Public Verifiability from Pairings in Secret Sharing Schemes*

Somayeh Heidarvand and Jorge L. Villar

Universitat Politècnica de Catalunya, Spain
{somayeh,jvillar}@ma4.upc.edu

Abstract. In this paper we propose a new publicly verifiable secret sharing scheme using pairings with close relations to Shoenmakers' scheme. This scheme is efficient, multiplicatively homomorphic and with unconditional verifiability in the standard model. We formalize the notion of Indistinguishability of Secrets and prove that out scheme achieves it under the Decisional Bilinear Square (DBS) Assumption that is a natural variant of the Decisional Bilinear Diffie Hellman Assumption. Moreover, our scheme tolerates active and adaptive adversaries.

Keywords. Public Verification, Secret Sharing, Pairings, Semantic Security, Threshold Cryptography.

1 Introduction

Most of the work on secret sharing dates from the eighties and the nineties, before the invention of Paillier's cryptosystem [10] and the first positive use of pairings in cryptography [8]. As a consequence, little attention has been paid to the potential use of recently invented cryptographic tools in the design of secret sharing schemes. However, pairings have been successfully used in the design of some distributed cryptographic protocols like threshold encryption and threshold signatures, in the last years.

Background. In a secret sharing (SS) scheme, a dealer D wants to share a secret among a set of participants in such a way that only special (qualified) subsets are able to recover the secret. Here we are interested in (t, n)-threshold secret sharing schemes, in which the qualified subsets are those with at least t participants. Since the publication of the seminal paper by Shamir [15], secret sharing has found innumerable applications and is nowadays considered as a fundamental tool for the design of distributed cryptographic protocols.

The first constructions of secret sharing schemes achieved a high level of security (secrecy): The Shamir scheme provides secrecy even in the presence of a passive adversary (i.e., an eavesdropper who controls the secret information of at most $t-1$ participants) with unlimited computational power. However, these schemes do not provide enough protection against dishonest participants or a dishonest dealer.

* This research was partially supported by the Centre de Recerca Matemàtica (CRM).

R. Avanzi, L. Keliher, and F. Sica (Eds.): SAC 2008, LNCS 5381, pp. 294–308, 2009.

Verifiable secret sharing (VSS) schemes have been introduced in [2] to solve the problem of dishonest dealers and dishonest participants who try to deceive other participants. Feldman's VSS scheme [3] is a very practical scheme in which secrecy is based on a computational assumption related to the Discrete Logarithm problem. However, since a deterministic function of the secret is published, only a weak notion of secrecy (one-wayness) is guaranteed. Pedersen [11], proposed a VSS scheme in which secrecy is guaranteed for an unbounded passive adversary, but verifiability relies on a computational assumption.

In [16], Stadler proposed a publicly verifiable secret sharing (PVSS) scheme. In this scheme, the validity of the shares can be verified by anyone only from the public information. Typically, in a PVSS scheme, the dealer only broadcasts some information to the participants, and no private channels are needed for the distribution of the shares. Shoenmakers' PVSS scheme [14] works in a group in which the Discrete Logarithm problem is intractable. His scheme is quite simple, but to make it publicly verifiable some non-interactive zero-knowledge proofs have been used.

In most publicly verifiable secret sharing schemes [5,14], the verification procedure involves interactive proofs of knowledge. These proofs are made non-interactive by means of the Fiat-Shamir technique [4]. This implies that verifiability relies on the properties of some hash function. Actually, known security proofs for verifiability work only in the Random Oracle Model (ROM), and there is a known negative result about the universal validity of Fiat-Shamir heuristics [6].

There are other known ways to obtain non-interactive zero knowledge proofs without using Fiat-Shamir. For instance, a recent work by Groth *et al.* [7] shows a generic non-interactive zero knowledge proof for any language in NP in the common reference string model (CRS), that takes advantage of pairings. However, these zero knowledge proofs are still quite inefficient.

Based on a PVSS scheme by Fujisaki and Okamoto [5], Ruiz and Villar in [12] overcame some of the above problems: a new PVSS scheme is proposed which makes use of the additive homomorphic property of Paillier's encryption. The dealer commits to the coefficients of the polynomial of the underlying Shamir SS scheme by broadcasting their encryptions. The resulting PVSS scheme is unconditionally verifiable (in the Standard Model) and the verification protocol is intrinsically non-interactive (i.e., does not make use of Fiat-Shamir heuristics). The main drawback of this scheme is that it requires an additional step of interaction in the sharing phase: the dealer holds a secret/public key pair and every participant sends an encrypted random value to him in order to establish a secure channel through which the corresponding share is sent.

Moreover, compared to Feldman's scheme, the Ruiz-Villar scheme provides a higher level of secrecy called indistinguishability (IND) based on the Decisional Composite Residuosity (DCR) assumption. Due to the unconditional verifiability, this secrecy is guaranteed even in the presence of an active and adaptive adversary.

Another consequence of not using Fiat-Shamir non-interactive zero-knowledge proofs is that the Ruiz-Villar scheme is an additively homomorphic PVSS scheme (i.e., anyone can compute the public information of a sharing of $s_1 + s_2$ from the individual sharing of s_1 and s_2, including the verification information, in such a way that nobody can distinguish this sharing of $s_1 + s_2$ from a direct sharing of the same value).

Boldyreva [1] proposed a threshold signature scheme based on gap Diffie-Hellman groups (that can be naturally instantiated with pairings). The signing key is distributed by using Feldman's VSS and the signature verification procedure takes advantage of the DDH oracle. The techniques used in Boldyreva's paper are somewhat similar to ours but in the different context of threshold signatures.

Contributions. On the one hand, we give two formal definitions of secrecy in publicly verifiable secret sharing, which capture the notion of indistinguishability of shared secrets. We also discuss their relationship.

On the other hand, we propose a new PVSS scheme that overcomes the use of Fiat-Shamir zero-knowledge proofs in a different way than in [12], and which does not require any additional interaction in the sharing phase: we basically replace zero-knowledge proofs in Shoenmakers' scheme by equalities involving bilinear map computations. The resulting scheme has the following features:

- Public Verifiability: a misbehaving dealer or participant is unconditionally detected.
- Secrecy: indistinguishability of secrets is based on the Decisional Bilinear Square Assumption, which is a variant of the Decisional Bilinear Diffie-Hellman Assumption.
- Active adversaries are tolerated (even adaptive ones), whenever there is a majority of honest participants.
- Multiplicatively homomorphic property (compatible with public verifiability).
- Efficiency comparable to Shoenmakers' scheme, with a more efficient dealer but a less efficient verifier.

Reducing the computational cost of the dealer can be desirable in applications in which there exist many potential dealers but a limited amount of participants, as in electronic voting. A variant of the basic scheme is also presented, which allows secret reconstruction via open channels in an efficient way. We also show how the scheme generalizes to linear access structures other than threshold ones.

Organization. The paper is organized as follows: In Section 2 we recall the characteristics of PVSS schemes. Computational secrecy for PVSS schemes is revisited in Section 3. The proposed PVSS scheme is presented in Section 4 and its security is analyzed in Section 5. In Section 6 a variant of the scheme which allows the reconstruction of the secret on public channels is presented. Finally, the multiplicatively homomorphic property of the scheme is discussed in Section 7.

Notation. As usual in cryptography papers, we use the convenient notation $x \xleftarrow{\$} X$ to denote that x is a uniformly distributed random element of a set X.

2 Publicly Verifiable Secret Sharing Schemes

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n participants. We only refer to a (t, n)-threshold access structure, although the schemes proposed in this paper can easily be generalized to any vector space access structure. The dealer D wants to share a secret s between the participants of $\mathcal{P}$ in such a way that every set of at least t participants can recover the secret, and no set of at most $t-1$ participants can get any information about the secret. V is any (external) verifier who wants to check any phase of the scheme.

In a basic secret sharing scheme, three subprotocols are needed: *setup*, *distribution* of the shares and *reconstruction* of the secret. In a PVSS scheme, the dealer is supposed to communicate with participants via open channels. In spite of simplicity, we assume the existence of an authenticated broadcast channel. Furthermore, we can assume the existence of private channels between participants during the secret reconstruction. However, some basic PVSS schemes can be modified in order to remove this last assumption. An additional public *verification* subprotocol is also considered.

Setup. All the parameters of the scheme are generated and published by the dealer D. Also every participant publishes his public key and withholds the corresponding secret key.[1]

Distribution. For a secret s the dealer creates $s_1, s_2, \ldots, s_n$ as the shares of $P_1, P_2, \ldots, P_n$ respectively. The dealer computes and publishes the encrypted shares $E_i(s_i)$ for each participant $P_i \in \mathcal{P}$. He also must publish $PROOF_D$ to ensure the verifier that the published values are encryptions of correct shares.

Verification. From all the public information generated during setup and distribution phases, V verifies non-interactively that the published information is consistent and that every authorized subset of (honest) participants will recover the same secret. If verification fails, the whole protocol is aborted (i.e., honest participants exit the protocol).

Reconstruction. Using his secret key, every participant P_i in a qualified subset $A \subset \mathcal{P}$ decrypts his encrypted share and gets s_i. Then, all participants in A exchange their shares s_i together with a proof $PROOF_{P_i}$ via private channels. Every participant in A locally reconstructs the secret from a subset of t correct

[1] This public key could be the encryption with the public key of the dealer of a random value chosen by the participant as a one-time key. However, different instances of the protocol need independent one-time keys, and this adds a new interaction step in the distribution subprotocol, as in [12].

shares (i.e., those with a valid proof). If no private channels are available to participants, they then can send encryptions of the shares instead of the shares themselves.

As usual, adversaries can be classified into *passive* and *active*, depending on the behavior of corrupted participants: a passive adversary cannot change the behavior of a corrupted participant, while an active adversary can change it in any possible way, but in any case the adversary learns all the participant's secret information. Also adversaries can be *static* or *adaptive*. A static adversary decides the participants whom will be corrupted at the very beginning of the protocol, while an adaptive one can decide to corrupt a new participant at any time, as a function of his knowledge. We always consider a *rushing* adversary, who makes corrupted participants wait for honest participants' messages before sending theirs in each communication round.

The three properties required for a PVSS scheme: *correctness*, *verifiability* and *secrecy* are defined below.

Correctness. If the dealer and the participants act honestly, every qualified subset of participants reconstructs the secret s in the reconstruction phase. This obviously implies that the dealer passes the verification subprotocol.

Verifiability. If a dishonest dealer passes the verification subprotocol, then there exists a unique value s such that the honest participants in any qualified subset with at least t honest participants recover s as the secret. We can consider weaker notions of verifiability by tolerating a negligible error probability (statistical verifiability) or by considering a computationally bounded adversary (computational verifiability).

Secrecy. For an honest dealer, the adversary cannot learn any information about the secret, even after the execution of the reconstruction subprotocol by all honest participants. We can also consider weaker notions of secrecy, depending on the type of adversary and the tolerated amount of information he can learn about the secret.

Unconditional secrecy is not possible in PVSS schemes, since the encrypted shares are sent by public channels, so an unbounded adversary can decrypt them and then compute the secret. In the following section we review some notions of computational secrecy (i.e., secrecy against a computationally bounded adversary).

3 Computational Secrecy

PVSS schemes can be related to threshold decryption schemes, in which only qualified subsets can decrypt ciphertext encrypted with a certain public key. In this analogy, the shared secret is the encrypted message and the information published by the dealer is the ciphertext.

Hence, one-way secrecy in PVSS means that the adversary wants to know the whole secret. However, achieving only this secrecy level (as in Feldman's scheme) does not appear to be enough in the real world.

A formalization of the intuitive notion of semantic security for a PVSS scheme was first introduced in [12]. We refine that secrecy notion for the worst case active and adaptive adversary and give two secrecy levels that we call IND1 and IND2. The weaker notion (IND1) informally means that the adversary cannot tell apart the shared secret from a random value. This is a natural definition if the PVSS scheme is seen as a Key Encapsulation Mechanism (KEM).

Definition 1 (Indistinguishability of secrets (IND1)). *We say that a (t, n)-threshold PVSS scheme is IND1-secret if any probabilistic polynomial time $\mathcal{A}$ has a negligible advantage in the following game played against a challenger $\mathcal{C}$. During the game, $\mathcal{A}$ can corrupt a new participant at any time, but up to $t-1$ participants in total. When $\mathcal{A}$ corrupts a participant, he receives his secret key (only after step 1 in the game). A list of corrupted participants is maintained during the game.*

1. *$\mathcal{C}$ runs the setup subprotocol and sends the public parameters to $\mathcal{A}$ along with the public keys of still uncorrupted participants. $\mathcal{C}$ stores the secret keys of those participants.*
2. *$\mathcal{A}$ sends the public keys of already corrupted participants.*
3. *$\mathcal{C}$ picks two random secrets x_0, x_1 and a random bit $b \in \{0, 1\}$. Then he runs the distribution subprotocol for secret x_0 and sends all the resulting information to $\mathcal{A}$, along with x_b.*
4. *$\mathcal{C}$ runs the reconstruction subprotocol for the set of all uncorrupted participants and sends all the messages exchanged via public channels (if any) to $\mathcal{A}$. No new corruptions are allowed from this point.*
5. *$\mathcal{A}$ outputs a guess bit b'.*

The advantage of $\mathcal{A}$ in that game is defined as $\left|\mathsf{Prob}[b' = b] - \frac{1}{2}\right|$.

The stronger notion (IND2) is similar to (IND1) but now x_0, x_1 are chosen by the adversary.

Definition 2 (Indistinguishability of secrets (IND2)). *We define IND2-secrecy of a (t, n)-threshold PVSS scheme exactly as in the definition of IND1-secrecy but replacing item 3 by*

3' $\mathcal{A}$ selects to secrets x_0, x_1 and sends them to $\mathcal{C}$. Then $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and runs the distribution subprotocol for the secret x_b. Finally, $\mathcal{C}$ sends all the resulting information to $\mathcal{A}$.

Clearly, IND2-secrecy implies IND1-secrecy but the converse is not true. However, one can upgrade an IND1-secret PVSS scheme to achieve IND2-secrecy by using the original PVSS scheme to share a random session key K, and then the dealer publishes $K \oplus s$, where s is the secret and $\oplus$ is a suitable group operation. We refer to this PVSS scheme as a *hybrid* PVSS scheme. See Appendix B for a detailed proof of this fact.

4 The Proposed PVSS Scheme

Assume that G is a group of order q, q a prime number, and g and h are two independent generators of this group. Let e be a non-degenerated bilinear map $e : G \times G \to G_1$. This means that the map $e : G \times G \to G_1$ fulfils the following properties:

1. $e(g^\alpha, g^\beta) = e(g, g)^{\alpha\beta}$ for all $\alpha, \beta \in \mathbb{F}_q$.
2. $e(g, g) \neq 1$.
3. $e(x, y)$ is efficiently computable given x and y in G.

We wish to use Shoenmakers' protocol together with the bilinear map to build a (t, n)-threshold PVSS scheme to share a secret in G_1 among the participants $P_1, \ldots, P_n$, where $n \geq 2t-1$, in such a way that the public verifiability does not require the use of Fiat-Shamir non-interactive zero-knowledge proofs. To do this, the dealer chooses $z_0 \in \mathbb{F}_q^*$ randomly and distributes the secret $S = e(h, h)^{z_0}$ in the following way:

Setup. Every participant P_i chooses a random secret value $d_i \in \mathbb{F}_q^*$ and publishes $h_i = h^{d_i}$ as his public key.

Distribution. The dealer chooses a random polynomial $P(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ of degree at most $t-1$ with coefficients in $\mathbb{F}_q$ and $\alpha_0 = z_0$. The dealer publishes the commitments $C_j = g^{\alpha_j}$, for $0 \leq j < t$. He also publishes the encryptions of the shares $Y_i = {h_i}^{P(i)}$ for $1 \leq i \leq n$.

Verification. Every (external) verifier can compute the value $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$ for every participant P_i by himself and check the correctness of the shares by checking the equation $e(X_i, h_i) = e(g, Y_i)$. If the verification fails, all participants exit the protocol (i.e., they refuse to take part in the reconstruction subprotocol).

Reconstruction. Let A be a qualified subset of participants. Each participant in A gets the encrypted share $S_i = h^{P(i)}$ by using its private key and computing $S_i = {Y_i}^{1/d_i}$. Then all participants in A pool their shares. All shares can be verified easily by other participants of A by checking the equation $e(S_i, h_i) = e(Y_i, h)$. After the verification, if there are at least t correct shares, then for an arbitrary set $B \subseteq A$ of t participants which have pooled correct shares, every participant in A can get h^{z_0} by Lagrange interpolation: $\prod_{P_i \in B} {S_i}^{\lambda_i} = \prod_{P_i \in B} (h^{P(i)})^{\lambda_i} = h^{\sum_{P_i \in B} \lambda_i P(i)} = h^{P(0)} = h^{z_0}$, where $\lambda_i = \prod_{P_j \in B \setminus \{P_i\}} \frac{j}{j-i}$ is a Lagrange coefficient. The secret S will be recovered by computing $e(h^{z_0}, h)$. The protocol is summarized in Figure 1.

5 Analysis of the Scheme

5.1 Correctness

Correctness of the scheme means that: (i) an honest D always passes the verification procedure, and (ii) any subset of at least $2t-1$ participants (which

Setup:

▷ Each participant P_i computes $(pk, sk) = (h_i, d_i) = (h^{d_i}, d_i)$, where $d_i \xleftarrow{\$} \mathbb{F}_q^*$, and broadcasts pk.

Distribution:

Let $z_0 \xleftarrow{\$} \mathbb{F}_q^*$ and $S = e(h,h)^{z_0}$ be the secret to be shared.

▷ D picks a random polynomial $P(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ where $\alpha_0 = z_0$ and $\alpha_j \xleftarrow{\$} \mathbb{F}_q$.

▷ D publishes $C_j = g^{\alpha_j}$ and the encrypted shares $Y_i = {h_i}^{P(i)}$.

Verification:

▷ V computes $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$ and checks if $e(X_i, h_i) = e(g, Y_i)$.

Reconstruction:

Let A be a subset of participants running the reconstruction subprotocol.

▷ Every $P_i \in A$, sends $S_i = {Y_i}^{1/d_i}$ to the other participants in A.

▷ Every $P_i \in A$, checks if $e(S_j, h_j) = e(Y_j, h)$ for all $P_j \in A \setminus \{P_i\}$ and defines B_i as a subset of t participants that pass the test.

▷ Every $P_i \in A$, computes $\prod_{P_j \in B_i} {S_j}^{\lambda_j} = h^{z_0}$ where $\lambda_j = \prod_{P_k \in B_i \setminus \{P_j\}} \frac{k}{k-j}$ and gets $S = e(h^{z_0}, h)$.

Fig. 1. PVSS scheme with reconstruction via private channels

ensures us that there are at least t honest participants) is always able to recover the secret shared by an honest D. Checking these requirements for the above protocol is straightforward.

5.2 Verifiability

Now we show that if D passes the verification, then all participants in the protocol must behave honestly or will be detected. More precisely, on the one hand, the dealer must be honest in the distribution subprotocol and, on the other hand, participants must be honest in the reconstruction subprotocol.

Verifiability of the Distribution. In the following, we prove that a dishonest D cannot cheat the participants without being detected in the verification. More precisely, if D passes the verification, then all qualified subsets of honest participants will reconstruct the same secret.

Lemma 1. *If V accepts, then there exists a unique polynomial $P(x)$ such that the encrypted share of participant P_i is $Y_i = {h_i}^{P(i)}$ for $1 \leq i \leq n$.*

Proof. Assume that the share of participant P_i is equal to $Y_i = {h_i}^s$. Let us write $C_j = g^{\alpha_j}$ for suitable α_j and consider the polynomial $P(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{t-1} x^{t-1}$. If V accepts, then for every $1 \leq i \leq n$ the dealer passes the equation $e(X_i, h_i) = e(g, Y_i)$, where $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$. By the definition of $P(x)$, we have $e(g, h_i)^{P(i)} = e(g, h_i)^s$, which leads to $s = P(i)$. The uniqueness of $P(x)$ implies that all sets of t correct shares get the same secret in the reconstruction subprotocol. □

In the actual protocol, all participants act as verifiers after the secret distribution stage. Then, if D broadcasts corrupt information, then all honest participants drop out of the protocol. Hence, nobody can successfully run the reconstruction subprotocol. It is worth noticing that verifiability of D is unconditional (i.e., does not depend on any computational assumption).

Verifiability of the Shares in the Reconstruction Subprotocol. Consider now that D behaves honestly. Let P_i and P_j be two participants taking part in the reconstruction subprotocol. Suppose P_i opens his secret value $S_i = {Y_i}^s$ to P_j, and P_j behaves honestly.

Lemma 2. *If P_j accepts P_i's value, then $S_i = {Y_i}^{1/d_i}$, where d_i is the secret key of P_i, that is $h_i = h^{d_i}$.*

Proof. Since P_j accepts P_i's value, then $e(S_i, h_i) = e(Y_i, h)$, and so $e({Y_i}^s, h^{d_i}) = e(Y_i, h)$. By using the properties of the bilinear map we get $e(Y_i, h)^{sd_i} = e(Y_i, h)$, which results in $sd_i = 1$. So if P_j accepts the secret share of P_i, then $S_i = {Y_i}^{1/d_i}$. □

Thus, by the two previous lemmas, all honest participants involved in the reconstruction subprotocol accept only correct shares $S_i = {Y_i}^{1/d_i} = h^{P(i)}$ (whether the shares come from honest or dishonest participants). If there are at least t honest participants in the subset A running the reconstruction subprotocol, then every honest participant in A accepts at least t correct shares, which lead to the secret $S = h^{P(0)}$ by Lagrange interpolation in the exponent. Notice that this property does not depend on any computational assumption.

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

Theorem 1. *The proposed PVSS scheme is publicly verifiable even in the presence of an unbounded adversary.*

5.3 Secrecy

Our goal now is to show that an active and adaptive probabilistic polynomial time adversary corrupting at most $t - 1$ participants cannot obtain any information about the shared secret S, assuming an honest D. To show this, we first define the following assumption:

Assumption 1 (Decisional Bilinear Square (DBS)). *Let G and G_1 be two groups of prime order q, g be a random generator of G and $e : G \times G \to G_1$ be a non-degenerated bilinear map. For random values μ, ν and s chosen uniformly and independently from $\mathbb{F}_q^*$ and given $h = g^\mu$, $u = g^\nu$, the following probability distributions are polynomially indistinguishable: $D_0 = (g, h, u, T_0 = e(h, h)^\nu)$ and $D_1 = (g, h, u, T_1 = e(h, h)^s)$.*

This assumption is equivalent to the Decisional Bilinear Quotient (DBQ) Assumption, recently introduced in [9], and it is a natural variant of the standard Decisional Bilinear Diffie-Hellman Assumption, in which informally, an

adversary aims to tell apart $e(g,g)^{xyz}$ from a random value, given (g, g^x, g^y, g^z). DBS Assumption corresponds to the case $x = y$. See Appendix C for more details about the relations of these assumptions.

Theorem 2. *If the DBS Assumption holds, then the proposed scheme is IND1- secret.*

Proof. Assume that there is an active and adaptive probabilistic polynomial time adversary, $\mathcal{A}$, playing the game in Definition 1 with a non-negligible advantage $\varepsilon_{\mathcal{A}}$. Then we describe a simulator $\mathcal{F}$ that using $\mathcal{A}$ as a subroutine can break the DBS Assumption with a non-negligible advantage $\varepsilon_{\mathcal{F}}$.

1. Once $\mathcal{F}$ receives the description of a random instance of the DBS Problem $(q, G, G_1, e, g, h = g^{\mu}, u = g^{\nu}, T_b)$, as described in Assumption 1, he simulates the (t,n)-threshold PVSS scheme as a challenger for $\mathcal{A}$. So $\mathcal{F}$ sends $(n, t, \mathcal{P}, q, G, G_1, e, g, h)$ to $\mathcal{A}$ as the public parameters of the scheme. $\mathcal{A}$ chooses a subset $B_0 \subset \mathcal{P}$ of initially corrupted players and gives it to $\mathcal{F}$. Now $\mathcal{F}$ guesses the set of all players corrupted by $\mathcal{A}$ at the end of the game by choosing at random B such that $B_0 \subset B \subset \mathcal{P}$ and $|B| = t-1$. Then $\mathcal{F}$ computes the public keys of the players as follows: $\forall P_i \in B \setminus B_0$; $h_i = h^{d_i}$, $d_i \xleftarrow{\$} \mathbb{F}_q^*$ and $\forall P_i \in \mathcal{P} \setminus B$; $h_i = g^{r_i}$, $r_i \xleftarrow{\$} \mathbb{F}_q^*$, and sends them to the adversary.
2. $\mathcal{A}$ sends the public keys of the corrupted players which have been arbitrary chosen by himself.
3. $\mathcal{F}$ chooses $s_i \xleftarrow{\$} \mathbb{F}_q$ and sets $Y_i = {h_i}^{s_i}$ for all players $P_i \in B$. There exists a unique interpolating polynomial $P(x) = \alpha_0 + \alpha_1 x + ... + \alpha_{t-1} x^{t-1} \in \mathbb{F}_q[x]$, such that $\forall P_i \in B$; $P(i) = s_i$ and $g^{P(0)} = u$. Thus all the coefficients can be uniquely determined for some efficiently computable constants μ_{ij} (that only depend on B) as $\alpha_j = \sum_{P_i \in B} \mu_{ij} s_j + \mu_{0j}\nu$. Now $\mathcal{F}$ computes $C_j = g^{\sum_{P_i \in B} \mu_{ij} s_i} u^{\mu_{0j}}$, $1 \le j \le t-1$ and sets $C_0 = u$. Then $\forall P_i \in \mathcal{P} \setminus B$, $\mathcal{F}$ sets $Y_i = {h_i}^{P(i)} = g^{P(i) r_i} = [u \prod_{j=1}^{t-1} C_j^{i^j}]^{r_i}$, and sends all Y_i, all C_j and T_b to $\mathcal{A}$.
4. $\mathcal{A}$ chooses $B_1 \in \mathcal{P} \setminus B_0$ such that $|B_0 \cup B_1| \le t-1$, and corrupts the participants in B_1. If $B_1 \nsubseteq B \setminus B_0$, then $\mathcal{F}$ exits the game giving a random bit b' as output. Otherwise $\mathcal{F}$ sends the secret key d_i of every participant $P_i \in B_1$ to $\mathcal{A}$.
5. Eventually $\mathcal{A}$ ends by outputting a bit b' which is forwarded by $\mathcal{F}$.

Let Fail denote the event that $\mathcal{F}$ exits the game at step 4. Notice that, if Fail occurs then the probability of success of $\mathcal{F}$ (i.e., $b' = b$) is exactly $1/2$. Otherwise, $\mathcal{F}$ perfectly simulates the challenger for $\mathcal{A}$. On the other hand, the choice of B is independent of all the variables of the secret sharing scheme, and then Fail is independent of the success of $\mathcal{A}$. Thus, the probability of success of $\mathcal{F}$ is $\mathsf{Succ}_{\mathcal{F}} = \frac{1}{2}\mathsf{Prob}[\mathsf{Fail}] + \mathsf{Succ}_{\mathcal{A}}\,\mathsf{Prob}[\neg\mathsf{Fail}]$ and $\varepsilon_{\mathcal{F}} = \varepsilon_{\mathcal{A}}\,\mathsf{Prob}[\neg\mathsf{Fail}]$. The (conditional) probability of $\neg\mathsf{Fail}$ can be easily computed as $\mathsf{Prob}[\neg\mathsf{Fail}] = \binom{t-1-|B_0|}{|B_1|} / \binom{n-|B_0|}{|B_1|}$, for any possible choice of B_1, which ranges from 1 if $B_1 = \emptyset$ (that is, in the case of an active but static adversary $\varepsilon_{\mathcal{F}} = \varepsilon_{\mathcal{A}}$) to $\binom{n}{t-1}^{-1}$ if $B_0 = \emptyset$ and $|B_1| = t-1$ (that is, the worst case adversary). ◻

As seen in Section 3, we can modify the basic PVSS scheme to achieve IND2-secrecy by letting the dealer share a random value $K = e(h,h)^{z_0} \in G_1$, and then publish the product $T = KS$, where $S \in G_1$ is the actual secret he wants to share.

6 Secret Reconstruction on Public Channels

In the basic scheme the secret reconstruction supposes the existence of private channels between participants. In this section we remove this requirement without losing any good property of the scheme.

Assume that participant P_i wants to sends his encrypted share, $S_i = {Y_i}^{1/d_i}$, to P_j. To do that publicly, he chooses a random value ρ and sends $(r, z, w) = ({h_i}^{\rho}, {Y_i}^{\rho}, {h_j}^{1/(d_i \rho)})$, where h_i, h_j are the public keys of P_i, P_j respectively. Now everybody can verify the correctness by checking the equations $e(r, Y_i) = e(z, h_i)$ and $e(r, w) = e(h_j, h)$, since Y_i is publicly available from the sharing information broadcast by the dealer. Notice that this verification is unconditional.

Then P_j computes the share of P_i as $e(h,h)^{P(i)} = e(z,w)^{1/d_j}$. From t correct shares, P_i can locally compute the secret $S = e(h,h)^{P(0)}$ as usual, by means of Lagrange interpolation in the exponent. The secrecy of the modified protocol is also based on the DBS Assumption.

Theorem 3. *The protocol is IND1-secret under the DBS Assumption.*

Proof. We only have to modify step 5 of the simulation in the proof of Theorem 2 to provide $\mathcal{A}$ with all the messages exchanged by the uncorrupted participants during the secret reconstruction.

5' For every ordered pair (P_i, P_j) of different uncorrupted participants, $\mathcal{F}$ chooses $\rho \xleftarrow{\$} \mathbb{F}_q^*$ and sends $(r = {h_i}^{\rho}, z = {Y_i}^{\rho}, w = h^{r_j/(r_i \rho)})$ to $\mathcal{A}$. Eventually $\mathcal{A}$ ends by outputting a bit b' which is forwarded by $\mathcal{F}$.

Notice that the simulation of the secret reconstruction subprotocol is perfect. Therefore, the advantage of $\mathcal{F}$ fulfils exactly the same equation as in Theorem 2. □

7 Homomorphic Properties

It is well known that some basic secret sharing schemes have nice homomorphic properties. For instance, in Shamir's scheme, if every participant P_i locally computes a linear combination of his shares s_i and t_i for the secrets s and t, respectively, then he obtains a new share corresponding to the same linear combination of the secrets. This interesting property has found a lot of applications in electronic voting or multiparty computation, for example.

However, if the same idea is applied to a publicly verifiable secret sharing scheme, then new difficulties arise: one wants to compute the sharing information (including verification information) of the operation of two secrets from the information of the individual sharing processes. This seems very hard to achieve

if the public verifiability depends on non-interactive zero-knowledge proofs, but it is straightforward in our scheme (as it was in [12]). Our basic scheme has the following multiplicatively homomorphic property. We assume that public keys of the participants are reused for multiple secret sharing.

Proposition 1. *Let* $(C_0, \ldots, C_{t-1}, Y_1, \ldots, Y_n)$ *and* $(\tilde{C}_0, \ldots, \tilde{C}_{t-1}, \tilde{Y}_1, \ldots, \tilde{Y}_n)$ *be the sharing information broadcast by the dealer for secrets* S *and* $\tilde{S}$*, respectively. Then, for any* $\alpha, \beta \in \mathbb{F}_q^*$ *the tuple* $(C_0^\alpha \tilde{C}_0^\beta, \ldots, C_{t-1}^\alpha \tilde{C}_{t-1}^\beta, Y_1^\alpha \tilde{Y}_1^\beta, \ldots, Y_n^\alpha \tilde{Y}_n^\beta)$ *has the same probability distribution as a direct sharing of the secret* $S^\alpha \tilde{S}^\beta$*.*

The same property applies to the IND2-secret improved scheme. Indeed, it suffices to do the same operation $T^\alpha \tilde{T}^\beta$ with the additional public elements T and $\tilde{T}$.

8 Final Remarks

As in Shoenmakers' scheme, the PVSS scheme proposed in this paper can be easily extended to linear access structures other than the (t, n)-threshold ones by following a standard procedure. Firstly, assign to every participant P_i a vector $\mathbf{v}_i = (v_{i,0}, \ldots, v_{i,t-1}) \in \mathbb{F}_q^t$ for a suitable dimension t, and let $\mathbf{v}_0 = (1, 0, \ldots, 0)$ be the vector associated to the dealer. Then replace the sharing polynomial $P(x)$ by a (dual) vector $\alpha = (\alpha_0, \ldots, \alpha_{t-1})$, and $P(i)$ by the dot product $\mathbf{v}_i \cdot \alpha$. Hence, X_i is computed as $X_i = \prod_{j=0}^{t-1} C_j^{v_{i,j}}$. All the remaining equations are maintained except for Lagrange interpolation coefficients, which are replaced by the coefficients of the expression of $\mathbf{v}_0$ as a linear combination of the vectors associated to a qualified subset of participants.

On the other hand, the proposed PVSS scheme has a performance comparable to Shoenmakers' scheme. Indeed, the dealer's computational effort of computing the non-interactive zero-knowledge proofs ($2n$ exponentiations) and the verification of them by the verifier ($2n$ multi exponentiations) have been replaced by the computation of $2n$ pairings by the verifier. Hence, the dealer's computation complexity is reduced in about a 50%. If we tolerate a positive error probability in the verification procedure, then the verifier can check a random combination of the n equations, reducing the number of pairing computations to only $n + 1$. Moreover, every participant taking part of our scheme's reconstruction subprotocol must compute some extra pairings (typically $2t-1$), but he does not have to compute and check the non-interactive zero-knowledge proofs (saving 2 exponentiations and $2t - 2$ multi exponentiations).

References

1. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
2. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Proc. 26th IEEE Symp. on Found. of Comp. Sci., pp. 383–395 (1985)

3. Feldman, P.: A Practical Scheme for Non-interactive Verifiable Secret Sharing. In: Proceedings 28th IEEE Symp. on Found. of Comp. Sci., pp. 427–437 (1987)
4. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
5. Fujisaki, E., Okamoto, T.: A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 32–46. Springer, Heidelberg (1998)
6. Goldwasser, S., Tauman, Y.: On the (In)security of the Fiat-Shamir Paradigm. In: Proc. 44th Annual IEEE Symp. on Found. of Comp. Sci., pp. 102–115 (2003)
7. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006)
8. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–394. Springer, Heidelberg (2000)
9. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 360–379. Springer, Heidelberg (2008)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
11. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
12. Ruiz, A., Villar, J.L.: Publicly Verifiable Secret Sharing from Paillier's Cryptosystem. In: WEWoRC 2005. LNI P-74, pp. 98–108 (2005)
13. Sadeghi, A.-R., Steiner, M.: Assumptions related to discrete logarithms: Why subtleties make a real difference. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 243–260. Springer, Heidelberg (2001)
14. Schoenmakers, B.: A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 148–164. Springer, Heidelberg (1999)
15. Shamir, A.: How to share a secret. Commun. of the ACM 22, 612–613 (1979)
16. Stadler, M.A.: Publicly verifiable secret sharing. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 190–199. Springer, Heidelberg (1996)

A A Short Description of the Ruiz-Villar PVSS Scheme

Ruiz-Villar PVSS uses the additively homomorphic Paillier cryptosystem to add public verifiability to Shamir's secret sharing scheme over the ring $\mathbb{Z}_N$, where $N = pq$ is an RSA modulus. Let g be an element with multiplicative order N in $\mathbb{Z}^*_{N^2}$ (e.g., $g = 1 + N$) and suppose that only the dealer knows p and q. The distribution subprotocol for a secret $s \in \mathbb{Z}_N$ works as follows:

1. P_i picks $(m_i, r_i) \xleftarrow{\$} \mathbb{Z}_N \times \mathbb{Z}^*_N$ and broadcasts $c_i = g^{m_i} {r_i}^N \bmod N^2$.
2. D picks a random polynomial $P(x) = \sum_{j=0}^{t-1} \alpha_j x^j$ where $\alpha_0 = s$ and $\alpha_j \xleftarrow{\$} \mathbb{Z}_N$. Then D sets $s_i = P(i) \bmod N$.
3. D decrypts all ciphertexts c_i, thus obtaining the pairs (m_i, r_i), and broadcasts $d_i = s_i + m_i \bmod N$.

4. D picks $R_j \stackrel{\$}{\leftarrow} \mathbb{Z}_N^*$ and broadcasts $A_j = g^{\alpha_j} R_j^N \bmod N^2$, for $0 \leq j < t$.
5. D also broadcasts $t_i = R_0 R_1^i \cdots R_{t-1}^{i^{t-1}} r_i \bmod N$, for every $i = 1, \ldots, n$.

For each $1 \leq i \leq n$, a verifier can check $A_0 A_1^i \cdots A_{t-1}^{i^{t-1}} = \frac{g^{d_i}}{c_i} t_i^N \bmod N^2$. Finally, the secret reconstruction subprotocol (on private channels) for a subset A with at least t honest participants, works as follows:

1. Every $P_i \in A$ sends the secret pair (m_i, r_i) to the other participants in A, who check that c_i is the corresponding Pallier's ciphertext.
2. P_i computes the valid shares $s_j = d_j - m_j \bmod N$ for the other participants in A who passed the verification in the previous step.
3. P_i computes s by Lagrange interpolation in $\mathbb{Z}_N$ from a set of t valid shares, as in Shamir's secret sharing scheme.

The above PVSS scheme is unconditionally verifiable and it is IND2-secret under the Decisional Composite Residuosity (DCR) Assumption, and it is also additively homomorphic. The scheme does not make use of Fiat-Shamir non-interactive zero-knowledge proofs: instead it uses the homomorphic property of Paillier's encryption at the cost of an additional communication round in the distribution subprotocol.

B Generic Transformation from IND1 to IND2-Secrecy

Let us consider an IND1-secret PVSS scheme. Let $\mathsf{sharing}(S)$ be the information published by the dealer during the distribution subprotocol for a secret S. Let us assume that the set of possible secrets is a group $\mathcal{G}$, and let $\oplus$ denote the group operation.

A new *hybrid* PVSS scheme can be defined from the original one by letting the dealer choose and share a random secret $K \in \mathcal{G}$ and then publish $T = K \oplus S$ along with $\mathsf{sharing}(K)$. Obviously, this modification has no effect on the correctness and the public verifiability properties of the scheme. The reconstruction subprotocol is slightly modified by just adding a last step in which every participant computes $S = K^{-1} \oplus T$ after the computation of K. Let us show that if the basic scheme is IND1-secret, then the hybrid scheme is IND1-secret. Let $\mathcal{A}_2$ be an adversary playing the IND2 game in Definition 2 for the hybrid PVSS scheme, with a non-negligible advantage ε_2. We show an adversary $\mathcal{A}_1$ playing the IND1 game in Definition 1 for the basic scheme, also with a non-negligible advantage ε_1. Let $\mathcal{C}$ be the challenger for $\mathcal{A}_1$ in that game. $\mathcal{A}_1$ will act as a challenger for $\mathcal{A}_2$. In particular, $\mathcal{A}_1$ will forward all the corruption queries and responses of $\mathcal{A}_2$ to and from $\mathcal{C}$ during the game. The only nontrivial part of $\mathcal{A}_1$ is in step 3.

1. $\mathcal{A}_1$ forwards the distribution information from $\mathcal{C}$ to $\mathcal{A}_2$.
2. $\mathcal{A}_1$ forwards the corrupted participants' public keys from $\mathcal{A}_2$ to $\mathcal{C}$.
3. $\mathcal{A}_1$ receives $(\mathsf{sharing}(K_0), K_b)$ from $\mathcal{C}$, where $K_0, K_1 \stackrel{\$}{\leftarrow} \mathcal{G}$ and $b \stackrel{\$}{\leftarrow} \{0,1\}$ are chosen by $\mathcal{C}$. $\mathcal{A}_1$ also receives $S_0, S_1 \in \mathcal{G}$ from $\mathcal{A}_2$. Then, $\mathcal{A}_1$ picks $\beta \stackrel{\$}{\leftarrow} \{0,1\}$ and sends $\mathsf{sharing}(K_0)$ and $T_{b,\beta} = K_b \oplus S_\beta$ to $\mathcal{A}_2$.

4. $\mathcal{A}_1$ forwards the reconstruction information from $\mathcal{C}$ to $\mathcal{A}_2$.
5. If $\mathcal{A}_2$'s output β' equals β, then $\mathcal{A}_1$ outputs $b' = 0$. Otherwise $\mathcal{A}_1$ outputs $b' = 1$.

Notice that if $b = 0$, then $\mathcal{A}_1$ perfectly simulates a challenger for $\mathcal{A}_2$ since $T_{0,\beta} = K_0 \oplus S_\beta$ and then $\mathcal{A}_1$ sent a correct sharing of S_β for a random β. Otherwise $b = 1$, and then the view of $\mathcal{A}_2$ is independent of β. Indeed $T_{1,\beta} = K_1 \oplus S_\beta$, which is independent of $\mathsf{sharing}(K_0)$ and S_β. Hence, the probability that $\beta' = \beta$ is exactly $\frac{1}{2}$. So $\varepsilon_1 = \varepsilon_2/2$. On the other hand, $\mathcal{A}_1$ runs within the same time as $\mathcal{A}_2$ plus a small number of group operations.

This hybrid construction can be generalized to an arbitrary symmetric encryption scheme, $T = E_K(S)$, such that for any possible value of S, $E_K(S)$ is pseudorandom. Obviously, the above reduction should be modified to take into account the maximum advantage of an attacker against the pseudorandomness of the encryption scheme.

C Decisional Bilinear Square and Related Assumptions

We show here that the DBS Assumption is equivalent to the DBQ Assumption, which is defined below.

Assumption 2 (Decisional Bilinear Quotient (DBQ)). *Let G and G_1 be two groups of prime order q, g be a random generator of G and $e : G \times G \to G_1$ be a non-degenerated bilinear map. For $\mu, \nu, s \xleftarrow{\$} \mathbb{F}_q^*$, the probability distributions $D_0 = (g, g^\nu, g^\mu, T_0 = e(g,g)^{\nu/\mu})$ and $D_1 = (g, g^\nu, g^\mu, T_1 = e(g,g)^s)$ are polynomially indistinguishable.*

Lemma 3. *DBQ Assumption implies the DBS Assumption.*

Proof. We can solve the DBQ problem by using a solver for the DBS problem as follows. On input of a DBQ tuple $(g, u = g^\nu, v = g^\mu, T_b)$ we construct a correct DBS tuple $(v, g, u = v^{\nu/\mu}, T_b)$. Indeed, $T_0 = e(g,g)^{\nu/\mu}$ and T_1 is a random value independent of the rest of the tuple. □

Lemma 4. *DBS Assumption implies the DBQ Assumption.*

Proof. Similarly, on input of a DBS tuple $(g, u = g^\nu, v = g^\mu, T_b)$ we construct a correct DBQ tuple $(u, v = u^{\mu/\nu}, g = u^{1/\nu}, T_b)$. Indeed, $T_0 = e(u,u)^\mu = e(u,u)^{(\mu/\nu)(1/\nu)^{-1}}$ and T_1 is random and independent of the rest of the tuple. □

Lemma 5. *DBS Assumption implies the DBDH Assumption.*

Proof. On input of a DBS tuple $(g, u = g^\nu, v = g^\mu, T_b)$ we construct a correct DBDH tuple $(g, u, u^\gamma, v, T_b^\gamma)$ where $\gamma \xleftarrow{\$} \mathbb{F}_q^*$. Indeed, $T_0^\gamma = e(u,u)^{\mu\gamma} = e(g,g)^{\nu(\nu\gamma)\mu}$ and T_1^γ is random and independent of (g, u, u^γ, v). □

These relations are very similar to the relations between the Decisional Diffie Hellman (DDH), the Decisional Square Exponent (DSE) and the Decisional Inverse Exponent (DIE) Assumptions (see [13]).