A New PVSS Scheme with a Simple Encryption Function

Assia Ben Shil
LIP2
Tunis, Tunisia
Faculty of Sciences of Tunis
University of El Manar
essia.benshil@gmail.com

Kaouther Blibech
kaouther.blibech@gmail.com

Riadh Robbana
LIP2
Tunis, Tunisia
riadh.robbana@fst.rnu.tn

Wafa Neji
neji.wafa@yahoo.fr

A Publicly Verifiable Secret Sharing (PVSS) scheme allows anyone to verify the validity of the shares computed and distributed by a dealer. The idea of PVSS was introduced by Stadler in [18] where he presented a PVSS scheme based on Discrete Logarithm. Later, several PVSS schemes were proposed. In [2], Behnad and Eghlidos present an interesting PVSS scheme with explicit membership and disputation processes. In this paper, we present a new PVSS having the advantage of being simpler while offering the same features.

1 Introduction

A secret sharing scheme is a cryptographic method allowing splitting a secret between a set of participants such that only some predefined subsets of participants can recover the shared secret. These qualified subsets are called access structures. A secret sharing scheme proceeds in two phases: a dealing phase in which a dealer computes shares and gives to every participant his own share and a reconstruction phase that consists in trying to reconstruct the shared secret by pooling the elements of a qualified subset of shares.

Secret sharing schemes were introduced firstly and independently by Shamir [16] and Blakley [3]. The first scheme is based on polynomial interpolation while the latter is based on hyperplane geometry. Most of the proposed secret sharing schemes [1, 11] are based on Shamir's secret sharing scheme. Although its efficiency, Shamir's scheme still presents some problems. In fact, there is an absolute trust in the dealer. This latter, can distribute some inconsistent shares leading the participants to recover a secret which differs from the initial one. Verifiable Secret Sharing (VSS) schemes [5, 6, 13] were proposed to allow participants to verify the validity of the shares they received from the dealer. However, a malicious shareholder can receive a valid share but submit an invalid one in the reconstruction phase. Publicly Verifiable Secret Sharing (PVSS) schemes [2, 4], [8-10], [14, 15], [17-21] were proposed to solve this problem. In fact, PVSS schemes were proposed to prevent cheating by the dealer or/and the shareholders. In a PVSS scheme, the validity of the distributed shares can be verified by anyone.

In [2], Behnad and Eghlidos present an interesting PVSS scheme where participants can prove their membership and the validity of their shares to prevent unauthorized parties from participating in the reconstruction process. Moreover, their scheme offers an explicit disputation process aiming to prove to a third party in conflict situations between the dealer and a participant who among them is lying.

Adel Bouhoula, Tetsuo Ida and Fairouz Kamareddine (Eds.):
Symbolic Computation in Software Science 2012 (SCSS2012)
EPTCS 122, 2013, pp. 11–22, doi:10.4204/EPTCS.122.2

In this paper, we present a new PVSS scheme providing a disputation and a membership proof processes. We show that our PVSS scheme is simpler than the PVSS scheme presented in [2] while still being as secure as the mentioned scheme.

This paper is organized as follows: First, PVSS schemes are presented. After that, our new PVSS scheme is introduced. Then, the security of our PVSS scheme is studied and a comparison between it and the previous PVSS schemes is done. Finally, we provide some concluding remarks.

2 PVSS Schemes

PVSS schemes as introduced by Stadler in [18] aim to allow anyone, not only participants, to verify that shares were correctly distributed by the dealer. This property has been defined by Stadler in [18] and has been denoted public verifiability.

Stadler proposed in this paper, two PVSS schemes that can be used with general access structures. The first one is used for sharing a discrete logarithm. It requires a non standard assumption called DDLP "Double Discrete Logarithm Assumption". In fact, Stadler dealt with expressions of the form $y = g^{(h^x)}$ (with g a generator of a group of order p, and h a fixed element of high order in Z_p^*) such that given y, it is hard to find x. Under this assumption, his scheme is as secure as the Decisional-Diffie-Hellman problem. The second one is based on the RSA root problem. It is used for sharing the n-th root and depends on the RSA assumption. Encryptions are based on a variant of the Diffie-Hellman key-exchange protocol. But we should notice here that the security of this scheme was not formally studied. Moreover, the verification in these two schemes requires information exchanges between the verifier and the shareholder. We say that it is an interactive verification.

In [8], Fujisaki and Okamoto defined the non-interactivity for a PVSS scheme as the fact that the verification of a share can be done without communicating with the dealer or with any other participant. The scheme they proposed in [8] depends on the "modified RSA assumption" assuming that inverting the RSA function is still hard. This modified RSA assumption allows partial recovery.

Notice that the schemes of [8, 18] depend on some non standard assumptions. However, Schoenmakers provided in [15] a stronger PVSS scheme by adding the fact that when submitting his share, the shareholder must provide its correctness proof. His PVSS scheme is simpler than the previous schemes. It uses techniques working in any group for which the Discrete Logarithm Problem is hard. This scheme is as hard to break as the Decisional Diffie-Hellman problem.

In [20], Young and Yung proposed an improvement of Schoenmakers's PVSS scheme. The scheme they proposed to share discrete logarithm is as hard as the Discrete Logarithm Problem itself. They proved in [21] that their scheme is computational zero-knowledge. In addition, in PVSS schemes, secure encryption assumptions are employed. But in their scheme, Young and Yung can use any probabilistic encryption function.

In [4], Boudot and Traoré proposed new PVSS schemes allowing shareholders to recover their shares quickly (fast recovery) or after a predetermined amount of computations (delayed recovery). In fact, they provide a PVSS scheme for sharing discrete logarithm with fast recovery and a PVSS scheme for sharing factorization with fast recovery. They also present a PVSS scheme for sharing discrete logarithm with delayed recovery and a PVSS scheme for sharing factorization with delayed recovery.

In most of the existing PVSS schemes, the verification phase is interactive. This is due to the use of Fiat-Shamir zero knowledge protocol [7]. In [14], Ruiz and Villar proposed a PVSS scheme with non interactive verification. It is the first efficient PVSS that does not use the Fiat-Shamir technique. It is based on the homomorphic properties of Paillier's encryption scheme [12]. It is the first known PVSS

scheme based on the DCRA[1] (Decisional Composite Residuosity Assumption). The verification process in this scheme is simpler than in the other known schemes.

In [9] , Heidarvand and Villar proposed a new PVSS scheme based on pairing. They took back the scheme of Shoenmakers using the pairing. The security of this scheme is based on the DBSDH [2] problem (Decisional Bilinear Square Diffie-Hellman problem). In [10], Jhanwar proposed a new non-interactive PVSS scheme based on pairing. In this scheme, the dealer has not to compute and to distribute the shares of a given secret; he provides a set of private keys for participants. Then, every participant uses his private key, joined to another public value to compute his share.

Recently, other PVSS schemes have been proposed. In [21], Yu and all proposed a publicly verifiable secret sharing scheme with the possibility of enrollment. In [19], Wu and all proposed a pairing based PVSS scheme reducing the computation cost while keeping the same security level of the existing public key systems.

Behnad and Eghlidos provided, in [2], a PVSS scheme with non interactive verification and having two peculiarities. First, after distributing the shares and in case of any complaint from any participant, a third party can run a disputation process to identify who is lying. This third party can then vote against the dealer or against the participant. Second, Behnad and Eghlidos added a membership proof process in the beginning of the reconstruction phase. In this phase a shareholder has to prove his membership and the validity of his share at the same time. In [17], Ben Shil, Blibech and Robbana proposed another PVSS scheme with a disputation and a membership proof processes. In this scheme, rather than publishing the encrypted coefficients of the polynomial used to compute the shares, the encrypted shares are published. Thus, the set of shares is public and any insertion or deletion will be detected by all the old participants. This scheme is, then, recommended for applications where the number of participants is limited while the access structure is dynamic and where it is worthy to keep a track of any change in the set of participants.

In this paper we introduce a new PVSS scheme providing a non-interactive verification process and presenting explicit disputation and membership processes. We show that our PVSS scheme is simpler than the schemes proposed in [2] and [17] while keeping the same level of security.

3 A new PVSS scheme

In our scheme, given two large prime numbers p and q such that $q|p-1$[3], the following notations are used:

- G_q is a subgroup of prime order q in Z_p^*, such that computing discrete logarithm in this group is infeasible and $g \in G_q$ is a generator of the group.

In our PVSS scheme, we perform all the computations in Z_q.

[1]The Decisional Composite Residuosity Assumption, used in the proof of the Paillier cryptosystem, says that given an integer z and a composite n, it is hard to decide whether z is a n-residue modulo n^2 or not.

[2]Let $e : G_1 * G_1 \to G_2$ a bilinear application such that G_1 and G_2 are two multiplicative group with the same order p. Let g be a generator of G_1 and a, b and z elements of Z_p^*. The Decisional Bilinear Square Diffie-Hellman (DBSDH) problem says that g^a, g^b and $e(g,g)^z$ is hard to decide whether $e(g,g)^{a^2b} = e(g,g)^z$.

[3]q divides p-1.

3.1 Dealing phase

3.1.1 Distribution process

In the distribution process, the dealer sets $F(x) = F_0 + F_1 x + ... + F_{k-1}x^{k-1}$, where $F_1, \ldots, F_{k-1} \in_R$ [4] Zq and F_0 is the secret to share. Moreover:

1. Every participant chooses a private key a_i where $a_i \in_R Zq$ and publishes g^{a_i} as his public key, for $1 \leq i \leq n$ where n is the number of participants.
2. The dealer D computes the shares $s_i = F(i)$, for $0 \leq i \leq n$.
3. He publishes $C_j = g^{F_j}$, for $0 \leq j \leq k-1$ and g^{s_i}, for $0 \leq i \leq n$.
4. He sends an encrypted share $E_i = s_i \oplus (g^{a_i})^{s_i}$ to the participant Pr_i, for $1 \leq i \leq n$ (Notice that s_0 is the secret and thus there is no associated encrypted share to be sent to anyone).

3.1.2 Verification process

Every shareholder Pr_i, computes $s_i = E_i \oplus [(g^{s_i})^{a_i}]$, then, verifies the following equality[5]: $g^{s_i} = \prod_{j=0}^{k-1}(C_j)^{i^j}$. Otherwise, the shareholder complains against the dealer.

3.1.3 Disputation process

In the case of any complaint, both the dealer D and the shareholder Pr_i try to prove their honesty to a third party R. For doing that, D has to publish an encrypted value leading Pr_i to extract g^{s_i} and to verify the validity of the associated share s_i. If D sends an invalid share, Pr_i has to prove this fact to R. This process is done using the following protocol:

1. Pr_i chooses his private key a_i and publishes his public key g^{a_i}.
2. Pr_i and D publish independently $g^{[(g^{a_i})^{s_i}]^{-1}}$. Then, R verifies that D and Pr_i published the same value. Else, Pr_i sends a_i to R. R computes g^{a_i} and $g^{[(g^{a_i})^{s_i}]^{-1}}$ in order to discover who is lying. Notice that R can compute g^{s_i} from the published values $g^{s_i} = \prod_{j=0}^{k-1}(C_j)^{i^j}$.
3. D computes and publishes $\lambda = s_i \oplus (g^{a_i})^{s_i}$.
4. Pr_i computes $\alpha = \lambda \oplus (g^{a_i})^{s_i}$. If $g^{\alpha} = \prod_{j=0}^{k-1}(C_j)^{i^j}$, he sends a commitment to R and the disputation process is stopped. Else, he sends α to R.
5. R computes g^{α} and verifies that $g^{\alpha} \neq \prod_{j=0}^{k-1}(C_j)^{i^j}$. Then, he verifies that $g^{1/(\lambda \oplus \alpha)} = g^{[(g^{a_i})^{s_i}]^{-1}}$. If it holds, D lied else Pr_i lied.

3.2 Reconstruction phase

3.2.1 Membership only proof

If a verifier wants to verify that Pr_i is an authorized participant, this latter has to prove his membership to the verifier without revealing his share. Our membership proof is the following:

[4] Randomly chosen.

[5] Given $C_j = g^{F_j}$, we compute:
$\prod_{j=0}^{k-1}(C_j)^{i^j} = \prod_{j=0}^{k-1}(g^{F_j})^{i^j} = \prod_{j=0}^{k-1} g^{F_j * i^j} = g^{\sum_{j=0}^{k-1} F_j * i^j} = g^{F(i)} = g^{s_i}$,(since $s_i = F(i)$).

1. The verifier chooses $a \in_R Zq$ and sends g^a to the prover.
2. The prover sends $R_P = g^{[(g^a)^{s_i}]^{-1}}$ to the verifier.
3. The verifier computes $R_V = g^{[(g^{s_i})^a]^{-1}}$ ($g^{s_i} = \prod_{j=0}^{k-1}(C_j)^{i^j}$).
4. If $R_V = R_P$, the prover is the shareholder who possesses the share s_i.

3.2.2 Pooling the shares

The secret is reconstructed from the submitted shares, as follows: $s = \sum_{i=1}^{k} w_i s_i$ where $w_i = \sum_{i \neq j} i/(j-1)$.

Notice that the shares can be submitted using the same encryption function of the distribution process ($E_i = s_i \oplus (g^{s_i})^a$) where a is the private key of the party concerned by the reconstruction of the secret and g^a is its public key.

Notice also that this party does not need to run the membership process before the pooling phase since using this encryption function allows the verification of a share and its extraction at the same time.

4 Security

In this section, we prove the security properties of our PVSS scheme. First of all, we provide our definition of a secure PVSS scheme:

Definition A PVSS scheme is secure if and only if:

- During the dealing phase, neither the dealer D can cheat by sending an invalid share to a given participant Pr_i, nor the participant Pr_i can claim that he received a non valid share while it was.
- During the reconstruction phase, an unauthorized party cannot pretend to be a shareholder.
- During all the stages of the scheme, the secrecy property is verified.

Let's prove at first that, in our scheme, the dealer D cannot cheat by sending an invalid share to the participant Pr_i. We show here that Pr_i can prove this fact to the third party R in the disputation phase. Thus, we prove the following lemma:

Lemma 4.1 *"The dealer D cannot cheat by sending an invalid share to the participant Pr_i".*

Proof In the disputation phase, a honest dealer has to compute $\lambda = s_i \oplus (g^{a_i})^{s_i}$. But a malicious dealer can have another behavior. In fact, he can compute λ using an invalid share s_i' or an incorrect value $g^{a_i'}$ rather than the public key g^{a_i} of the participant Pr_i.

So, there are seven values of λ that D can use: $\lambda = s_i' \oplus (g^{a_i})^{s_i}$ or $\lambda = s_i \oplus (g^{a_i})^{s_i'}$ or $\lambda = s_i' \oplus (g^{a_i})^{s_i'}$ or $\lambda = s_i \oplus (g^{a_i'})^{s_i}$ or $\lambda = s_i' \oplus (g^{a_i'})^{s_i}$ or $\lambda = s_i \oplus (g^{a_i'})^{s_i'}$ or $\lambda = s_i' \oplus (g^{a_i'})^{s_i'}$.

In each of these cases, Pr_i will compute $\alpha = \lambda \oplus (g^{s_i})^{a_i}$ at step 3 of the disputation process, and since $\lambda \neq s_i \oplus (g^{a_i})^{s_i}$, he will find $\alpha \neq s_i$ and he will send this value to R.

R will verify that $g^{\alpha} \neq \prod_{j=0}^{k-1}(C_j)^{i^j}$ and that $g^{1/(\lambda \oplus \alpha)} = g^{1/(\lambda \oplus \lambda \oplus (g^{s_i})^{a_i})} = g^{[(g^{a_i})^{s_i}]^{-1}}$. So R will conclude that D lied.

We prove also that, in our scheme, a malicious behavior of a participant Pr_i, who received a valid share from the dealer D, but claims that his share is invalid, will be detected. We show here that, in the disputation phase, the dealer D can prove to a third party R that Pr_i cheated. Thus, we prove the following lemma:

Lemma 4.2 *"The participant Pr_i, cannot claim that he received a non valid share while it was".*

Proof In the disputation phase, if a participant Pr_i received a correct share s_i but claims that he received an invalid one, he has to send a fake value α' to R. In fact, Pr_i computes $\alpha = \lambda \oplus (g^{s_i})^{a_i}$ but sends α' $\neq \alpha$ to R. So, R computes, $g^{\alpha'}$ and verifies that it is not a public value. Then, R verifies, at step 5 of the disputation process, that $g^{1/(\lambda \oplus \alpha')} \neq g^{\,[(g^{a_i})^{s_i}]^{-1}}$. Since it does hold, R concludes that Pr_i lied.

In addition, we prove the following lemma:

Lemma 4.3 *"Under the Computational Diffie-Hellman assumption, it is infeasible to break the encryption of the shares".*

Proof Breaking the encryption of the shares is equivalent to computing s_i from the encrypted share $E_i = s_i \oplus (g^{a_i})^{s_i}$.

To be able to do that, we have to compute $s_i = E_i \oplus (g^{a_i})^{s_i}$ from the inputs E_i, g^{a_i}, g^{s_i}. This implies computing $g^{a_i * s_i}$ given g^{a_i} and g^{s_i}.

Recall that the Computational Diffie-Hellman assumption states that it is infeasible to compute $g^{a_i * s_i}$ given g^{a_i} and g^{s_i}. Therefore the unauthorized party is not able to compute the share s_i.

Furthermore, to break the encryption of a share s_i, the adversary should be able to compute s_i from g^{s_i}. This implies solving the Discrete Logarithm Problem.

Given that computing the discrete log in G_q is infeasible, the unauthorized party is not able to compute s_i from g^{s_i}.

Then, we prove the following lemma:

Lemma 4.4 *"Under the Computational Diffie-Hellman assumption, an unauthorized party cannot extract the share s_i from g^{a_i}, g^{s_i} and the published masked value λ in the disputation process".*

Proof To extract the share s_i, the adversary has to compute s_i from the public masked value $\lambda = s_i \oplus g^{a_i * s_i}$. This implies that he needs to compute $s_i = \lambda \oplus g^{a_i * s_i}$ given λ, g^{a_i} and g^{s_i}.

For doing that, the adversary should be able to compute $g^{a_i * s_i}$ from the inputs g^{a_i} and g^{s_i}. However, the adversary is not able to compute s_i due to the Computational Diffie-Hellman assumption.

We prove also that:

Lemma 4.5 *"Under the Computational Diffie-Hellman assumption, an unauthorized party cannot retrieve the share s_i from g^{a_i}, g^{s_i} and $g^{[(g^{s_i})^{a_i}]^{-1}}$ in the two first steps of the disputation process".*

Proof Under the assumption that computing Discrete Logarithm in G_q is hard, an unauthorized party cannot extract $g^{a_i * s_i}$ from $g^{[(g^{s_i})^{a_i}]^{-1}}$ and under the Computational Diffie-Hellman assumption, it is not possible to retrieve s_i from $g^{s_i}$and g^{a_i}.

Moreover, we prove that:

Lemma 4.6 *"Under the Computational Diffie-Hellman assumption, an unauthorized party cannot pretend to be a shareholder".*

Proof This feature is fulfilled within the membership process. In this process, to pretend to be the shareholder possessing s_i, the unauthorized party should be able to compute $(g^{a_i})^{s_i}$ from the values g^{a_i} and g^{s_i} in the membership process. However, under the Computational Diffie-Hellman assumption, this is infeasible.

Finally, we prove that:

Lemma 4.7 *"Under the Computational Diffie-Hellman assumption, it is infeasible to break the encryption of the shares submitted in the reconstruction phase".*

Proof In the reconstruction phase, only the party possessing the private key a can extract the share s_i from the encrypted value $E_i = s_i \oplus (g^a)^{s_i}$. This party has just to compute $s_i = E_i \oplus [(g^{s_i})^a]$.

For a dishonest party knowing only E_i, g^a and g^{s_i}, breaking the encryption of the shares means computing $(g^a)^{s_i}$ from the public value g^{s_i} and the public key g^a which is infeasible under the Computational Diffie-Hellman assumption.

In this section, we proved that neither the dealer can cheat by distributing invalid shares nor a dishonest participant can cheat by claiming that the share he received is not valid while it was. Moreover, we proved that under the Computational Diffie-Hellman assumption, no one can break the encryption of the shares neither in the distribution process, nor in the disputation process or in the reconstruction phase. We proved also that, under the Computational Diffie-Hellman assumption, an unauthorized party cannot pretend to be a shareholder possessing a valid share.

In the following section, we compare our new PVSS scheme to the PVSS schemes presented in section 2.

5 Comparison with previous PVSS schemes

In this section, in order to compare our PVSS scheme to the existent PVSS schemes, we first present the different security properties of the most known schemes. We point that the schemes proposed in [12] and [18] do not appear in this section because we consider that these schemes have a specific context[6]. However, we include the scheme of Feldman [6] in this comparison since we consider that it is the first PVSS scheme, although public verifiability was not defined yet when this scheme was proposed. So, for each studied PVSS scheme, we identify the cryptographic techniques it uses in every process (distribution, verification...) and we verify if they satisfy our definition of security. Since most of the used cryptographic techniques are based on some hard problems, we classify these hard problems into four classes:

- Discrete Logarithms: Hard problems based on the Discrete Logarithm Problem.
- Factoring : Hard problems based on the Factorization Problem.
- Paillier's cryptosystem: Hard problems based on the Paillier's cryptosystem proof.
- Pairings: Hard problems based on the Bilinear Pairings.

As we said before, a comparison is done for every process of PVSS schemes. For the distribution process, we study the security assumptions (DLP[7], CDH[8], ...) of the encryption functions used to encrypt the shares before distributing them among the set of participants. Then, we evaluate the problem on which the security of the process is based. The evaluation is based on the following reduction: ELGamal $\leq_P$ CDH $\leq_P$ DLP.

However, for the scheme of Feldman and the scheme of Young and Yung, this evaluation is infeasible, because the cryptographic techniques used in these schemes are not specified. For more details, see table1.

[6]process fast or delayed, and the scheme proposed in [21] focused on how to make a new member join the scheme without exposing the secret and the old shares.

[7]Discrete Logarithm Problem.

[8]Computational Diffie-Hellman Problem.

Encryption and distribution of shares			
Category	**PVSS Scheme**	**Problem**	**Evaluation**
Discrete Log	Stadler (1996)	ELGamal cryptosystem	Hard
	Schoenmakers (1999)	DLP	Very hard
	Behnad & Eghlidos (2008)	CDH	Hard
	Heidarvand &Villar (2009)	DLP	Very hard
	Jhanwar (2010)	DLP	Very hard
	Ben Shil, Blibech & Robbana (2011)	CDH	Hard
	Our PVSS (2012)	CDH	Hard
Factoring	Okamoto & Fujisaki (1998)	Modified RSA assumption	Non Proved
Paillier cryptosystem	Ruiz & Villar (2005)	Paillier probabilistic encryption scheme	Hard
Pairings	Wu & Tseng (2011)	BDH	Hard
Non specified problem	Feldman (1987)	No encryption function	-
	Young & Yung (2001)	Public key encryption algorithm	-

Table 1: Evaluation of the distribution process

For the verification process, we explicit also the problem on which the security of the verification process is based. This evaluation is based on the following reductions:

- ELGamal $\leq_P$ CDH $\leq_P$ DLP.
- RSA $\leq_P$Factoring.

We also classify the verification process into two classes: interactive verification and non-interactive verification. The verification is interactive if the verifier has to communicate with other participants and/or with the dealer to verify the validity of a share. It is non-interactive if the verifier can verify the validity of a share without any communication with other participants or with the dealer. Obviously, non-interactivity is preferred in order to reduce communications. For more details, see table 4.

After the verification process, a participant can initiate a disputation process to complain about the validity of the share he received. The disputation process aims to verify if the dealer is honest. We say that this process is explicit if it leads the dealer to send the share to the participant who complains in the presence of a third party. This latter has to identify who among the dealer and the participant is lying. Otherwise, the disputation process is supposed to be implicit (the dealer is considered as dishonest if the number of participants complaining about the validity of their shares is greater than a given parameter). Notice that only the three schemes of table 2 offer an explicit disputation process. The security assumptions of this process for these schemes are studied in table 3.

Verification of shares					
Category	**PVSS scheme**	**Problem**	**Evaluation**	**Proof**	**Evaluation**
Discrete Log	Feldman (1987)	DLP	Very hard	Non-interactive	Standard Model
	Stadler (1996)	DDLP	Non proved	Interactive	Zero-Knowledge
				Non-interactive	Random Oracle Model
	Schoenmakers (1999)	DDH	Hard	Interactive	Zero-Knowledge
				Non-interactive	Random Oracle Model
	Young & Yung (2001)	DLP	Very hard	Interactive	Zero-Knowledge
				Non-interactive	Random Oracle Model
	Behnad & Eghlidos (2008)	DLP	Very hard	Non-interactive	Standard Model
	Wu & Tseng (2011)	CDH	Hard	Non-interactive	Random Oracle Model
	Ben Shil, Blibech & Robbana (2011)	DLP	Very hard	Non-interactive	Standard Model
	Our PVSS (2012)	DLP	Very hard	Non-interactive	Standard Model
Factoring	Okamoto & Fujisaki (1998)	Factoring	Very hard	Interactive	Zero-Knowledge
		RSA	Hard	Interactive	Zero-Knowledge
Paillier cryptosystem	Ruiz & Villar (2005)	DCRA	Hard	Non-interactive	Random Oracle Model
Pairings	Heidarvand & Villar (2009)	DBSDH	Hard	Non-interactive	Standard Model
	Jhanwar (2010)	MSEDH	Hard	Non-interactive	Standard Model

Table 2: Evaluation of the verification process

Disputation				
Category	**PVSS scheme**	**Problem**	**Evaluation**	**Proof**
Discrete Log	Behnad & Eghlidos (2008)	CDH	Hard	Interactive
	Ben Shil, Blibech & Robbana (2011)	CDH	Hard	Interactive
	Our PVSS (2012)	CDH	Hard	Interactive

Table 3: Evaluation of the disputation process

Membership proof					
Category	**PVSS scheme**	**Problem**	**Evaluation**	**Proof**	**Evaluation**
Discrete Log	Schoenmakers (1999)	DDH	Hard	Interactive	Zero-Knowledge
				Non-interactive	Random Oracle Model
	Behnad & Eghlidos (2008)	CDH	Hard	Interactive	Zero-Knowledge
	Ben Shil, Blibech & Robbana (2011)	CDH	Hard	Interactive	Zero-Knowledge
	Our PVSS (2012)	CDH	Hard	Interactive	Zero-Knowledge
Pairings	Heidarvand &Villar (2009)	DBSDH	Hard	Non-interactive	Standard Model

Table 4: Evaluation of the membership proof

The membership proof can be implicit (a participant has to give his part, in the reconstruction process, to prove that he is an authorized participant) or explicit (a participant can prove to a verifier that he is an authorized participant possessing a valid share without revealing this share). When this process is explicit, it can be interactive or non-interactive. In table 3, we focus on PVSS schemes with explicit membership proof process and study the interactivity of each process and its security assumptions.

To summarize, we provide in this paper a new PVSS scheme having the following properties:

First, during the distribution process, our scheme uses a simple encryption function to encrypt the shares before distributing them. The encryption of the shares is secure under the CDH assumption.

When he receives a share of the secret, a participant can extract and verify the validity of his share without any communication with any party, even the dealer. We say that our verification process is non-interactive.

In case of any complaint against the dealer, the concerned participant, the dealer and a third party R can run a disputation process in order to establish who is cheating. The disputation process is secure under the CDH assumption.

Later, an explicit Zero-Knowledge membership process can be run to allow every participant to prove interactively his membership to a verifier who asked for that. This process is secure under the CDH assumption. Notice here that only three schemes offer an explicit membership proof and an explicit disputation process at the same time: the present scheme, the scheme of Behnad and Eghlidos [2] and

the scheme of Ben Shil, Blibech and Robbana [17].

Moreover, notice that in our scheme, when submitting an encrypted share to the party concerned by computing the secret, an implicit membership proof is given and it is not necessary to run the explicit membership only proof.

Finally, we point that the use of the XOR operator in our scheme makes it less timeconsuming than the schemes presented in [2] and [17].

6 Conclusion

The new PVSS scheme proposed in this paper is very simple while being secure. In fact, thanks to the use of a simple encryption function, we reduce computations in all the processes of the scheme. In addition, like in the scheme proposed in [2] we added two new processes: a disputation process and a membership proof process. Thanks to these processes, no one can cheat.

References

[1] C. Asmuth & J. Bloom (1983): *A modular approach to key safeguarding*. *IEEE Transactions on Information Theory* 29(2), pp. 208–211, doi:10.1109/TIT.1983.1056651.

[2] A. Behnad & T. Eghlidos (2008): *A new, publicly verifiable, secret sharing scheme*. *Sci. Iran.* 15(2), pp. 246–251.

[3] G. R. Blakley (1979): *Safeguarding cryptographic keys*. *Managing Requirements Knowledge, International Workshop on* 0, p. 313, doi:10.1109/AFIPS.1979.98.

[4] F. Boudot & J. Traoré (1999): *Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery*. In: *Proceedings of the Second International Conference, ICICS99*, 0302-9743, Springer-Verlag, Berlin, Heidelberg, pp. 87–102, doi:10.1007/978-3-540-47942-0_8.

[5] B. Chor, S. Goldwasser, S. Micali & B. Awerbuch (1985): *Verifiable secret sharing and achieving simultaneity in the presence of faults*. In: *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, SFCS '85, IEEE Computer Society, Washington, DC, USA, pp. 383–395, doi:10.1109/SFCS.1985.64.

[6] P. Feldman (1987): *A practical scheme for non-interactive verifiable secret sharing*. In: *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, SFCS '87, IEEE Computer Society, Washington, DC, USA, pp. 427–438, doi:10.1109/SFCS.1987.4.

[7] A. Fiat & A. Shamir (1986): *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*. In: *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, *Lecture Notes in Computer Science* 263, Springer, pp. 186–194, doi:10.1007/3-540-47721-7_12.

[8] E. Fujisaki & T. Okamoto (1998): *A practical and provably secure scheme for publicly verifiable secret sharing and its applications*. In: *Proceedings of the annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'98, Springer-Verlag, Berlin, Heidelberg, pp. 32–46, doi:10.1007/BFb0054115.

[9] S. Heidarvand & J. L. Villar (2009): *Selected Areas in Cryptography*. chapter Public Verifiability from Pairings in Secret Sharing Schemes, Springer-Verlag, Berlin, Heidelberg, pp. 294–308, doi:10.1007/978-3-642-04159-4_19.

[10] M. P. Jhanwar (2011): *A Practical (Non-interactive) Publicly Verifiable Secret Sharing Scheme*. In: *ISPEC'11*, pp. 273–287.

[11] E. D. Karnin, J. W. Greene & M. E. Hellman (1983): *On secret sharing systems*. *IEEE Transactions on Information Theory* 29(1), pp. 35–41, doi:10.1109/TIT.1983.1056621.

[12] P. Paillier (1999): *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*. In: *EUROCRYPT*, pp. 223–238, doi:10.1007/3-540-48910-X_16.

[13] T. Pedersen (1992): *Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing*. In: *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, Springer-Verlag, London, UK, UK, pp. 129–140, doi:10.1007/3-540-46766-1_9.

[14] A. Ruiz & J. L. Villar (2005): *Publicly Verfiable Secret Sharing from Paillier's Cryptosystem*. In: *WEWoRC*, pp. 98–108. Available at `http://subs.emis.de/LNI/Proceedings/Proceedings74/article3801.html`.

[15] B. Schoenmakers (1999): *A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic*. In: *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, Springer-Verlag, London, UK, UK, pp. 148–164, doi:10.1007/3-540-48405-1_10.

[16] A. Shamir (1979): *How to share a secret*. *Commun. ACM* 22(11), pp. 612–613, doi:10.1145/359168.359176.

[17] A. Ben Shil, K. Blibech & R. Robbana (2012): *Un nouveau schéma de partage de secrets publiquement vérifiable*. In: *Proceedings of the 7th Conference on Network and Information Systems Security (SAR-SSI)*.

[18] M. Stadler (1996): *Publicly verifiable secret sharing*. In: *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, Springer-Verlag, Berlin, Heidelberg, pp. 190–199, doi:10.1007/3-540-68339-9_17.

[19] T. Y. Wu & Y. M. Tseng (2011): *A pairing-based publicly verifiable secret sharing scheme*. *Journal of Systems Science and Complexity* 24(1), pp. 186–194, doi:10.1007/s11424-011-8408-6.

[20] A. Young & M. Yung (2001): *A PVSS as Hard as Discrete Log and Shareholder Separability*. In: *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '01, Springer-Verlag, London, UK, UK, pp. 287–299, doi:10.1007/3-540-44586-2_21.

[21] J. Yu, F. Kong & R. Hao (2007): *Publicly Verifiable Secret Sharing with Enrollment Ability*. In: *SNPD (3)*, pp. 194–199, doi:10.1109/SNPD.2007.435.