that ELGamal encryption is semantically secure, which is known to be equivalent to the Decision DH assumption. The latter assumption states that it is infeasible to determine whether a given triple is of the form $(g^{\alpha}, g^{\beta}, g^{\alpha\beta})$ or $(g^{\alpha}, g^{\beta}, g^{\delta})$ for random α, β, δ.

The above results are easily adapted to this case. For the equivalent of Lemma 1 we reason as follows. Suppose that an adversary is able to determine whether an encrypted share is equal to a given value g^{δ} or not. We then obtain a contradiction with the Decision DH assumption, closely following Lemma 1, by setting $G = g^{\alpha}, X_i = g^{2\beta}$, and for random γ, setting $y_i = g^{\gamma}$ and $Y_i = (X_i)^{\gamma} = g^{2\beta\gamma}$. Since the share is equal to $G^{2\beta} = g^{2\alpha\beta}$ it follows that we are able to distinguish $g^{2\alpha\beta}$ from g^{δ}, if the adversary is able to distinguish the share from g^{delta}. The equivalent of Lemma 2 can proved in a similar way. This leads to the following conclusion.

Theorem 2 *Under the DDH assumption and the random oracle assumption, our simplified PVSS scheme is secure. That is, (i) the reconstruction protocol results in the secret distributed by the dealer for any qualified set of participants, (ii) any non-qualified set of participants is not able to recover any(partial) information on the secret.*

The simplified PVSS scheme is a special model of protocols constructed in § 5.1. Lemma 1, Lemma 2, Theorem 1 and Theorem 2 hold in the simplified model, then, they also hold in the model § 5.1, i.e., the following theorem holds:

Theorem 3 *Under the DDH assumption and the random oracle assumption, the PVSS scheme in* § 5.1 *is secure. That is, (i) the reconstruction protocol results in the secret distributed by the dealer for any qualified set of participants, (ii) any non-qualified set of participants is not able to recover any(partial) information on the secret.*

Remark 1: In fact, proof of our scheme's security is just like to that of Schoenmakers scheme's security, where Lemma 1, Lemma 2, Theorem 1 and Theorem 2 also hold in Schoenmakers scheme.

5.4 Our PVSS vs. Schoenmakers' PVSS

In our PVSS scheme, the dealer publishes related commitments $C_j = g^{\alpha_j} h^{\beta_j}$ and $Y_j = y_{i1}^{f(i)} y_{i2}^{g(i)}$, however, the dealer publishes $C_j = g^{\alpha_j}$ and $Y_j = y_i^{f(i)}$ in Schoenmakers' scheme. We find that the comparision between them is like to that between Feldman's VSS scheme and Pedersen's VSS scheme. So Schonmakers PVSS is only computationally secure, however, our scheme is information-theoretic secure.

References

[1] A Shamir, How to Share a Secret. Comm. ACM, 22(1979), 612-613.

[2] G.R Blakey, Safeguarding Cryptographic Keys. Proc. NCC, 48(1979), 313-317.

[3] B. Schoenmakers, A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting. Advances in Cryptology-CRYPTO'99, (1999)148-164.

[4] D. Chaum, J.H. Evertse, and J van de Graaf, An improved protocol for demonstrating possession of discrete logarithms and some generalizations, Advances in Crypto-EUROCRYPT'87, (1988)127-141.

[5] D. Chaum and T.P. Pedersen, Transferred cash grows in size. Advances in Cryptology-EUROCRYPT'92, (1993)390-407.

[6] A. Fiat, and A. Shamir, How to Prove Yourself: Practical Solution to Identification and Signature Problems. Advances In CRYPTO'86, (1987)186-189.

[7] M. Stadler, Publicly Verifiable Secret Sharing. Advances in Cryptology-EUROCRYPT'96, (1996)190-199.

[8] P. Feldman, A Practical Scheme for Non-interactive Verifiable Secret Sharing. Proceedings of the 28 IEEE Symposium on Foundation of Computer Science (FOCS), IEEE, (1987)427–437.

[9] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing. Advances in Cryptology-CRYPTO'91, (1992)129-140.