A Practical (Non-interactive) Publicly Verifiable Secret Sharing Scheme

Mahabir Prasad Jhanwar

C R RAO Advanced Institute of Mathematics, Statistics and Computer Science
University of Hyderabad Campus
Hyderabad, India
mahavir.jhawar@gmail.com

Abstract. A publicly verifiable secret sharing (PVSS) scheme, proposed by Stadler in [29], is a VSS scheme in which anyone, not only the shareholders, can verify that the secret shares are correctly distributed. PVSS can play essential roles in the systems using VSS. Achieving simultaneously the following two features for PVSS is a challenging job:

– Efficient non-interactive public verification.
– Proving security for the public verifiability in the standard model.

In this paper we propose a (t, n)-threshold PVSS scheme which satisfies both of these properties. Efficiency of the non-interactive public verification step of the proposed scheme is optimal (in terms of computations of bilinear maps (pairing)) while comparing with the earlier solution by [18]. In public verification step of [18], one needs to compute $2n$ many pairings, where n is the number of shareholders, whereas in our scheme the number of pairing computations is 4 only. This count is irrespective of the number of shareholders. We also provide a formal proof for the semantic security (IND) of our scheme based on the hardness of a problem that we call the (n, t)-multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [5].

Keywords: Secret sharing, non-interactive PVSS, general Diffie-Hellman exponent problem.

1 Introduction

(Verifiable) Secret Sharing is one of the most important tools in modern cryptography. The concept and the first realization of secret sharing were presented independently in [26] and in [4]. Since then much work has been put into the investigation of such schemes (see [28,30] for a list of references). In a secret sharing scheme, there exists a dealer and n shareholders (sometimes referred to participants). The dealer splits a **secret**, say s, into n different pieces, called **shares**, and sends one share to each shareholder. An access structure describes which subsets of shareholders are qualified to recover the secret. By a (t, n)-threshold access structure, $1 \leq t \leq n$, we means that any subset of t or more

F. Bao and J. Weng (Eds.): ISPEC 2011, LNCS 6672, pp. 273–287, 2011.

shareholders will be able to recover the secret; any smaller subset of shareholders will not be able to gain any information about the secret.

The verifiable secret sharing (VSS) schemes constitute a particular interesting class of schemes as they allow each receiver of information about the secret (share of the secret) to verify that the share is consistent with the other shares. If the dealer trusts one of the shareholders completely, he could share the 'whole' secret with the person and thus altogether avoid the trouble of using a secret sharing scheme. Therefore in many applications the dealer doesn't trust the shareholders completely, and therefore it is reasonable to expect that (some of) the shareholders do not trust the dealer either. For this reason efficient verifiable secret sharing schemes are necessary in practice. Verifiable secret sharing was proposed first in [10]. In a VSS scheme, the shareholders can verify the validity of their shares and thus overcome the problem of dishonest dealers. VSS is known to play important roles in various cryptographic protocols such as the multiparty protocols [9,3], key-escrow cryptosystems [20], and threshold cryptography. A VSS scheme is called **non-interactive** if the shareholders can verify their share without talking to each other or the dealer. Proposals by [13,23] contributed to non-interactiveness and improved efficiency.

(Non-interactive) Publicly Verifiable Secret Sharing: The first proposed VSS scheme [10] has the special property that anyone, not only the shareholders, can verify that the shares were correctly distributed. In [29], the property was named **public verifiability** and the VSS schemes with the above property were named publicly verifiable secret sharing schemes (PVSS). Some of the important PVSS schemes were presented in [29,15,25].

In most PVSS schemes, the verification procedure involves interactive proofs of knowledge. These proofs are made non-interactive by means of the Fiat-Shamir technique [14] and thus security for verifiability can only be carried out in the random oracle model [2]. Transforming security analysis of cryptographic primitives from the framework of random oracle model to the standard model have always turned out to be a theoretically important task which is seemingly difficult in most of the cases. Some of these problems were dealt in [24,18]. Some of the positive features of [18] are: non-interactive PVSS, Fiat-Shamir technique is not used, unconditional security for public verifiability and security for indistinguishability of secrets. Although, [18] successfully avoids Fiat-Shamir technique, their public verification algorithm is inefficient. In particular, for n shareholders, one has to compute $2n$ many pairings in the public verification algorithm. This number of pairing computations is expensive. Therefore, an important problem was to reduce the number of pairing computations during the public verification algorithm.

Our Contribution: In this paper we propose a practical and provably secure non-interactive (t, n)-threshold PVSS scheme. Our scheme achieves the following:

- Public verification algorithm is non-interactive and is obtained without using Fiat-Shamir zero knowledge proofs.
- Comparing with the public verification step of [18], our scheme provides optimal efficiency in terms of the number of pairing computations. In public

verification step of [18], one needs to compute $2n$ many pairings, where n is the number of shareholders, whereas in our scheme the number of pairing computations is 4 only. This count is irrespective of the number of shareholders.

- The scheme is provably secure against a SA-IND (see Section 2.2) adversary. The security relies on the hardness of a problem that we call the (n, t)- multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [5].

2 Preliminaries

In this section we describe the algorithms that form a non-interactive (t, n)-threshold publicly verifiable secret sharing (PVSS) scheme, as well as the basic security requirements for such schemes. We also introduce the computational problem called the (n, t)-MSE-DDH problem, to which we will relate the security of our scheme.

2.1 (Non-interactive) PVSS

In this section we describe a model for non-interactive PVSS. In a PVSS scheme, a dealer D wishes to distribute shares of a secret value "s" among n shareholders $P_1, \ldots, P_n$. In this article, we consider (t, n)-threshold access structure, $1 \leq t \leq n$. A PVSS scheme is described by the following standard algorithms.

- **Initialization:** This algorithm generates all system parameters. Furthermore, each shareholder P_i registers its public-key (may be issued by the dealer with the corresponding secret key). The actual set of shareholders taking part in a run of PVSS scheme must be a subset of the registered shareholders. We assume w.l.o.g. that shareholders $P_1, \ldots, P_n$ are the actual shareholders in the run described below.
- **Distribution:** The distribution of the shares of a secret "s" is performed by the dealer D. The dealer computes and publishes the secret commitment value(s) and the share deriving value(s) respectively. The secret commitment value(s) ensures the dealer's commitment to the value of secret s, whereas the share deriving value(s) can be used with the shareholders' secret keys to derive the respective shares.
- **Verification:** It is required that the dealer's commitment to the secret can be verified *publicly*. Thus any party knowing only the publicly available information may verify that share deriving information is consistent with the share commitment information, i.e., it guarantees that the reconstruction protocol will be able to recover the same secret s. Furthermore, this verification runs non-interactively.
- **Reconstruction:** The shareholders construct their shares S_i from the share deriving value using the secret keys. It is not required that all shareholders succeed in doing so, as long as a qualified set of shareholders is successful.

These shareholders then release S_i and also the share commitment value(s) to verify that the released shares are correct. The share commitment information is used to exclude the shareholders which are dishonest or fail to reproduce their share S_i correctly. Reconstruction of the secret s can be done from the shares of any qualified set of shareholders.

In non-interactive PVSS schemes it is essential that all commitments can be verified non-interactively. Since any party can verify the output of the dealer, so we don't budget operations for the individual participants to check their own shares. Hence it suffices to have just one public verifier.

2.2 Security Model

Such a scheme must satisfy the following properties.

- **Correctness:** If the dealer and the shareholders act honestly, every qualified subset of shareholders reconstructs the secret during the reconstruction algorithm.
- **Verifiability:** If a dealer passes the verification step, then it implies that the secret commitment values are consistent with the share deriving values.
- **Privacy:** The very basic requirement is that, for an honest dealer, the adversary cannot learn any information about the secret at the end of the protocol.

Privacy: Following [24,18], we can more formally define the above privacy notion, under the classical semantic-security notion [17], using a game between an adversary $\mathcal{A}$ and a challenger. The adversary here is a static one i.e., at the beginning of the game, he is given the secret keys of the corrupted shareholders.

Indistinguishability of Secrets (IND): The security notion is defined via the following game between a challenger and a probabilistic polynomial time (PPT) adversary $\mathcal{A}$. Both the adversary and the challenger are given as input a security parameter λ.

- **Initialization:** The challenger runs Initialization(λ) to obtain the set of public parameters along with the public keys and the secret keys of all the shareholders. Besides all the public keys, the adversary is also given the secret keys of $t-1$ corrupted shareholders.
- **Challenge:** The challenger picks two random secrets T_0 and T_1 and a random bit $b \in \{0,1\}$. Then he runs the distribution algorithm for the secret T_b and sends all the resulting information to $\mathcal{A}$ along with $\{T_0, T_1\}$.
- **Guess:** Finally, the adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b' = b$.

We define the advantage of this static adversary (SA), $\mathcal{A}$, against a (t,n)-threshold PVSS as follows:

$$\mathrm{Adv}_{PVSS,\mathcal{A}}^{SA-IND}(\lambda) = \left|\mathrm{Prob}[b' = b] - \frac{1}{2}\right|$$

The advantage is a function of the security parameter λ.

Definition 1. *We say that a* (t, n)*-threshold PVSS scheme is SA-IND secure if for all PPT adversaries* $\mathcal{A}$*, we have that* $Adv_{PVSS,\mathcal{A}}^{SA-IND}(\lambda)$ *is a negligible function in* λ*.*

2.3 Bilinear Map

Let G_1, G_2 and $\tilde{G}$ be three cyclic groups of prime order p. The group laws for all the three groups are noted multiplicatively. A mapping $e : G_1 \times G_2 \rightarrow \tilde{G}$ is called an admissible bilinear map (pairing) if it satisfies the following properties:

- **Bilinearity:** $e(g_1^{\alpha}, g_2^{\beta}) = e(g_1, g_2)^{\alpha\beta}$ for all $g_1 \in G_1, g_2 \in G_2$ and $\alpha, \beta \in \mathbb{Z}_p$.
- **Non-degeneracy:** $e(g_1, g_2) \neq 1$ unless $g_1 = 1$ or $g_2 = 1$.
- **Computability:** There exist efficient algorithms to compute the group operations in $G_1, G_2, \tilde{G}$ as well as the map $e(\cdot, \cdot)$.

A bilinear map group system is a tuple $(p, G_1, G_2, \tilde{G}, e(\cdot, \cdot))$ composed of the objects as described above. The above bilinear map is defined in *asymmetric* setting [7,8]. Also in asymmetric setting, existence of an efficiently computable isomorphism $\phi : G_2 \rightarrow G_1$ is known [21,22]. As we see later, we make use of a bilinear map group system where we require the existence of an efficient isomorphism goint from G_2 to G_1. In symmetric setting, we have $G_1 = G_2$. Known examples of $e(\cdot, \cdot)$ usually have G_1, G_2 to be the groups of Elliptic Curve or Hyperelliptic Curve points and $\tilde{G}$ to be a subgroup of a multiplicative group of finite field. Modified Weil pairing [6], Tate pairing [1,16] are some of the practical examples of bilinear maps.

2.4 (n, t)-MSE-DDH (The Multi-Sequence of Exponents Diffie-Hellman Assumption)

Our scheme's security relies on the hardness of a problem that we call the (n, t)-multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework [5]. Some of the problems that are similar to (n, t)-MSE-DDH, were considered in [11,19,12] and all of them fit the framework of general Diffie-Hellman exponent problem. [5] provides an intractability bound for the general Diffie-Hellman exponent problem in the generic model [27], where the underlying groups are equipped with pairings. Thus the generic complexity of (n, t)-MSE-DDH and the other similar problems mentioned in [11,19,12] are covered by the analysis in [5]. A proof to show the (n, t)-MSE-DDH problem as a particular instance of general Diffie-Hellman exponent problem is similar to the proof of [12], where it has been shown that the (l, m, t)-MSE-DDH (l, m, t are integers) problem fit the framework of general Diffie-Hellman exponent problem.

Let $G_1, G_2, \tilde{G}$ be the three groups of the same prime order p, and let $e : G_1 \times G_2 \rightarrow \tilde{G}$ be a non-degenerate and efficiently computable bilinear map. Let g_1 be a generator of G_1 and g_2 be a generator of G_2.

Let n, t be two positive integers ($t \leq n$). The (n, t)-multi-sequence of exponents Diffie-Hellman problem ((n, t)-MSE-DDH) related to the group triplet $(G_1, G_2, \tilde{G})$ is as follows:

– **Input:** Two polynomials θ_1, θ_2 as

$$\theta_1(x) = \prod_{i=1}^{n}(x + a_i) \text{ and } \theta_2(x) = \prod_{i=1}^{n-t+1}(x + b_i)$$

where $a_1, \ldots, a_n$ and $b_1, \ldots, b_{n-t+1}$ are all distinct elements in $\mathbb{F}_p$. Thus degrees of θ_1, θ_2 are n and $n - t + 1$ respectively. We call $a_1, \ldots, a_n$ and $b_1, \ldots, b_{n-t+1}$ to be the **negative roots** of θ_1, θ_2 respectively. Beside polynomials θ_1, θ_2, the following sequences of exponentiations are also given as input,
  - $\hat{g_1} := [g_1, g_1^{\alpha}, \{g_1^{\gamma^i}\}_{i=1}^{n+t-2}, \{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t} \text{ and } g_1^{k\alpha\theta_1(\gamma)}]$,
  - $\hat{g_2} := [g_2, g_2^{\alpha}, \{g_2^{\gamma^i}\}_{i=1}^{n-t-1}, \{g_2^{\alpha\gamma^i}\}_{i=1}^{n} \text{ and } g_2^{k\theta_2(\gamma)}]$,
  - an element $T \in \tilde{G}$,

  where $k, \alpha, \gamma \in \mathbb{F}_p^*$ and are not known.
– **Output:** a bit $b \in \{0, 1\}$ as,

$$b = \begin{cases} 1 & \text{if } T = e(g_1, g_2)^{k\theta_1(\gamma)} \\ 0 & \text{if } T \text{ is a random element of } \tilde{G} \end{cases}$$

Thus the problem is to distinguish if T is a random value or if it is equal to $e(g_1, g_2)^{k\theta_1(\gamma)}$. To be more precise, let us denote by **real** the event that $T = e(g_1, g_2)^{k\theta_1(\gamma)}$, by **random** the event that T is a random element from $\tilde{G}$ and by $\mathcal{I}(\theta_1, \theta_2, \hat{g_1}, \hat{g_2}, T)$ the input of the problem. Let λ be the size of the underlying group order. We define the **advantage** of an algorithm $\mathcal{A}$ in solving (n, t)-MSE-DDH problem as

$$\text{Adv}_{\mathcal{A}}^{(n,t)-\text{MSE-DDH}}(\lambda) = \Big|\Pr[\mathcal{A}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = 1|\text{real}] - \Pr[\mathcal{A}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = 1|\text{random}]\Big|$$

where the probability is taken over all the random coins consumed by $\mathcal{A}$.

3 The New (t, n)-threshold PVSS Scheme

The earlier proposals for (publicly) verifiable secret sharing scheme mostly rely on the idea of interpolating (Lagrange Interpolation) a polynomial on the exponent of a generator of a group. A sketch of the idea can be given as follows:

– Fix a cyclic group G of prime order p and a generator $g \in G$.
– Choose a polynomial $f \in \mathbb{F}_p[x]$ of degree $t - 1$, say $f(x) = a_0 + a_1x + \cdots + a_{t-1}x^{t-1}$.
– The polynomial is kept secret but a **commitment** to the polynomial is published by publicly distributing the coefficient of f on the exponent of g. Shares (usually $f(i)$'s for the ith shareholder) are also published on the exponents of g.

– When t or more participants come together, they can interpolate f on the exponent of g, i.e., $g^{f(x)}$.

Our proposal, though works with polynomial interpolation, is based on a different approach. This idea is very prominent in threshold cryptography, e.g., broadcast encryption, threshold encryption, attribute based encryption etc. The approach for our scheme is inspired by the work of [11,19,12]. An overview of this idea can briefly be described as follows:

– Fix a cyclic group G of prime order p and a generator $g \in G$.
– Choose a polynomial $f \in \mathbb{F}_p[x]$ and **publish** it (unlike the earlier approach, f is not kept secret).
– Instead what is kept secret is a **value** (say $\gamma \in \mathbb{F}_p$) where this polynomial would later be evaluated. Some public information is made available so that one can compute $g^{f(\gamma)}$.

Scheme: Now we describe a (t, n)-threshold publicly verifiable secret sharing scheme. A special property of this scheme is that the participants are initially issued secret keys such that for every new secret that the dealer wants to share, the participants can use the same secret keys to derive the respective shares of the secret in question. Let λ be the underlying security parameter of the system.

– **Initialization:** This algorithm consists of two steps:
  • Setting up public parameters: Generates a bilinear map group system $(p, G_1, G_2, \tilde{G}, e(\cdot,\cdot))$. Let $\phi : G_2 \to G_1$ be an efficiently computable group isomorphism. Also, two generators $g \in G_1$ and $h \in G_2$ are randomly selected as well as the secret values $\alpha, \gamma \in \mathbb{F}_p^*$. We assume that p is significantly larger than n. The dealer then computes and publishes

$$\left(u = g^{\alpha\gamma}, h, h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}\right)$$

  • User keys generation: There are n participants $P_1, \ldots, P_n$ and each of them is given a pair of public key and secret key as: the dealer first randomly selects n many distinct elements $a_1, \ldots, a_n \in \mathbb{F}_p^*$ and consider the following polynomial,

$$f(x) = \prod_{i=1}^{n}(x + a_i)$$

  Then the ith participant P_i is given public key and secret key as

$$(pk_i, sk_i) = (a_i, g^{\frac{1}{\gamma + a_i}})$$

  Thus $\{a_i\}$'s are known to all, i.e., f is public. The Remark 1 below describes how the participants can verify the correctness of their respective secret keys. Also the dealer can publicly send the encrypted secret keys using any standard ElGamal like public key encryption scheme.

– **Distribution:** The dealer wishes to share a secret, which is an element in $\tilde{G}$. The secret is of the form $e(g, h)^{\alpha k}$, where k is selected randomly from $\mathbb{F}_p^*$. The dealer then computes and publishes the following values:

- Share commitment element (SCE): This value binds the dealer's commitment to the secret and is given as,
$$\text{SCE} = u^{-k} = g^{-k\alpha\gamma}$$
- Share deriving element (SDE): This value contains information about all the shares of the secret for which the dealer rendered his commitment. Participants will get their share by using the respective secret keys with SDE. This value is given as,
$$\text{SDE} = h^{\alpha k f(\gamma)}$$
The ith participant gets his share S_i by computing,
$$S_i = e(g^{\frac{1}{\gamma+a_i}}, \text{SDE})$$

– **Verification:** Any (external) verifier first computes and checks the following equality,
$$e(\phi(h^\alpha), h^{\sum_{i=0}^{n-t-1}\gamma^i}) = e(\phi(h), h^{\sum_{i=0}^{n-t-1}\alpha\gamma^i})$$
Then it computes,
$$\text{SCE}' = u^{-1} \text{ and } \text{SDE}' = h^{\alpha f(\gamma)}$$
One may note that $h^{\alpha f(\gamma)}$ can be computed using $\{h^{\alpha\gamma^i}\}_{i=1}^n$. The verifier then check the correctness by checking
$$e(\text{SCE}', \text{SDE}) = e(\text{SCE}, \text{SDE}')$$
One should also note that the share deriving element SDE is consistent with the share commitment element SCE if and only if there exists a scalar k such that $\text{SCE} = (\text{SCE}')^k$ and $\text{SDE} = (\text{SDE}')^k$. If the verification fails, all participants exit the protocol.

– **Reconstruction:** Let A be a qualified set of participants, i.e. it consists of at least t many participants. Let the public-keys of the participants are $a_{r_1}, \ldots, a_{r_s}$, $s \geq t$. Together with their respective shares $e(g^{\frac{1}{\gamma+a_{r_i}}}, h^{\alpha k f(\gamma)})$'s, they reconstruct the secret as follows. They first compute
$$R_1 = e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)}, \text{ where } f_{r_1,\ldots,r_s}(\gamma) = \frac{f(\gamma)}{\prod_{i=1}^{s}(\gamma+a_{r_i})}$$
[The computation of R_1 is done recursively. A simple case is described here for convenience. With $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_1}}}$ and $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}$, the element $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})}}$ is derived as:
$$\left(\frac{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_1}}}}{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}}\right)^{\frac{1}{(a_{r_2}-a_{r_1})}}$$
Thus, in order to compute $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})(\gamma+a_{r_3})}}$, one can repeat the above technique twice: first with the inputs $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_2}}}$ and $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma+a_{r_3}}}$ (which will output $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_2})(\gamma+a_{r_3})}}$) and secondly with the inputs $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_1})(\gamma+a_{r_2})}}$ and $e(g,h^{\alpha k f(\gamma)})^{\frac{1}{(\gamma+a_{r_2})(\gamma+a_{r_3})}}$] Next they compute,

$$R_2 = h^{\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))}$$

The computation of R_2 can successfully be carried out using $\{h^{\gamma^i}\}_{i=1}^{n-t-1}$ as degree of $\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0)) = n - s - 1$ and $n - s - 1 \leq n - t - 1$ as $(t \leq s)$. Now compute

$$\begin{aligned} e(\mathrm{SCE}, R_2) \cdot R_1 &= e(g^{-k\alpha\gamma}, h^{\frac{1}{\gamma}(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))}) \cdot e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)} \\ &= e(g,h)^{-k\alpha(f_{r_1,\ldots,r_s}(\gamma) - f_{r_1,\ldots,r_s}(0))} \cdot e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)} \\ &= e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(0)} \end{aligned}$$

Finally the secret is reconstructed by computing

$$e(g,h)^{k\alpha} = \left(e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(0)}\right)^{\frac{1}{f_{r_1,\ldots,r_s}(0)}}$$

Remark 1. The ith participant P_i, with its pair of keys

$$(\mathrm{pk}_i, \mathrm{sk}_i) = (a_i, g^{\frac{1}{\gamma + a_i}}),$$

can check the correctness of its secret key as follows. It first computes $h^{\alpha\gamma a_i}$ and checks if

$$\begin{aligned} e(\mathrm{sk}_i, h^{\alpha\gamma^2} \cdot h^{\alpha\gamma a_i}) &= e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha\gamma^2} \cdot h^{\alpha\gamma a_i}) \\ &= e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha\gamma(\gamma + a_i)}) \\ &= e(g^{\alpha\gamma}, h) = e(u, h) \end{aligned}$$

Remark 2. Our scheme couldn't provide a satisfactory answer to the following problem. During the reconstruction phase when a shareholder releases his **share commitment value** $e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha k f(\gamma)})$, there seems to be no obvious method to verify this value. One, not so interesting, wayout is to publish the hash digests of $e(g^{\frac{1}{\gamma + a_i}}, h^{\alpha k f(\gamma)})$'s, $(1 \leq i \leq n)$ during the distribution step of the scheme. But then this would mean that the correctness of the verification can only be carried out in the random oracle model.

Remark 3. The **Reconstruction** algorithm of the scheme requires the computation of $R_1 = e(g,h)^{k\alpha f_{r_1,\ldots,r_s}(\gamma)}$ given $\{a_{r_i}\}_{i=1}^{s}$ and $\{e(g^{\frac{1}{\gamma + a_{r_i}}}, h^{\alpha k f(\gamma)})\}_{i=1}^{s}$. The recursive method (described in the scheme) takes time that is bounded by $\frac{(s-1)s}{2} \cdot (T_p + T_{\tilde{G}})$, where T_p is the total time of a subtraction and an inversion in $\mathbb{F}_p$ and $T_{\tilde{G}}$ the total time of a division and exponentiation in $\tilde{G}$. One may note that the computation of the elements $\left(\frac{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_i}}}}{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_j}}}}\right)^{\frac{1}{(a_{r_j} - a_{r_i})}}$'s is done by exponentiation (not by computing the high order roots) as $(a_{r_j} - a_{r_i})$ is invertible modulo the order of the elements $\left(\frac{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_i}}}}{e(g,h^{\alpha k f(\gamma)})^{\frac{1}{\gamma + a_{r_j}}}}\right)$ which is p.

3.1 Security Analysis

In this Section we present the security analysis of our scheme.

3.2 Verifiability

We describe that a dishonest dealer cannot cheat the shareholders without being detected in the verification.

Lemma 1. *If the dealer passes the verification step, then all qualified subsets of honest shareholders will reconstruct the same secret that dealer had wished to share.*

The dealer puts forward its commitment to the secret $e(g,h)^{\alpha k}$ by binding its essential value k as part of the secret commitment element $\mathrm{SCE} = u^{-k} = g^{-k\alpha\gamma}$. Thus, the consistency of the pubilc parameters ($u = g^{\alpha\gamma}, h, h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}$, $\{h^{\alpha\gamma^i}\}_{i=1}^{n}$) and of the share deriving element SDE=$h^{k\alpha f(\gamma)}$ with the secret commitment element SCE follows from the facts that,

- $e(\phi(h^{\alpha}), h^{\sum_{i=0}^{n-t-1}\gamma^i}) = e(\phi(h), h^{\sum_{i=0}^{n-t-1}\alpha\gamma^i})$,
- SDE' ($= h^{\alpha f(\gamma)}$) and SCE' are obtained respectively from the dealer's publicly committed values $\{h^{\alpha\gamma^i}\}_{i=1}^{n}$ and $u = g^{\alpha\gamma}$,
- and the equality $e(\mathrm{SCE}', \mathrm{SDE}) = e(\mathrm{SCE}, \mathrm{SDE}')$ which essentially ensures that the scalar k is same such that $\mathrm{SCE} = (\mathrm{SCE}')^k$ and $\mathrm{SDE} = (\mathrm{SDE}')^k$.

3.3 Indistinguishability of Secrets (IND)

In this section, we show that our (t,n)-threshold PVSS scheme is SA-IND secure, assuming that the (n,t)-MSE-DDH problem is hard to solve.

Theorem 1. *Let n, t ($t \le n$) be two positive integers. For any PPT adversary $\mathcal{A}$ against the SA-IND security of our (t,n)-threshold PVSS scheme, there exists an algorithm $\mathcal{B}$ that distinguishes the two distributions of the (n,t)-MSE-DDH problem, such that*

$$\mathrm{Adv}_{\mathcal{A}}^{SA-IND}(\lambda) \le 2 \cdot \mathrm{Adv}_{\mathcal{B}}^{(n,t)-MSE-DDH}(\lambda)$$

Proof: The security reduction is to show that if there is an adversary ($\mathcal{A}$) which can break our (t,n)-threshold PVSS then one obtains an algorithm to solve (n,t)-MDE-DDH. The heart of such an algorithm is a simulator ($\mathcal{B}$) which is constructed as follows. Given an instance of (n,t)-MSE-DDH as input, the simulator plays the security game SA-IND with an adversary against (t,n)-threshold PVSS. The simulator sets up the (t,n)-threshold PVSS based on the (n,t)-MSE-DDH instance. The simulator gives the public parameters to the adversary and continues the game by answering all queries made by the adversary. The queries include, public keys of n participants and private keys of $t-1$ corrupted participants. In the process, it randomly chooses a bit b and distributes the shares of the secret T_b using the (n,t)-MSE-DDH instance provided as input. Finally, the adversary outputs a bit b'. Based on the value of b and b', the simulator decides whether the instance it received is **real** or **random**. Intuitively, if the adversary has an advantage in breaking the scheme, the simulator also has an advantage in distinguishing between **real** and **random** instances. This leads to an upper bound on the advantage of the adversary in terms of the advantage of the simulator in solving (n,t)-MSE-DDH.

– (n,t)**-MSE-DDH Instance:** The simulator, $\mathcal{B}$, receives an instance of (n,t)-MSE-DDH as described in Section 2.4. Thus $\mathcal{B}$ is given a bilinear map group system $(p, G_1, G_2, \tilde{G}, e(\cdot,\cdot))$ where the size of p is λ. We assume that p is significantly larger than n. $\mathcal{B}$ is further given polynomials $\theta_1, \theta_2 \in \mathbb{F}_p[x]$ of degrees n and $n-t+1$ respectively as described in Section 2.4. The negative roots of θ_1 and θ_2 are denoted by $a_1, \ldots, a_n$ and $a_{n+t}, \ldots, a_{2n}$ respectively. This instance also includes,
  - $\hat{g_1} := \ [g_1, g_1^{\alpha}, \{g_1^{\gamma^i}\}_{i=1}^{n+t-2}, \{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t}$ and $g_1^{k\gamma\theta_1(\gamma)}]$,
  - $\hat{g_2} := \ [g_2, g_2^{\alpha}, \{g_2^{\gamma^i}\}_{i=1}^{n-t-1}, \{g_2^{\alpha\gamma^i}\}_{i=1}^{n}$ and $g_2^{k\theta_2(\gamma)}]$,
  - an element $T \in \tilde{G}$.

– **Initialization:** $\mathcal{B}$ selects randomly $t-1$ elements $a_{n+1}, \ldots, a_{n+t-1} \in \mathbb{F}_p^*$ (different from the input a_i's) and construct a polynomial of degree $t-1$ as,

$$\theta_0(x) = \textstyle\prod_{i=n+1}^{n+t-1}(x + a_i)$$

The public parameters are defined and published in the following manner.
  - $g = g_1^{\theta_1(\gamma)\theta_0(\gamma)}$,
  - $h = g_2$,
  - $h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}$,
  - $u = g_1^{\alpha\gamma\theta_1(\gamma)\theta_0(\gamma)} = (g_1^{\theta_1(\gamma)\theta_0(\gamma)})^{\alpha\gamma} = g^{\alpha\gamma}$.

One may note that $\mathcal{B}$ cannot compute g, as degree of $\theta_1(x)\theta_0(x)$ is $n+t-1$. As we see subsequently that the form of g is required only for the security analysis and $\mathcal{B}$ doesn't have to publish it. Of course $\mathcal{B}$ can compute u with $\{g_1^{\alpha\gamma^i}\}_{i=1}^{n+t}$. Thus to be precise the published parameters are,

$$u, h, h^{\alpha}, \{h^{\gamma^i}\}_{i=1}^{n-t-1}, \{h^{\alpha\gamma^i}\}_{i=1}^{n}$$

– **User Keys Generation:** There are n participants and $t-1$ of them are assumed to be corrupted, i.e., $\mathcal{B}$ will issue respective public key and secret key pairs to $\mathcal{A}$ for $t-1$ corrupted participants and only public keys for the remaining $n-t+1$ participants.
  - Corrupted participants: For $t-1$ corrupted participants $(P_{w_{n+1}}, \ldots, P_{w_{n+t-1}})$, the key pairs are issued to $\mathcal{A}$ as
  $$(\mathrm{pk}_{w_i}, \mathrm{sk}_{w_i}) = (a_i, g_1^{\frac{\theta_1(\gamma)\theta_0(\gamma)}{\gamma + a_i}}) = (a_i, g^{\frac{1}{\gamma+a_i}}),\ i = n+1 \text{ to } n+t-1$$
  - Honest Participants: The remaining $n-t+1$ participants assigned their respective public keys from
  $$\{a_i\}_{i=n+t}^{2n}$$

One may note that $\mathcal{B}$ can compute $g_1^{\frac{\theta_1(\gamma)\theta_0(\gamma)}{\gamma+a_i}}$'s using $\{g_1^{\gamma^i}\}_{i=1}^{n+t-2}$.

- **Distribution of secret commitment element and share deriving element:** $\mathcal{B}$ defines polynomial f, as described in the scheme, whose negative roots correspond to the public keys of all the participants. Thus,

$$f(x) = \theta_0(x)\theta_2(x)$$

$\mathcal{B}$ then proceed to select T as the **secret** that it intends to share among the n participants by publishing the secret commitment element and share deriving element as,

$$(\text{SCE}, \text{SDE}) = (g_1^{k\alpha\theta_1(\gamma)}, g_2^{k\theta_2(\gamma)})$$

One may note that, if we set $k' = \frac{k}{\alpha\theta_0(\gamma)}$, then

$$\begin{aligned}\text{SCE} &= g_1^{-k\gamma\theta_1(\gamma)} \\ &= g_1^{-k'\alpha\gamma\theta_0(\gamma)\theta_1(\gamma)} \\ &= (g_1^{\theta_1(\gamma)\theta_0(\gamma)})^{-k'\alpha\gamma} \\ &= (g)^{-k'\alpha\gamma} = u^{-k'}\end{aligned}$$

and

$$\begin{aligned}\text{SDE} &= g_2^{k\theta_2(\gamma)} \\ &= g_2^{\alpha k'\theta_0(\gamma)\theta_2(\gamma)} \\ &= h^{\alpha k' f(\gamma)}\end{aligned}$$

Further, if T is **real**, then

$$\begin{aligned}T &= e(g_1, g_2)^{k\theta_1(\gamma)} \\ &= e(g_1, g_2)^{\alpha k'\theta_0(\gamma)\theta_1(\gamma)} \\ &= e(g, h)^{\alpha k'}\end{aligned}$$

Thus the secret T is of required form as described in the scheme. With this, the simulator now randomly selects a bit $b \in \{0, 1\}$ and sets $T_b = T$ and assigns a random value in the secret space $\tilde{G}$ to T_{1-b}. $\mathcal{A}$ is then issued

$$(\text{SCE}, \text{SDE}, T_0, T_1)$$

- **Guess:** Finally $\mathcal{A}$ outputs its guess, a bit b' for b.

Based on the value of b and b', $\mathcal{B}$ goes on to solve the (n, t)-MSE-DDH problem instance at hand as follows:

- if $b' = b$, $\mathcal{B}$ answers 1, meaning that $T = e(g_1, g_2)^{k\theta_1(\gamma)}$,
- otherwise, $\mathcal{B}$ answers 0, meaning that T is a random element of $\tilde{G}$.

Thus, the advantage of the algorithm $\mathcal{B}$ in solving the input (n, t)-MSE-DDH problem is

$$\text{Adv}_{\mathcal{B}}^{(n,t)-\text{MSE-DDH}}(\lambda)$$
$$= \Big|\Pr[\mathcal{B}(\mathcal{I}(\theta_1, \theta_2, [g_1], [g_2], T)) = 1|\text{real}] - \Pr[\mathcal{B}(\mathcal{I}(\theta_1, \theta_2, [g_1], \hat{g_2}, T)) = 1|\text{random}]\Big|$$
$$= \Big|\Pr[b' = b|\text{real}] - \Pr[b' = b|\text{random}]\Big|.$$

In the above simulation, when the event **real** occurs, the simulator $\mathcal{B}$ poses as a real challenger for $\mathcal{A}$, i.e., the distribution of all the parameters during the simulation perfectly comply with the IND security game and therefore $\Big|\Pr[b' = b|\text{real}] - \frac{1}{2}\Big| = \frac{1}{2}\text{Adv}_{\mathcal{A}}^{SA-IND}(\lambda)$. Whereas, when the event **random** occurs, the distribution of the guess bit b' is completely independent of the distribution of the bit b and thus $\Pr[b' = b]$ is equal to $\frac{1}{2}$. Putting it altogether, we obtain

$$\text{Adv}_{\mathcal{A}}^{SA-IND}(\lambda) \leq 2 \cdot \text{Adv}_{\mathcal{B}}^{(n,t)-MSE-DDH}(\lambda)$$

4 Comparison

We use $\mathcal{HV}$ and $\mathcal{PS}$ to denote the schemes proposed in [18] and this article respectively. In Table 1, we compare $\mathcal{PS}$ with $\mathcal{HV}$ in terms of exponentiations (in the underlying groups) and pairing computations. $\mathcal{HV}$ uses *symmetric* pairing and $\mathcal{PS}$ is based on *asymmetric* pairing. Thus the group exponents are computed in G for $\mathcal{HV}$, and in G_1, G_2 for $\mathcal{PS}$ (see Section 3). A list of points to better understand the comparison table is given as follows:

- n, t bears the usual meaning.
- For Reconstruction algorithm, comparison is done based on the number of operations required for t shareholders to reconstruct the secret.
- $\mathcal{HV}$ requires computation of $2t$ pairings to verify the shares, released by the t shareholders during the Reconstruction algorithm. But this number has not been counted in the comparison table as $\mathcal{PS}$ does not satisfy this property.

Table 1. Comparison Table

Algorithms	Schemes	Exponentiation in G, G_1 or G_2	Exponentiation in $\tilde{G}$	Pairing
Setup	$\mathcal{HV}$	n	–	–
	$\mathcal{PS}$	$3n - t + 1$	–	–
Distribution	$\mathcal{HV}$	$n + t$	–	–
	$\mathcal{PS}$	2	–	n
Verification	$\mathcal{HV}$	$n \cdot t$	–	$2n$
	$\mathcal{PS}$	n	–	4
Reconstruction	$\mathcal{HV}$	$2t$	–	1
	$\mathcal{PS}$	$n - t - 1$	$\approx \frac{(t-1)t}{2}$	1

5 Conclusion

We have proposed in this paper a practical (t, n)-threshold PVSS scheme that achieves simultaneously:

– Efficient non-interactive public verification.
– Provable security for the public verifiability in the standard model.

Efficiency of the non-interactive public verification step of our scheme is optimal while comparing with the earlier proposal [18]. We provide formal security proof for our scheme, against indistinguishability of secrets (IND) attack model, based on the hardness of a problem that we call the (n, t)- multi-sequence of exponents Diffie-Hellman problem (MSE-DDH). This problem falls under the general Diffie-Hellman exponent problem framework. [5]. The security proof (for indistinguishability of secrets) could handle only static adversaries. An interesting task would be to modify the scheme accordingly so that an adaptive adversary can be handled during the security analysis. The other challenging task is to provide the security proof under a more standard assumption.

Acknowledgments

The author is thankful to Ayineedi Venkateswarlu, Rana Barua, and Mridul Nandi for many useful discussions.

References

1. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
3. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC, pp. 1–10 (1988)
4. Blakley, G.: Safeguarding cryptographic keys. AFIPS National Computer Conference 48, 313–317 (1979)
5. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
8. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
9. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC, pp. 11–19 (1988)
10. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: FOCS, pp. 383–395 (1985)

11. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 39–59. Springer, Heidelberg (2007)
12. Delerablée, C., Pointcheval, D.: Dynamic threshold public-key encryption. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 317–334. Springer, Heidelberg (2008)
13. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: Annual IEEE Symposium on Foundations of Computer Science, vol. 0, pp. 427–438 (1987)
14. 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)
15. 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)
16. Galbraith, S.D., Harrison, K., Soldera, D.: Implementing the tate pairing. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 324–337. Springer, Heidelberg (2002)
17. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
18. Heidarvand, S., Villar, J.L.: Public verifiability from pairings in secret sharing schemes. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 294–308. Springer, Heidelberg (2009)
19. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010)
20. Micali, S.: Fair public-key cryptosystems. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 113–138. Springer, Heidelberg (1993)
21. Miyaji, A., Nakabayashi, M., Takano, S.: Characterization of Elliptic Curve Traces under FR-Reduction. In: Won, D. (ed.) ICISC 2000. LNCS, vol. 2015, pp. 90–108. Springer, Heidelberg (2001)
22. Miyaji, A., Nakabayashi, M., Takano, S.: New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. IEICE Transactions on Fundamentals E84-A(5), 1234–1243 (2001)
23. 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)
24. Ruiz, A., Villar, J.L.: Publicly verfiable secret sharing from paillier's cryptosystem. In: WEWoRC, pp. 98–108 (2005)
25. 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)
26. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
27. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
28. Simmons, G.J.: How to (Really) share a secret. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 390–448. Springer, Heidelberg (1990)
29. Stadler, M.: Publicly verifiable secret sharing. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 190–199. Springer, Heidelberg (1996)
30. Stinson, D.R.: An explication of secret sharing schemes. Des. Codes Cryptography 2(4), 357–390 (1992)