Publicly Verifiable Secret Sharing from Paillier’s Cryptosystem

Alexandre Ruiz and Jorge L. Villar
Dept. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya.
C/Jordi Girona, 1-3, 08034 Barcelona, Spain
{aruiz,jvillar}@ma4.upc.edu

Abstract: In this paper we propose a simple PVSS scheme based on the homomorphic properties of Paillier’s encryption scheme. This new scheme is the first known PVSS scheme based on the decisional composite residuosity assumption. The verification process in this scheme is much simpler than in the other known schemes. Furthermore, in our proposal, verification is made non-interactive without using the Fiat-Shamir technique or any additional Zero Knowledge proof.

Keywords. Publicly verifiable secret sharing, Paillier’s encryption, homomorphic encryption, decisional composite residuosity assumption, semantic security.

1 Introduction

In a secret sharing (SS) scheme there are some participants and a dealer \( D \) who shares a secret with them. In general, threshold secret sharing schemes solve this problem in the case where all the players (dealer and participants) of the scheme are honest, but security is not guaranteed if one or more players are dishonest. Verifiable secret sharing (VSS) was proposed in [2] to solve the problem of dishonest dealers and dishonest participants who try to deceive other participants. In a publicly verifiable secret sharing (PVSS), introduced by Stadler [12], not only the participants can verify the validity of their shares, but also anyone can do it from the public information. Note that in a PVSS no private channels between \( D \) and the participants are assumed.

The originality of the scheme we propose lies in the holder of the secret key. When we analyze other PVSS we can see that every participant has a secret key and when the dealer wants to send them the shares or other information, he uses their respective public keys. On the contrary, in our scheme the dealer has a secret key and when he wants to broadcast the shares, he uses some information from the participants that they have sent him previously. This information holds a one-time key or session key (i.e., the participants must use a new key whenever they want to receive shares from another secret). However, we are using only one secret/public key, which yields a very simple scheme. Consequently
this represents an advantage with respect to other similar PVSS schemes like [5], [10] and [12], which use several secret/public keys.

Following the idea from Fujisaki and Okamoto in [5], a PVSS is called non-interactive if the verifier can verify the validity of the shares without interacting to any other player, not even the dealer. In this sense we state that our scheme is non-interactive, although there is some communication at the beginning of the protocol, when the participants must send some information.

PVSS schemes like [10] and [12] are interactive and use the Fiat-Shamir technique [4] for making such a protocols non-interactive. Pedersen in [8], however, does not use any additional Zero Knowledge proof and his scheme is non-interactive, but not publicly verifiable.

We propose a PVSS scheme with a non-interactive verification which, to our knowledge, is the first efficient PVSS scheme that does not use the Fiat-Shamir technique. Using Fiat-Shamir technique for non-interactive Zero Knowledge proofs has the drawback that security proofs are valid only in the random oracle model. In fact, there is a separation result about Fiat-Shamir heuristics (see [6]) that question the validity of security proofs in the real (i.e. without random oracle) model.

As we prove later in Section 3, the verification of all the players in our scheme (i.e. Dealer’s honesty in the share distribution process and participants’ honesty when they reconstruct the secret) is unconditionally secure since Paillier’s encryption function is bijective and the players only need to do a simple computation. This is the reason there is no need of Zero Knowledge proofs (not even against active adversaries). All these points are advantages of our scheme and accounts for our statement that the verification process is much simpler than in other PVSS in the literature.

Almost all known PVSS are based on the Diffie-Hellman assumption (e.g. [5], [10] or [12]) whereas we present a scheme which security (privacy) is based on the Decisional Composite Residuosity Assumption (DCRA), which comes from Paillier’s encryption (see [7]). However, the verification process of our scheme is unconditionally secure. If we compare security of our scheme and [8], we see that Pedersen’s scheme is unconditionally secure, but the verification process is only computationally secure.

The potential applications of this scheme are the same that for any PVSS, like e.g. multiparty computation, key-escrow and threshold cryptography.

The building blocks of our scheme are Shamir \((t, n)\)-threshold scheme (see [13]) and homomorphic Paillier’s encryption scheme. The proposed scheme can be adapted for more general access structures (e.g. vector space access structures) and it can also be extended to other additive homomorphic encryption schemes.

1.1 Basics and Notation

Let \(x \leftarrow X\) denote that \(x\) is a random element in the set \(X\). Let \(p, q\) be two large primes and \(N = p \cdot q\). We will denote by \(\phi(N) = (p-1) \cdot (q-1)\) Euler’s totient function and by \(L(u) = u^{\frac{1}{N}}\) such that \(u \in S_N = \{u \in \mathbb{Z}_N^* \mid u \equiv 1 \mod N\}\) the logarithmic function described in Paillier’s encryption scheme.
Paillier’s encryption of a message \( m \) is defined by the bijective function

\[
\varepsilon_g : \mathbb{Z}_N \times \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^* \times \mathbb{Z}_N^2
\]

\[(m, r) \mapsto g^m \cdot r^N \mod N^2\]

where \( r \) is taken at random and \( g \in \mathbb{Z}_N^* \) is an element with order a multiple of \( N \). Note that this encryption function is additively homomorphic, i.e. \( \varepsilon_g(m_1, r_1) \cdot \varepsilon_g(m_2, r_2) = \varepsilon_g(m_1 + m_2, r_1 \cdot r_2) \). To decrypt a ciphertext \( c \in \mathbb{Z}_N^2 \), the message \( m \) is obtained as follows:

\[
m = \text{Dec}(c) := \frac{L(c^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} \mod N
\]

where \( \lambda \) is the Carmichael’s function \( \lambda(N) = \text{lcm}(p-1, q-1) \). Note that \( \text{Dec}(c_1 c_2) = \text{Dec}(c_1) + \text{Dec}(c_2) \mod N \). In relation to the efficiency, \( g = 1 + N \) can be taken without affecting the security of the encryption scheme.

Let \( P = \{P_i : 1 \leq i \leq n\} \) be a set of \( n \) participants and \( \Gamma \) a monotone set of subsets of \( P \) (i.e. \( A \in \Gamma \) and \( A \subset B \subseteq \mathcal{P} \) implies \( B \in \Gamma \)). \( \Gamma \) is called the access structure and the subsets in \( \Gamma \) are called authorized subsets. In this paper we will consider only a \((t, n)\)-threshold access structure, that is, \( A \in \Gamma \) if and only if \( |A| \geq t \).

### 1.2 Overview

In section 2 we recall the subprotocols of a PVSS and present our particular construction, together with the correctness of the scheme. Moreover, we state an interesting homomorphic property of our scheme. In section 3 we study the public verifiability. In section 4 we study the security (privacy) of the scheme. We first introduce a semantic-security like notion and then we prove that our PVSS is semantically secure under this definition. In section 5 we analyze the security against active adversaries. In order to prove it we introduce some slight modifications in the PVSS scheme.

### 2 Publicly verifiable secret sharing scheme

#### 2.1 A model for PVSS schemes

We now proceed to describe a secure \((t, n)\)-threshold PVSS scheme. We work with a Dealer \( D \) who shares the secret, the set \( P \) of \( n \) participants where \( n > 0 \) and an external Verifier \( V \) who verifies whether \( D \) is honest from the publicly available information. Taking into account the ideas from [10], we have four important steps in a PVSS:

1. **Setup.** The dealer \( D \) computes a secret/public key pair and broadcasts the public key along with the system parameters.
2. **Distribution of the shares.** Here $D$ shares the original secret as follows. Every participant chooses a secret key which uses to broadcast some information. Then $D$ computes the shares of the secret and encrypts these shares with the public information of every participant. Then, $D$ broadcasts all the encrypted shares, along with some verification information.

3. **Verification of the shares.** At this step $V$ can verify the encrypted shares from the publicly available information.

4. **Reconstruction of the secret.** Here every participant $P$ of an authorized subset $A$ reveals some secret information to the others, so that the other participants in $A$ can both compute and verify the share of $P$. With this information, every participant in $A$ can reconstruct the secret.

We have two important subprotocols in a PVSS. On the one hand, the distribution protocol which is specified at steps 2 and 3 and, on the other hand, the reconstruction protocol which is specified at step 4.

### 2.2 The PVSS scheme

The setup of the scheme is described as follows. $D$ chooses two large primes $p, q$ and publishes $N = p \cdot q$, then selects $g \in \mathbb{Z}_N^*$, which also makes public, such that the order of $g$ in $\mathbb{Z}_N^*$ is $N$, e.g. $g = 1 + N$.

We use Paillier’s probabilistic encryption scheme where $(N, g)$ is the public key and $\lambda(N)$ is the secret key. Note that $D$ is the only entity that knows the secret key $\lambda$ and the inverse of $N$ mod $\phi(N)$.

$D$ shares with the Shamir $(t, n)$-threshold scheme over the ring $\mathbb{Z}_N$, where the secret $a_0 \in \mathbb{Z}_N$ is hidden in a random polynomial $a(x) = a_0 + a_1x + \cdots + a_{t-1}x^{t-1}$ with coefficients in $\mathbb{Z}_N$. Then $D$ computes the value $(i, s_i) \in \{1, 2, \ldots, n\} \times \mathbb{Z}_N$, that is $P_i$’s share, where $s_i = a(i) \mod N$ and when a subset $B \subseteq P$ of $t$ participants $P_1, \ldots, P_t$ want to reconstruct the secret $a_0$, then they only need to solve the following system of $t$ linear equations in the $t$ unknowns $a_0, \ldots, a_{t-1}$:

$$
\begin{pmatrix}
1 & i_1 & i_1^2 & \cdots & i_1^{t-1} \\
1 & i_2 & i_2^2 & \cdots & i_2^{t-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & i_t & i_t^2 & \cdots & i_t^{t-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{t-1}
\end{pmatrix}
= 
\begin{pmatrix}
s_{i_1} \\
s_{i_2} \\
\vdots \\
s_{i_t}
\end{pmatrix}
$$

with determinant $det(A) = \prod_{1 \leq k < j \leq t} (i_j - i_k)$.

In order to be able to recover the secret the participants need that $det(A) \in \mathbb{Z}_N^*$. This condition can be guaranteed by choosing $p, q$ such that $n < p, n < q$. We can force $D$ to choose these suitable $p, q$ by proving that $i \nmid N$, for all $i$ such that $i \in \{1, \ldots, n\}$.

From now on we suppose that $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, t-1\}$. We describe now
the protocol below:

1. $D$ chooses two large primes $p, q \mid N = p \cdot q$ and a random $g \in \mathbb{Z}_N^*$ with order $N$. Then he broadcasts $(N, g)$

2. Let $s$ be the secret to share
   (a) Every $P_i$ selects a random pair $(m_i, r_i) \in \mathbb{Z}_N \times \mathbb{Z}_N^*$ which remains private, then broadcasts $c_i = g^{m_i} \cdot r_i^N \mod N^2$
   (b) $D$ sets $a_0 = s \in \mathbb{Z}_N$ and, for the remaining $j$, selects $t - 1$ random $a_j \in \mathbb{Z}_N$. All these $a_j$ remain private. Then $D$ defines $a(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$ and sets $s_i = a(i) \mod N$, where every $P_i$ knows its own value $i$
   (c) $D$ obtains $\{(m_i, r_i)\}_i$ using $m_i = \frac{L(c_i^2 \mod N^2)}{L(g^3 \mod N^2)} \mod N$ and $r_i = \left(\frac{c_i}{g^{m_i}}\right)^{N-1} \mod N$. Then $D$ broadcasts $\{d_i\}_i$, with $d_i = s_i + m_i \mod N$

3. (a) $D$ selects $t$ random $r'_{ij} \in \mathbb{Z}_N^*$, which remain private, and broadcasts $\{A_j\}_j$ and $\{t_i\}_i$, where $A_j = g^{a_j} \cdot r_j^N \mod N^2$ and $t_i = r'_0 \cdot r'_1 \cdots r'_{t-1} \cdot r_i \mod N$
   (b) $V$ checks that $A_0 \cdot A_1^i \cdots A_{t-1}^i = \frac{g^{d_i}}{c_i} \cdot t_i^N \mod N^2$ for every $i$

4. Let $A$ be an minimal authorized subset of participants ($|A| = t$)
   (a) Every $P_i \in A$ sends $(m_i, r_i) \in \mathbb{Z}_N \times \mathbb{Z}_N^*$ to the other participants in $A$, which check that $c_i = g^{m_i} \cdot r_i^N \mod N^2$ and compute $s_i = d_i - m_i$
   (b) Every participant in $A$ computes $a_0 = \sum_{i \mid P_i \in A} \left(\prod_{1 \leq h \leq t, h \neq i} \frac{h}{h - i}\right) \cdot s_i$

**Scheme. Publicly Verifiable Secret Sharing Scheme.**

Note that $V$ uses the homomorphic property of Paillier’s scheme at step 3 in this protocol. The correctness of the scheme means that a honest $D$ always pass the verification procedure, and honest participants always recover a unique secret, that is the same shared by $D$. These requirements can be easily checked in the above protocol.

---

1. We assume here the existence of a broadcast authentication protocol for our scheme. We can use e.g. the BiBa broadcast authentication protocol, which allows all receivers to verify the origin of the data. See [9] for more information. But it is sufficient to take a broadcast channel without authentication and a digital signature scheme for the participants to sign their $c_i$ at step 2(a) in order to assure that the honest participants receive their shares.
2.3 The homomorphic property

Following the idea from [1], we say that a SS scheme has the homomorphic property if the sum of the shares of two secrets $s$ and $\bar{s}$ sent to the participants are shares of the sum of secrets $s + \bar{s}$. Therefore, the participants are able to recover the sum of secrets only knowing the shares from $s$ and $\bar{s}$.

In the case of a PVSS, we say that such a scheme has this property when, in addition, the verification of the shares of the new secret $s + \bar{s}$ can be done from the broadcasted information for $s$ and $\bar{s}$.

Our scheme has the homomorphic property since Shamir’s scheme also has it. In relation to the verification process, if the elements $\{c_i, d_i, t_i\}_i$, $\{\bar{A}_j\}_j$ and $\{c_i, \bar{d}_i, \bar{t}_i\}_i$, $\{\bar{A}_j\}_j$ are used in the verifications of the shares of $s$ and $\bar{s}$ respectively, then it is easy to prove the equality

$$A_0 \bar{A}_0 (A_1 \bar{A}_1)^{t_1} \cdots (A_{t-1} \bar{A}_{t-1})^{t_{t-1}} = \frac{q^{d_i + \bar{d}_i}}{c_i \bar{c}_i} \cdot (t_i \bar{t}_i)^N \mod N^2,$$

for every $i$ is satisfied.

Then there is no need to aggregate any information for the verification of $s$ and $\bar{s}$ but only to store the values $A_j \bar{A}_j$, $d_i + \bar{d}_i$, $c_i \bar{c}_i$ and $t_i \bar{t}_i$ as if $s + \bar{s}$ had been directly shared. Note that this last property is not achieved if the protocol make use of a typical Zero Knowledge proof in the verification process.

3 Analysis of the verification

In this section we show that if $D$ passes the verification step then all players must be honest in this PVSS, i.e., on the one hand, the dealer must be honest in the distribution protocol and, on the other hand, the participants must be honest in the reconstruction protocol.

3.1 Distribution protocol

In the following, we are going to prove that when $D$ is dishonest he can convince nobody. In other words, when $\nu$ checks the third point in the protocol and he is convinced from his result, then all sets in $\Gamma$ reconstruct the same secret.

Let $Rec(A, S_A)$ denote the result of the reconstruction protocol executed by an authorized subset $A \subseteq P$ such that $|A| = t$, where $S_A = (s_i)_{i \in A}$ are the shares hold by the participants in $A$.

**Definition 3.1** We say that $D$ deceives if there exist different $A_1, A_2 \in \Gamma$ such that
Rec(A_1, S_{A_1}) \neq Rec(A_2, S_{A_2}). In the case that D cannot deceive, we say that D is honest.

**Theorem 3.2** If V accepts at step 3 in the scheme, then D must be honest.

**Proof.** In order to prove that all the authorized subsets in \( \Gamma \) reconstruct the same secret, it’s sufficient to prove that the secret any minimal authorized subset reconstruct from their shares is equal to the secret encrypted in \( A_0 \).

Let us suppose that every participant \( P_i \) has as share \( s_i = d_i - Dec(c_i) \), where \( c_i \) has been broadcasted by \( P_i \) and \( d_i \) by \( D \), and \( V \) has the elements \( A_j = g^{a_j} \cdot r^j \mod N^2 \), for \( j = 0, \ldots, t-1 \), broadcasted by \( D \). Since \( V \) has verified that \( A_0 \cdot A_1 \cdot A_{t-1} = g^{d_i} \cdot t^N \mod N^2 \), then \( Dec(A_0) = Dec(A_1) = \cdots = Dec(A_{t-1}) = a_0 + a_1 i + \cdots + a_{t-1} i^{t-1} \mod N \) and any authorized subset \( A \) will reconstruct the same secret \( a_0 \) by Lagrange interpolation.

So \( D \) is always committed to give the real secret \( s \) and it is guaranteed that any authorized set of participants obtain the same secret \( s \) in the reconstruction protocol. Therefore we don’t need any additional proof other than the information broadcasted at steps 2 and 3.

From now on we can suppose that the dealer is always honest with the other players.

### 3.2 Reconstruction protocol

When the \( t \) participants \( P_i \) of an authorized subset \( A (|A| = t) \) meet in order to reconstruct the secret \( s = a_0 \), every participant \( P_i \in A \) must open his commitment \( c_i \) by sending \( (m_i, r_i) \) to the others. After that, all participants in \( A \) can compute the others’ shares and recover the secret by Lagrange interpolation.

In the following we are going to remark that none of the participants can trick the other participants in this protocol, i.e., if some participant give a different information then the other participants, without interaction, will know that he is trying to cheat them.

To see this, notice that a \( P_i \) cannot cheat \( P_j \) because for every \( c_i \in \mathbb{Z}_{N^2}^* \) there exist unique \( m_i \in \mathbb{Z}_N \) and \( r_i \in \mathbb{Z}_N^* \) such that \( c_i = g^{m_i} \cdot r_i \mod N^2 \). Therefore, \( P_i \) is forced to send \( (m_i, r_i) \). Otherwise, \( P_i \) will be identified as a cheater.

At this point, cheating participants can be excluded from the protocol and the secret could be reconstructed if there remain at least \( t \) honest participants.
4 Security of the scheme

Our goal now is to see that any unauthorized subset not only cannot reconstruct the secret but also obtain no information about the secret. First of all we introduce a semantic security notion for a PVSS scheme. Furthermore, we’ll see that our PVSS scheme is semantically secure against passive adversaries.

In order to prove the security of the scheme, we suppose that there exists an adversary $\mathcal{A}$ that corrupts the participants of some $B \subset \mathcal{P}$ such that $|B| = t - 1$ (worst case).

Let $\mathcal{S}$ be the set of secrets and $s \in \mathcal{S}$ be the secret distributed by $D$. We will use from now on the following notation: $\text{Sec}_B$ contains all the secret information of all the participants in $B$, $\text{Pub}$ contains the public parameters, coming from the setup step and the broadcasted information, and $\text{View}_{AB} = (\text{Pub}, \text{Sec}_B)$ is $\mathcal{A}$’s view of the setup and distribution protocol.

DEFINITION 4.1 Let $s$ be as above. We say that the PVSS is semantically secure if for all $B \subset \mathcal{P}$ such that $|B| = t - 1$ and any adversary $\mathcal{A}$ who controls all the participants in $B$, the following two probability distributions are polynomially indistinguishable

\[
\begin{align*}
D_{\text{secret}} &= (\text{View}_{AB}, s) \\
D_{\text{random}} &= (\text{View}_{AB}, s'), \quad s' \leftarrow \mathcal{S}
\end{align*}
\]

We are going to prove that the scheme is semantically secure using the following Assumption, in which semantic security of Paillier’s cryptosystem is based.

Assumption. [Decisional Composite Residuosity Assumption (DCRA)]: Let $p, q$ be two different $l$-bit primes such that $N = p \cdot q$ and $g \in \mathbb{Z}^{*}_{N^2}$ with order $N$. The following two probability distributions are polynomially indistinguishable

\[
\begin{align*}
D_{\text{residue}} &= (N, g, \rho^{N}), \quad \rho \leftarrow \mathbb{Z}^{*}_{N} \\
D_{\text{random}} &= (N, g, y), \quad y \leftarrow \mathbb{Z}^{*}_{N^2}
\end{align*}
\]

i.e, there exists no probabilistic polynomial time distinguisher for $N$-th residues modulo $N^2$. See [7] for more information.

**Theorem 4.2** Let $p, q$ be two different $l$-bit primes such that $N = p \cdot q$ and $g \in \mathbb{Z}^{*}_{N^2}$ with order $N$. The proposed PVSS scheme is semantically secure under DCRA.

**Proof.** We are in the particular case where $\text{Sec}_B = (\{m_i\}_{i \in B}, \{r_i\}_{i \in B}, \{s_i\}_{i \in B})$, $\text{Pub} = (N, g, \{c_i\}_{1 \leq i \leq n}, \{d_i\}_{1 \leq i \leq n}, \{t_i\}_{1 \leq i \leq n}, \{A_j\}_{0 \leq j \leq t - 1})$ and $\mathcal{S} = \mathbb{Z}_N$.

We are going to construct now a distinguisher $\mathcal{F}'$ for DCRA from any distinguisher $\mathcal{F}$ for semantic security of the PVSS (see Figure 1).

The input of $\mathcal{F}'$ can be generated as follows: $N$ and $g$ as generated as specified in DCRA. Then a random bit $b \in \{0, 1\}$ is chosen and we define $z = g^{\mu} \cdot \rho^{N} \in \mathbb{Z}^{*}_{N^2}$, where $\rho \leftarrow \mathbb{Z}^{*}_{N}$ and $\mu = 0$, if $b = 0$, otherwise $\mu \leftarrow \mathbb{Z}_N$.

The goal of $\mathcal{F}'$ is to guess the bit $b$. To do this, $\mathcal{F}'$ chooses $d_i \leftarrow \mathbb{Z}_N$, $t_i \leftarrow \mathbb{Z}_N$, for all $i \in \{1, \ldots, n\}$ and $r_i \leftarrow \mathbb{Z}_N$, $s_i \leftarrow \mathbb{Z}_N$, for all $P_i \in B$. Let us assume that $B = \{1, \ldots, t - 1\}$ without loss of generality. Now, $\mathcal{F}'$ constructs the other elements $m_i, A_j, c_i$ as follows:
• $m_i = d_i - s_i$ for every $1 \leq i \leq t - 1$.

• $\mathcal{F}'$ chooses $s' \leftarrow \mathbb{Z}_N$, then computes $A_0 = z_b \cdot g^{s'}$. Note that the secret $s$ must verify $s = s' + \mu$. So, $s = s'$ if $b = 0$, otherwise $s'$ and $s$ are independent.

• $A_j$, $1 \leq j \leq t - 1$, are computed from $A_0$ and $(r_i, s_i)$, for every $P_i \in B$. We define $s_i = a_0 = s$ and let $a_1, \ldots, a_{t-1}$ be the unique solution of the system of linear equations $s_j = a_0 + a_1 j + a_2 j^2 + \cdots + a_{t-1} j^{t-1}$, $1 \leq j \leq t - 1$. Then we can write $A_j = \sum_{i=0}^{t-1} \nu_{ij} \cdot s_i$ for some efficiently computable coefficients $\nu_{ij}$. Therefore, $g^{s_i} = (g^\nu)^{\nu_{ij}} \cdot \prod_{i=1}^{t-1} (g^{a_i})^{\nu_{ij}}$ which enables us to compute the values $A_j = (g^\nu \cdot r_i^N)^{\nu_{ij}} \cdot \prod_{i=1}^{t-1} (g^{a_i \cdot r_i^N})^{\nu_{ij}} = A_0^{\nu_{ij}} \cdot \prod_{i=1}^{t-1} (g^{a_i \cdot r_i^N})^{\nu_{ij}}$, for all $j$ such that $1 \leq j \leq t - 1$.

• $c_i = \frac{g^{a_i \cdot r_i^N}}{A_0 \cdot A_1 \cdots A_{t-1}}$, $\forall i \in \{1, \ldots, n\}$. In the special case of $i \in \{1, \ldots, t-1\}$, $c_i$ can also be computed using $c_i = g^{a_i} \cdot r_i^N$.

Now $\mathcal{F}'$ gives the tuple $(\text{View}_{A_B}, s')$ to $\mathcal{F}$ who guesses the bit $b'$ that is also used as the output of $\mathcal{F}'$.

It is straightforward to prove that the probability distribution of $(\text{View}_{A_B}, s')$ generated as above by $\mathcal{F}'$ is the same as $(\text{View}_{A_B}, s)$, if $b = 0$, and $(\text{View}_{A_B}, s')$, if $b = 1$, in the definition of semantic security. Then $\mathcal{F}$ has exactly the same advantage as $\mathcal{F}$, and runs roughly in the same time as $\mathcal{F}$ plus the running time of the distribution protocol. \qed

\[ (N, g, z_b) \longrightarrow (\text{View}_{A_B}, s') \longrightarrow \mathcal{F} \longrightarrow b' \longrightarrow b' \]

\[ \text{Figure 1. Distinguisher $\mathcal{F}'$.} \]

5 Active adversaries

The proposed PVSS scheme, with some slight modifications, is also secure against active adversaries. In this case, corrupted participants can actively do what they want whereas honest ones behave correctly all the time.

We describe now the modified scheme following the original. Let $\mathcal{P}$ be the set of $n$ participants. We need $t$ of them to recover the secret and we have, in the worst case, $t - 1$ corrupted participants. Therefore, we need that $n \geq 2t - 1$ if the scheme is secure against active adversaries (whereas we only need $n \geq t$ if the adversary is passive).

At step 2(a) every participant $P_i$ must broadcast $c_i$. If $c_i$ is not suitable (e.g. $c_i \notin \mathbb{Z}_N^*$) or nothing is broadcasted, then this participant is excluded from the protocol. Hence we have
some $\mathcal{P}_1 \subseteq \mathcal{P}$, which contains the participants who were not excluded at step 2(a).
At steps 2(c) and 3(a), $D$ must broadcast $\{d_i\}_i$ and $\{A_j\}_j$, $\{t_i\}_i$, respectively. At step 3(b),
every participant checks if $A_0 \cdot A_1 \cdot \cdots \cdot A_{t-1} = \frac{g^{d_i} \cdot t_i^N}{c_i} \mod N^2$. If one of these last three
steps has not satisfactorily occurred, honest participants decide that $D$ is corrupted. Then
honest participants complain against $D$ using the broadcast channel and they drop out of
the protocol. So, the protocol is aborted since there remain at most $t - 1$ participants in
the protocol.
From now on we can suppose that $D$ is honest, i.e. steps 2(c), 3(a) and 3(b) have satisfactorily occurred. At step 4(a), every participant $P_i \in \mathcal{P}_1$ broadcasts his own $(m_i, r_i)$. Then, every participant checks if $c_i = g^{m_i} \cdot r_i^N \mod N^2$ from all the other participants $P_i \in \mathcal{P}_1$. This leads to defne $\mathcal{P}_2 = \{P_i | P_i \in \mathcal{P}_1, P_i$ broadcasts $(m_i, r_i)$ such that $c_i = g^{m_i} \cdot r_i^N \mod N^2 \}$. Then the participants of $\mathcal{P}_1$ who are not in $\mathcal{P}_2$ are excluded and the
remaining participants can recover the secret by interpolation from $t$ of the shares they
have computed.
All the non mentioned steps in the protocol remain unmodified.
In order to prove that the participants of $\mathcal{P}_2$ can recover the secret, we prove the following
theorem using the same notation as above.

**Theorem 5.1** If there exists some active adversary who controls at most $t - 1$ participants, 
then during the protocol execution one of the following holds:

1. All honest participants realize that $D$ is corrupt and nobody can recover the secret.
2. Let $A$ be a subset from $\mathcal{P}_1$ such that $|A| \geq n_1 - (n - 2t + 1)$, where $n_1 = |\mathcal{P}_1|$ and $n \geq 2t - 1$ running the reconstruction protocol. Then all honest participants in $A$ recover the same secret.

**Proof.** If $D$ is corrupted then all the honest participants accuse $D$ and drop out from the protocol. That means that at most $t - 1$ participants (the corrupted ones) remain and they cannot recover the secret anymore. If this happens, we are in case 1.

If $D$ is honest then the protocol follow with step 4. We note that $n_1 \geq n - (t - 1) \geq t$ and that at step 2(a) we have at most $(t - 1) - (n - n_1)$ corrupted participants to be uncovered.
As there are at most $(t - 1) - (n - n_1)$ corrupted participants in $A$ and, by hypothesis, $|A| \geq n_1 - (n - 2t + 1)$ then there are at least $t$ honest participants in $A$. Hence, since all honest participants of $A$ are in $\mathcal{P}_2$ we get that $|\mathcal{P}_2| \geq t$ and all honest participants in $A$ can recover the secret from any subset of $t$ different shares, i.e. we are in case 2. 

**Acknowledgements**

Many thanks to the anonymous reviewers for their useful comments, which have allowed to improve this paper.
References


