IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

How to construct a verifiable multi-secret sharing scheme based on graded encoding schemes

ISSN 1751-8709
Received on 14th June 2018
Revised 25th November 2018
Accepted on 1st February 2019
E-First on 28th February 2019
doi: 10.1049/iet-ifs.2018.5306
www.ietdl.org

Massoud Hadian Dehkordi[1], Hossein Oraei[1] ✉

[1]*Cryptography and Data Security Laboratory, School of Mathematics, Iran University of Science & Technology, Narmak, Tehran 1684613114, Iran*

✉ *E-mail: hossein_oraei@mathdep.iust.ac.ir*

Abstract: In a verifiable multi-secret sharing scheme, a dealer distributes multiple secrets between a group of participants and also additional information is given that allows each participant to check whether his share is valid. In this study, the authors present a novel verifiable multi-secret sharing (VMSS) scheme with general access structure using monotone span programs, which its security is based on graded encoding schemes. More precisely, they reduce the hardness of graded decision-Diffie–Hellman problem to the computational security of the authors' scheme in the standard model. To the best of the authors' knowledge, this is the first study to present a VMSS scheme based on graded encoding schemes.

1 Introduction

In a multi-secret sharing (MSS) scheme, several secrets are distributed among a group of participants by a dealer D such that only authorised sets can reconstruct the secrets by combining their shares (or their pseudo shares), while other subsets cannot know any information about them [1]. For sharing the secrets, there are many techniques such as multilinear maps [2], polynomial interpolation [3, 4], using the Chinese remainder theorem [5, 6], monotone span program (MSP) [7, 8] and so on. The notion of MSP which will be described later, was introduced by Karchmer and Wigdersonm [9]. In an MSS scheme based on MSP, the secrets are taken from a finite field, and each participant's share is obtained by computing a linear combination of some random numbers and the secrets [10, 11]. In the reconstruction phase, all participants of an authorised set compute a linear combination of their shares.

1.1 Background

The notion of multilinear maps was introduced by Boneh and Silverberg [12] as an extension of bilinear maps and provided two applications: a one-round multi-party key exchange protocol and a broadcast encryption scheme. Different from bilinear maps, which can be built from the pairing of elliptic curves, constructing multilinear maps was a long-standing open problem. This problem was eventually solved by Garg *et al.* [13], who constructed the first approximate construction of multilinear maps. They introduced the notion of graded encoding scheme as a variant of multilinear maps and proposed a candidate construction by using ideal lattices which is called GGH13.

Graded encoding schemes are useful tools, enabling many important applications, such as functional encryption [14, 15], obfuscation [16, 17], witness encryption [18–20], multipartite key exchange [13] and so on.

1.2 Related work

So far, several provably secure secret sharing schemes for different purposes have been proposed by many researchers [21–29]. Among them, there are verifiable secret sharing (VSS) schemes such as [22, 24–26, 29]. In 2009, Heidarvand and Villar [22] proposed the notion of indistinguishability (IND) of secrets and also two new secure definitions for (Publicly) VSS schemes, that capture the notion of IND of secrets in the standard model (IND-secrecy and CSA-secrecy). In the first definition (IND-secrecy), an adversary cannot distinguish between a random value and the shared secret, while in the second definition (CSA-secrecy) secrets of the challenge are chosen by the adversary. In the random oracle model, Wu and Tseng [24] proposed a secure (Publicly) VSS scheme. They also presented a realisation of the (Publicly) verifiable property to propose their (Publicly) VMSS scheme from bilinear pairings in [26]. Afterward, using decisional bilinear Diffie–Hellman (DBDH) assumption, a (t, n) threshold (Publicly) VSS scheme was proposed which has CSA-secrecy [25]. They used one-way hash functions to prove the indistinguishability of their scheme against adaptively chosen secret attacks in the standard model. The use of bilinear maps for constructing verifiable secret sharing schemes is a common method used in other papers [30]. It should be noted that there are other methods for constructing verifiable secret sharing schemes, such as lattices [29].

By multiple linear pairs, two (Publicly) VSS schemes were proposed by Peng and Tian [2, 31]. These two schemes are based on multilinear maps and not on graded encoding schemes.

1.3 Motivation

The VSS schemes [22, 24–26] are based on bilinear maps, which can be built from the pairing of elliptic curves. Also, there are VSS schemes such as [2, 31] which have been constructed using multilinear maps (the maps which do not exist). On the other hand, all of the above-mentioned schemes, have (t, n) threshold access structures in which only any at least t participants can reconstruct the secret, while in general access structure the participants do not have the same ability to recover the secret.

However, there are many situations in which general access structures are required [10, 11, 32, 33]. Unfortunately, in these MSP based schemes:

- There not exists computational verification property.
- When one of the secrets is reconstructed, the shares are not reusable.
- There not exists any formal proof.

1.4 Contribution

To the best of our knowledge, currently, there are no studies that indicate how a secret sharing scheme can be built from the realisation of multilinear maps (i.e. graded encoding schemes). In this paper, we use monotone span programs to construct a VMSS

scheme based on graded encoding schemes. In summary, our new proposed scheme has the following properties:

- It has the computational verification property.
- It supports the general access structure which makes it more applicable than threshold VMSS schemes.
- The share of each participant is reusable when secrets are reconstructed.

The most important part of this work is the formal security analysis that we provide for our VMSS scheme by giving the exact relation between the GDDH problem and the security of our scheme. We present a security experiment to prove that our VMSS scheme enjoys the IND-secrecy.

1.5 Organisation

The paper is organised as follows. Section 2 describes the required tools. In Section 3, we give formal definitions of computational security of a VMSS scheme. In Section 4, we propose a secret sharing scheme based on graded encoding schemes and its security is discussed in Section 5. In Section 6, we show how the first scheme can become a VMSS scheme and finally in Section 7, we conclude the paper.

2 Preliminaries

Here, we summarise some preliminary concepts about secret sharing schemes, multilinear maps and also graded encoding schemes.

2.1 Secret sharing schemes

Definition 1: Suppose that $\mathscr{P} = \{P_1, \ldots, P_n\}$ be a set of participants. Then, a monotone access structure Γ on $\mathscr{P}$ is a set of non-empty subsets of participants which satisfies the monotone ascending property

$$(A \in \Gamma, A \subseteq A' \subseteq \mathscr{P}) \Rightarrow A' \in \Gamma.$$

The sets in Γ and $\mathscr{A} = 2^{\mathscr{P}} \setminus \Gamma$ are called authorised sets and unauthorised sets, respectively, where $2^{\mathscr{P}}$ denotes the power set of $\mathscr{P}$. The set $\mathscr{A}$ that we call adversary structure satisfies the monotone descending property

$$(B \in \mathscr{A}, B' \subseteq B \subseteq \mathscr{P}) \Rightarrow B' \in \mathscr{A}.$$

We denote by $\Gamma_{\min}$ the minimal elements in Γ and by $\mathscr{A}_{\max}$ the maximal elements in $\mathscr{A}$. That is

$$\Gamma_{\min} = \{A \in \Gamma \,|\, \forall A' \subsetneq A \Rightarrow A' \notin \Gamma\},$$

and

$$\mathscr{A}_{\max} = \{B \in \mathscr{A} \,|\, \forall B \subsetneq B' \Rightarrow B' \notin \mathscr{A}\}.$$

Definition 2: For any adversary structure $\mathscr{A}$ over $\mathscr{P}$, its dual is defined as

$$\widetilde{\mathscr{A}} = \{B \subseteq \mathscr{P} \,|\, B^c \notin \mathscr{A}\}.$$

2.1.1 Monotone span program and LMSS schemes: In 1993, Wigdersonm and Karchmer introduced the concept of MSP which is a model of computation as follows [9]:

Definition 3: A MSP over a set $\mathscr{P}$ is a triple $(\mathbb{F}, \boldsymbol{M}, \psi)$, in which $\mathbb{F}$ is a finite field, $\boldsymbol{M}$ is a $d \times l$ distribution matrix with entries in $\mathbb{F}$ and $\psi: \{1, 2, \ldots, d\} \to \mathscr{P}$ is a function.

In the above definition, ψ is a surjective function that distributes to each participant of $\mathscr{P} = \{P_1, \ldots, P_n\}$ some rows of $\boldsymbol{M}$.

Definition 4: Suppose that Γ be an access structure for which $\mathscr{A} \subseteq \widetilde{\mathscr{A}}$. A monotone span program $(\mathbb{F}, \boldsymbol{M}, \psi)$ is called an MSP for Γ concerning a target vector $\boldsymbol{\xi} \in \mathbb{F}^l \setminus \{(0, \ldots, 0)\}$, if for all $A \subseteq \{P_1, \ldots, P_n\}$ the following conditions are satisfied:

- if $A \in \Gamma$, then $\boldsymbol{\xi} \in span\{\boldsymbol{M}_A\}$.
- if $A \in \mathscr{A}$, then there exists a sweeping vector $\boldsymbol{k} = (k_1, k_2, \ldots, k_l)^T \in \mathbb{F}^l$ such that $\boldsymbol{M}_A \boldsymbol{k} = \boldsymbol{0} \in \mathbb{F}^{n_1}$ with $k_1 = 1$.

where $\boldsymbol{M}_A$ is the matrix $\boldsymbol{M}$ restricted to the rows i with $\psi(i) \in A$, with the notation $\boldsymbol{\xi} \in span\{\boldsymbol{M}_A\}$ we mean that there is a vector $\boldsymbol{w}_A \in \mathbb{F}^{n_1}$ for which $\boldsymbol{\xi} = \boldsymbol{w}_A \boldsymbol{M}_A$ and n_1 is the number of participants in A.

Similar to the case of one target vector, we say that $(\mathbb{F}, \boldsymbol{M}, \psi)$ is an MSP for access structures Γ_j, $1 \le j \le m$ concerning some target vectors $\boldsymbol{\xi}_j \in \mathbb{F}^l \setminus \{(0, \ldots, 0)\}$, if it is true that for each $1 \le j \le m$, $\boldsymbol{\xi}_j \in span\{\boldsymbol{M}_A\}$ if $A \in \Gamma_j$, where $\boldsymbol{\xi}_j \in span\{\boldsymbol{M}_A\}$ means that there is $\boldsymbol{w}_{Aj}$ for which $\boldsymbol{\xi}_j = \boldsymbol{w}_{Aj} \boldsymbol{M}_A$.

It is proved that constructing an MSP $(\mathbb{F}, \boldsymbol{M}, \psi)$ for access structures Γ_j is equivalent to devising a linear multi-secret sharing (LMSS) scheme for Γ_j, $1 \le j \le m$ [34]. Also, $(\mathbb{F}, \boldsymbol{M}, \psi)$ is an MSP for access structures Γ_j, $1 \le j \le m$ if there exists a (target) vector $\boldsymbol{\xi}_j \in \bigcap_{A \in (\Gamma_j)_{\min}} \sum_{\psi(i) \in A} V_i \setminus \bigcup_{B \in (\mathscr{A}_j)_{\max}} \sum_{\psi(i) \in B} V_i$, in which V_i is the space spanned by the row vectors of $\boldsymbol{M}$ distributed to the player $\psi(i)$ and $\boldsymbol{\xi}_j$ can be considered as the above target vectors.

According to the above discussion, we consider the target vector $\boldsymbol{\xi}_j$ to be an l-rowed vector whose jth component is 1 and other components are 0, $1 \le j \le m$. Now, we describe how an LMSS scheme which realises access structure Γ_j, $1 \le j \le m$ can be constructed using any MSP $(\mathbb{F}, \boldsymbol{M}, \psi)$:

- *Distribution step:* Suppose the dealer D has secrets $s_1, s_2, \ldots, s_m$. Then, he can construct a distribution vector $\boldsymbol{\rho} = (s_1, s_2, \ldots, s_m, \rho_{m+1}, \ldots, \rho_l)^T$ in which $\rho_{m+1}, \ldots, \rho_l$ are random elements in $\mathbb{F}$. Next, he computes $\boldsymbol{z} = M\boldsymbol{\rho} = (z_1, z_2, \ldots, z_d)^T$ and gives z_i to the participant $P_{\psi(i)}$.
- *Reconstruction step:* In the following, suppose that the notation $\boldsymbol{z}_A$ be the vector $\boldsymbol{z}$ restricted to the indices in A. For each authorised set $A \in \Gamma_j$, there is a vector $\boldsymbol{w}_{Aj}$ for which $\boldsymbol{\xi}_j = \boldsymbol{w}_{Aj} \boldsymbol{M}_A$. So

$$\boldsymbol{w}_{Aj} \cdot \boldsymbol{z}_A = \boldsymbol{w}_{Aj} \cdot (\boldsymbol{M}_A \boldsymbol{\rho}) = (\boldsymbol{w}_{Aj} \cdot \boldsymbol{M}_A)\boldsymbol{\rho} = \boldsymbol{\xi}_j \cdot \boldsymbol{\rho} = s_j$$

  i.e. the secret s_j can be reconstructed by computing a linear computation of the shares of participants in A.

2.2 Multilinear maps

The notion of multilinear maps is defined by Boneh and Silverberg as follows [12]. For cyclic groups $G_1, \ldots, G_k$ and G_T of the same prime order q, a k-multilinear map $e: G_1 \times \cdots \times G_k \to G_T$ is a map such that

(i) *Multilinear:* For all $g_1 \in G_1, \ldots, g_k \in G_k$ and $a_1, \ldots, a_k \in \mathbb{Z}_q^*$, we have $e(g_1^{a_1}, \ldots, g_k^{a_k}) = e(g_1, \ldots, g_k)^{a_1 \cdots a_k}$.
(ii) *Non-degenerate:* If for $1 \le i \le k$, $g_i \in G_i$ be a generator of the group G_i, then $e(g_1, \ldots, g_k)$ is a generator of G_T.
(iii) *Computable:* For all $g_1 \in G_1, \ldots, g_k \in G_k$, the value $e(g_1, \ldots, g_k)$ is computed efficiently.

2.2.1 Multilinear DDH (MDDH) problem: Here for each multilinear map e, we propose a game between $\mathscr{B}$ as an adversary and $\mathscr{C}$ as a challenger which is called multilinear decision-Diffie–Hellman problem.

Game $\mathscr{G}_1$:

(i) Using the security parameter λ, the challenger $\mathscr{C}$ firstly chooses the groups G and G_T of prime order q, a generator g of G and a symmetric k-multilinear map $e: G \times \cdots \times G \to G_T$.
(ii) Using the integers $\alpha_0, \alpha_1, \ldots, \alpha_k \in \mathbb{Z}_q^*$ which are chosen uniformly at random, the challenger $\mathscr{C}$ computes $D_0 = e(g, \ldots, g)^{\prod_{i=0}^{k} \alpha_i}$.
(iii) Using the integer $\alpha \in \mathbb{Z}_q^*$ which is chosen uniformly at random, the challenger $\mathscr{C}$ computes $D_1 = e(g, \ldots, g)^{\alpha}$.
(iv) $\mathscr{C}$ randomly selects a bit $\beta \in \{0, 1\}$ and sends $(g, g^{\alpha_0}, g^{\alpha_1}, \ldots, g^{\alpha_k}, D_\beta)$ to the adversary $\mathscr{B}$.
(v) Finally, $\mathscr{B}$ publishes a bit $\beta' \in \{0, 1\}$.

In this game, we define the advantage of the adversary $\mathscr{B}$ in solving MDDH problem to be

$$\mathtt{Adv}_{\mathscr{B}}^{\mathrm{MDDH}}(\lambda) = \left| \mathtt{Prob}[\beta' = \beta] - \frac{1}{2} \right|.$$

We say that the MDDH problem is hard, if for each polynomial time adversary $\mathscr{B}$, $\mathtt{Adv}_{\mathscr{B}}^{\mathrm{MDDH}}(\lambda)$ is a negligible function of λ.

2.3 Graded encoding schemes

Garg *et al.* [13] defined the notion of k-graded encoding schemes as an approximation of k-multilinear maps as follows:

Definition 5: (k-graded encoding scheme): Let R be a ring and $\mathcal{S} = \{S_i^{(\alpha)} \subset \{0,1\}^* \,|\, 0 \le i \le k, \alpha \in R\}$ be a family of sets such that for each constant index i, the sets $\{S_i^{(\alpha)} \,|\, \alpha \in R\}$ are disjoint. Then a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ with the ring R and the family of sets $\mathcal{S}$ consists of the following procedures:

- $\mathrm{InstGen}(1^\lambda, k)$: The randomised instance-generation procedure takes as input a security parameter λ and also multilinearity parameter k. It outputs $(\mathtt{params}, P_{zt})$, where P_{zt} is a zero-test parameter (as below) and $\mathtt{params}$ is a description of a k-graded encoding scheme.
- $\mathtt{Samp}(\mathtt{params})$ The randomised ring sampler procedure outputs $a \in S_0^{(\alpha)}$ which is a 'level-zero encoding', where $\alpha \in R$ is a random and nearly uniform element.
- $\mathtt{Enc}(\mathtt{params}, i, a)$: The (possibly randomised) encoding procedure takes as input an index $i \le k$ and a 'level-zero' encoding $a \in S_0^{(\alpha)}$ and outputs $u \in S_i^{(\alpha)}$ which is a 'level-i' encoding for the same $\alpha \in R$.
- $\mathtt{Add}(\mathtt{params}, i, u_1, u_2), \mathtt{Neg}(\mathtt{params}, i, u_1)$: On input of $\mathtt{params}$, an index $i \le k$ and two level-i encodings $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, the addition and negation procedures compute $\mathtt{Add}(\mathtt{params}, i, u_1, u_2) = u_1 + u_2 \in S_i^{(\alpha_1 + \alpha_2)}$ and $\mathtt{Neg}(\mathtt{params}, i, u_1) = -u_1 \in S_i^{(-\alpha_1)}$, respectively. Here, $-\alpha_1$ and $\alpha_1 + \alpha_2$ are negation and addition in the ring R.

This implies that for a collection of h encodings $u_j \in S_i^{(\alpha_j)}$, where $j = 1, \ldots, h$, we get $u_1 + \cdots + u_h \in S_i^{(\alpha_1 + \cdots + \alpha_h)}$.

- $\mathtt{Mul}(\mathtt{params}, i_1, u_1, i_2, u_2)$: On input of $\mathtt{params}$, two indices i_1, i_2 with $i_1 + i_2 \le k$, a level-i_1 encoding $u_1 \in S_{i_1}^{(\alpha_1)}$ and a level-i_2 encoding $u_2 \in S_{i_2}^{(\alpha_2)}$, the multiplication procedure computes $\mathtt{Mul}(\mathtt{params}, i_1, u_1, i_2, u_2) = u_1 \times u_2 \in S_{i_1 + i_2}^{(\alpha_1 \cdot \alpha_2)}$, where $i_1 + i_2$ is integer addition and $\alpha_1 \cdot \alpha_2$ is multiplication in the ring R.

This implies that for a collection of h encodings $u_j \in S_{i_j}^{(\alpha_j)}$ where $j = 1, \ldots, h$, we get $u_1 \times \cdots \times u_h \in S_{i_1 + \cdots + i_h}^{(\prod_{j=1}^{h} \alpha_j)}$ as long as $\sum_{j=1}^{h} i_j \le k$.

- $\mathtt{isZero}(\mathtt{params}, u)$: On input of $\mathtt{params}$ and u, the zero-test procedure outputs 1 if $u \in S_k^{(0)}$ and 0 otherwise. In other words, this procedure only outputs 1 if u be the level-k encoding of 0.
- $\mathtt{Ext}(\mathtt{params}, P_{zt}, u)$: On input of $\mathtt{params}$, the zero-test parameter P_{zt} and $u \in S_k^{(\alpha)}$, the extraction procedure outputs $s \in \{0,1\}^\lambda$ such that
  a. For every $\alpha \in R$ and two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$,
  $$\mathtt{Ext}(\mathtt{params}, P_{zt}, u_1) = \mathtt{Ext}(\mathtt{params}, P_{zt}, u_2).$$
  b. The distribution $\{\mathtt{Ext}(\mathtt{params}, P_{zt}, u) \,|\, u \in S_k^{(\alpha)}, \alpha \in R\}$ over $\{0,1\}^\lambda$ is nearly uniform.

We now give a more precise description of the above procedures.

The real-life version of efficient procedures. Garg *et al.* proposed GGH13 which is a k-graded encoding scheme over ideal lattices. Their realisation of the above procedures has two changes in the zero-test and extraction procedures as follows:

- *Zero-test:* This procedure sometimes allows false positives, but not false negatives. More precisely, for every $u \in S_k^{(0)}$ it is held that $\mathtt{isZero}(\mathtt{params}, u) = 1$, but for $\alpha \in R$ which is a nearly uniform random element
$$\Pr[\exists u \in S_k^{(\alpha)} \,|\, \mathtt{isZero}(\mathtt{params}, u) = 1] = \mathtt{negl}(\mathrm{l}).$$
- *Extraction:* According to the ring sampler and encoding procedures, to get a level-i encoding of a nearly uniform random element $\alpha \in R$ where $1 \le i \le k$, we first run the ring sampler procedure to get a level-0 encoding of α, and then run the encoding procedure to get a level-i encoding of α. On the other hand, in the extraction procedure of GGH13, there is a good probability of generating the same output for any two different level-k encodings of α.

Thus, the properties (a) and (b) of the extraction procedure are replaced by two weaker requirements:

(a′) For every $a \leftarrow \mathtt{Samp}(\mathtt{params})$, where $a \in S_0^{(\alpha)}$, if the (randomised) encoding procedure is run twice on a to obtain two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$, then

$$\Pr[\mathtt{Ext}(\mathtt{params}, P_{zt}, u_1) = \mathtt{Ext}(\mathtt{params}, P_{zt}, u_2)] \ge 1 - \mathtt{negl}(\mathrm{l}).$$

(b′) The following distribution over $\{0,1\}^\lambda$ is nearly uniform:

$$\{\mathtt{Ext}(\mathtt{params}, P_{zt}, u) \,|\, a \leftarrow \mathtt{Samp}(\mathtt{params}), u \leftarrow \mathtt{Enc}(\mathtt{params}, i, a)\}.$$

Remark 1: As explained in the extraction procedure, for every $\alpha \in R$ and two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$, it is held that $\mathtt{Ext}(\mathtt{params}, P_{zt}, u_1) = \mathtt{Ext}(\mathtt{params}, P_{zt}, u_2) \in \{0,1\}^\lambda$ (with high probability in the real-life version). In the remaining of this paper, for simplicity, we work with the dream version of extraction procedure in which this probability is one. If we want to use the real-life version, we must consider the negligible probability that $\mathtt{Ext}(\mathtt{params}, P_{zt}, u_1) \ne \mathtt{Ext}(\mathtt{params}, P_{zt}, u_2)$. One better way is to use the idea presented in [35] in which a graded encoding scheme is presented that realises the 'dream version' of GGH13. Thus, for every $\alpha \in R$, we write $\mathtt{Ext}(\mathtt{params}, P_{zt}, S_k^{(\alpha)})$ to denote this λ bit string.

Remark 2: A k-graded encoding scheme may be consist of some secret parameters (e.g. see the description of $\mathtt{GGH13}$ in [36]). Nevertheless, anyone can use a k-graded encoding scheme without knowing their secret parameters.

2.3.1 Graded DDH (GDDH) problem: The analogue of MDDH in k-graded encoding schemes is called 'Graded Decision-Diffie–Hellman' problem. Now, between $\mathscr{B}$ as an adversary and $\mathscr{C}$ as a challenger, we define the GDDH problem for each k-graded encoding scheme GES($R, \mathcal{S}$), which is a game as follows:

Game $\mathscr{G}_2$:

(i) Based on the multilinearity parameter k and the security parameter λ, the challenger $\mathscr{C}$ runs $(\mathtt{params}, P_{zt}) \leftarrow \mathtt{InstGen}(1^\lambda, k)$ to get a description of a k-graded encoding scheme.
(ii) For $j = 1, \ldots, k+1$, the challenger firstly runs $a_j \leftarrow \mathtt{Samp}(\mathtt{params})$ to get level-zero encodings $a_j \in S_0^{(\alpha_j)}$, where $\alpha_1, \ldots, \alpha_{k+1} \in R$ are random and nearly uniform elements. Then, $\mathscr{C}$ performs the following three steps:

- $\mathscr{C}$ runs $u_j \leftarrow \mathtt{Enc}(\mathtt{params}, 1, a_j)$ to get level-1 encodings $u_j \in S_1^{(\alpha_j)}$.
- According to the multiplication procedure, $\mathscr{C}$ computes $\tilde{a} = a_1 \times \cdots \times a_{k+1} \in S_0^{\left(\prod_{j=1}^{k+1}\alpha_j\right)}$
- $\mathscr{C}$ runs $\tilde{u} \leftarrow \mathtt{Enc}(\mathtt{params}, k, \tilde{a})$ to get the level-k encoding $\tilde{u} \in S_k^{\left(\prod_{j=1}^{k+1}\alpha_j\right)}$.

(iii) The challenger $\mathscr{C}$ runs $\hat{a} \leftarrow \mathtt{Samp}(\mathtt{params})$ to get $\hat{a} \in S_0^{(\alpha)}$ which is a level-zero encoding, where $\alpha \in R$ is a random and nearly uniform element. Then

- $\mathscr{C}$ runs $\hat{u} \leftarrow \mathtt{Enc}(\mathtt{params}, k, \hat{a})$ to get level- k encoding $\hat{u} \in S_k^{(\alpha)}$.

(iv) Consequently, the challenger $\mathscr{C}$ considers two distributions $D_0 = \{(\mathtt{params}, P_{zt}, \{u_j\}_{j=0}^{k+1}, \tilde{u})\}$ in which $\tilde{u}$ is the level-k encoding of the right product $\prod_{j=1}^{k+1}\alpha_j$, and $D_1 = \{(\mathtt{params}, P_{zt}, \{u_j\}_{j=0}^{k+1}, \hat{u})\}$ where $\hat{u}$ is the level-k encoding of a random element α. Then, $\mathscr{C}$ randomly generates a bit $\beta \in \{0,1\}$ and sends D_β to $\mathscr{B}$.
(v) Finally, $\mathscr{B}$ publishes a bit $\beta' \in \{0,1\}$.

In this game, we define the advantage of the adversary $\mathscr{B}$ in solving GDDH problem to be

$$\mathtt{Adv}_{\mathscr{B}}^{\mathrm{GDDH}}(\lambda) = \left|\mathtt{Prob}[\beta' = \beta] - \frac{1}{2}\right|.$$

We say that the GDDH problem is hard, if for each polynomial time adversary $\mathscr{B}$, $\mathtt{Adv}_{\mathscr{B}}^{\mathrm{GDDH}}(\lambda)$ is a negligible function of λ.

Remark 3: We can also define the analogue of discrete logarithm problem in k-graded encoding schemes as follows: Given a level-i encoding $u \in S_i^{(\alpha)}$, where $1 \le i \le k$ and $\alpha \in R$ is a random and nearly uniform element, the adversary must output a level-zero encoding $a \in S_0^{(\alpha)}$.

3 Computational VMSS schemes

We now present the formal model of VMSS schemes. A VMSS scheme $\Omega = (\mathtt{Stp}, \mathtt{Dist}, \mathtt{Ver}, \mathtt{Rec})$ can be described by the following algorithms:

- On input of a security parameter 1^λ, the set of participants $\mathscr{P} = \{P_1, \ldots, P_n\}$ and access structures $\Gamma_1, \ldots, \Gamma_m$, the setup algorithm Stp outputs some public parameters: $\mathtt{pms} \leftarrow \mathtt{Stp}(1^\lambda, \mathscr{P}, \Gamma_1, \ldots, \Gamma_m)$.
- On input of pms and the secrets $S_1, \ldots, S_m$, the distribution algorithm Dist broadcasts the set of shares $\{sh_i\}_{P_i \in \mathscr{P}}$ and some public output Proof. This public value is used to verify the correctness of the shares: $(\{sh_i\}_{P_i \in \mathscr{P}}, \mathtt{Proof}) \leftarrow \mathtt{Dist}(\mathtt{pms}, S_1, \ldots, S_m)$.
- On input of pms, Proof and sh_i, the verification algorithm $\mathtt{Ver}(\mathtt{pms}, \mathtt{proof}, sh_i)$ outputs True if the share sh_i is a valid share for the participant P_i or $\perp$ otherwise.
- On input of pms, Proof and the shares $\{sh_i\}_{P_i \in A}$ of the participants in an authorised set $A \in \Gamma_j$, the reconstruction algorithm Rec outputs the secret $S_j := \mathtt{Rec}(\mathtt{pms}, \mathtt{proof}, \{sh_i\}_{P_i \in A})$, where $1 \le j \le m$.

There exist three properties which the VMSS scheme must satisfy them. Let A_j be an authorised subset of the access structure Γ_j, $1 \le j \le m$ and B_j is an unauthorised set. These properties are as follows:

(i) *Correctness:* This property means that by using the shares of participants of A_j, the same secret S_j must be reconstructed.
(ii) *Verifiability:* In a VMSS scheme, there must exist some tests to prevent cheating by dealer or participants. The participants of A_j will recover the same secret S_j, $1 \le j \le m$ if these tests are passed correctly.
(iii) *Secrecy:* Let the dealer is honest. Then, the participants of B_j cannot reconstruct the secret S_j, $1 \le j \le m$ or even know any information about it, even knowing some of the other secrets.

In a secret sharing scheme, adversaries can be divided into active and passive [22]. An active adversary has full control over the corrupted participants, while a passive one only obtains the information held by them. On the other hand, adversaries can be classified into adaptive and static. An adaptive adversary at any time can attempt to corrupt a new participant, while a static one can only corrupt a participant at the beginning of the protocol.

3.1 IND-secrecy

Here, we define the IND-secrecy for a verifiable secret sharing scheme Ω by proposing an experiment between $\mathscr{A}$ as an adversary and $\mathscr{C}$ as a challenger in the computational scenario. This experiment means that $\mathscr{A}$ cannot distinguish between a random value and the shared secret.

Game $\mathscr{G}_3$:

(i) The set of participants $\mathscr{P}$ and also an access structure Γ are broadcast by the adversary $\mathscr{A}$.
(ii) $\mathscr{C}$ runs and then sends $\mathtt{pms} \leftarrow \mathtt{Stp}(1^\lambda, \mathscr{P}, \Gamma)$ to the adversary $\mathscr{A}$.
(iii) A subset $B \subset \mathscr{P}$ of corrupted participants is published by $\mathscr{A}$ such that $B \notin \Gamma$.
(iv) $\mathscr{C}$ chooses two random secrets S_0, S_1 and also a random bit $\beta \in \{0,1\}$. Next, he runs $(\{sh_i\}_{P_i \in \mathscr{P}}, \mathtt{Proof}) \leftarrow \mathtt{Dist}(\mathtt{pms}, S_0)$ and sends $(\{sh_i\}_{P_i \in B}, \mathtt{Proof})$ to $\mathscr{A}$, along with S_β.
(v) Finally, $\mathscr{A}$ publishes a bit $\beta' \in \{0,1\}$.

In this game, we define the advantage of the adversary $\mathscr{A}$ to be

$$\mathtt{Adv}_{\mathscr{A}}^{\mathrm{IND}}(\lambda) = \left|\mathtt{Prob}[\beta' = \beta] - \frac{1}{2}\right|.$$

We say that a VSS scheme Ω has indistinguishability of secrets in the standard model, if for any polynomial time adversary $\mathscr{A}$, $\mathtt{Adv}_{\mathscr{A}}^{\mathrm{IND}}(\lambda)$ is a negligible function of λ.

4 Proposed secret sharing scheme

Here, using MSPs, we propose our secret sharing scheme based on k-graded encoding schemes. In the next section, we show how verifiability can be achieved by using the k-graded encoding schemes.

Suppose that $\mathscr{P} = \{P_1, \ldots, P_n\}$ be a set of n participants. The dealer D firstly chooses a k-graded encoding scheme GES($R, \mathcal{S}$) as defined in Definition 5. Next, D runs $s, c \leftarrow \mathtt{Samp}(\mathtt{params})$ to get two level-zero encodings $s \in S_0^{(\alpha)} \subset \{0,1\}^*$ and $c \in S_0^{(\gamma)}$, where

$\alpha, \gamma \in R$ are random and nearly uniform elements. Then, D considers $s \in \{0,1\}^*$ as an element of the field $\mathbb{Z}_q^*$ in which q is a sufficiently large prime number and distributes the secret $S = \mathtt{Ext}(\mathtt{params}, P_{zt}, S_k^{(\alpha \cdot \gamma)})$ according to a given general access structure Γ as follows.

4.1 Setup: $\mathrm{Stp}(1^\lambda, \mathscr{P}, \Gamma)$

The dealer D runs the encoding procedure $\mathtt{Enc}(\mathtt{params}, k, c)$ to get the level-k encoding $u_c \in S_k^{(\gamma)}$. The dealer also chooses an MSP $(\mathbb{F} = \mathbb{Z}_q, \boldsymbol{M}, \psi)$ with a target vector $\boldsymbol{\xi}$ for access structure Γ. The public parameters are $\mathtt{pms} = (\mathrm{GES}(R, \mathscr{S}), u_c, q, \boldsymbol{M})$.

4.2 Distribution: $\mathrm{Dist}(\mathtt{pms}, S)$

The dealer shares the secret $S = \mathtt{Ext}(\mathtt{params}, P_{zt}, S_k^{(\alpha \cdot \gamma)})$ among n participants as follows:

- Suppose that $\boldsymbol{M} = [m_{ij}] \in \mathbb{F}^{n \times l}$ be the distribution matrix. Choose a distribution vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_l)^T \in \mathbb{F}^l$, where $\rho_1 = \mathtt{s}$ and ρ_i's, $2 \le j \le l$ are uniformly random chosen elements in $\mathbb{F}$. Now, compute $\boldsymbol{M\rho} = (z_1, z_2, \ldots, z_n)^T \in \mathbb{F}^n$.
- By using the bijection function $\psi: \mathscr{P} \to \{1, \ldots, n\}$, secretly transmit z_i to the participant $P_{\psi(i)}$ where $1 \le i \le n$. For simplicity, we assume that $\psi(i) = i$. Thus, the share of each participant P_i is z_i, $1 \le i \le n$.

4.3 Reconstruction: $\mathtt{Rec}(\mathtt{pms}, z_1, \ldots, z_n)$

The participants of an authorised set $A \in \Gamma$ can reconstruct the secret S as follows:

Firstly, all participants pool their shares. Then, according to the previous discussion in Section 2, for the authorised set A, there exist a vector $\boldsymbol{w}_A = (w_1, \ldots, w_{|A|})$ such that $\boldsymbol{\xi} = \boldsymbol{w}_A \boldsymbol{M}_A$, where $|A|$ denotes the number of elements of A. Thus, the participants of A can get the level-zero encoding $\mathtt{s} \in S_0^{(\alpha)}$ by the equation

$$\sum_{P_i \in A} (w_i \cdot z_i) = \mathtt{s} \in S_0^{(\alpha)}. \tag{1}$$

Finally, the participants of A can recover the secret S by firstly running the multiplication procedure $\mathtt{Mul}(\mathtt{params}, 0, \mathtt{s}, k, u_c) = \mathtt{s} \times u_c \in S_k^{(\alpha \cdot \gamma)}$ and then running the extraction procedure to get the secret

$$S = \mathtt{Ext}(\mathtt{params}, P_{zt}, \mathtt{s} \times u_c) = \mathtt{Ext}(\mathtt{params}, P_{zt}, S_k^{(\alpha \cdot \gamma)}).$$

5 Correctness and security proofs

5.1 Correctness

Here we show the correctness of our scheme. If the dealer and the participants behave honestly, the participants of the authorised set $A \in \Gamma$ can get the secret by firstly computing (1) as

$$\sum_{P_i \in A} (w_i \cdot z_i) = \sum_{P_i \in A} (w_i \cdot \boldsymbol{M}_i \boldsymbol{\rho}) = (\boldsymbol{w}_A \boldsymbol{M}_A \boldsymbol{\rho}) = (\boldsymbol{\xi \rho}) = \mathtt{s} \in S_0^{(\alpha)}$$

and then running the multiplication procedure $\mathtt{Mul}(\mathtt{params}, 0, \mathtt{s}, k, u_c) = \mathtt{s} \times u_c \in S_k^{(\alpha \cdot \gamma)}$. Finally, they run the extraction procedure to get the secret

$$S = \mathtt{Ext}(\mathtt{params}, P_{zt}, \mathtt{s} \times u_c) = \mathtt{Ext}(\mathtt{params}, P_{zt}, S_k^{(\alpha \cdot \gamma)}).$$

5.2 Secrecy

Now, we prove the security of the secret sharing scheme. We explain how an adversary for the proposed secret sharing scheme can be used to define an adversary that is a solver for the GDDH problem. More precisely, we reduce the hardness of GDDH problem to the computational security of the proposed scheme.

Theorem 1: If there is any polynomial-time adversary $\mathscr{A}$ who can break the secret sharing scheme, then there exists a polynomial-time adversary $\mathscr{B}$ that is a solver for the GDDH problem such that

$$\mathtt{Adv}_{\mathscr{B}}^{\mathrm{GDDH}}(\lambda) \le \mathtt{Adv}_{\mathscr{A}}^{\mathrm{IND}}(\lambda).$$

Proof: Consider an adaptive and active probabilistic polynomial time adversary $\mathscr{A}$ which acts according to the game $\mathscr{G}_3$ against the security of the proposed scheme, such that his advantage $\mathtt{Adv}_{\mathscr{A}}^{\mathrm{IND}}(\lambda) = \varepsilon_{\mathscr{A}}$ is non-negligible. In the following, we will construct an adversary $\mathscr{B}$ which acts according to the game $\mathscr{G}_2$ to solve the GDDH problem with a non-negligible advantage $\mathtt{Adv}_{\mathscr{B}}^{\mathrm{GDDH}}(\lambda) = \varepsilon_{\mathscr{B}}$ in polynomial time and uses $\mathscr{A}$ as a sub-routine. □

1. Based on the multilinearity parameter k and the security parameter λ, the challenger of the game $\mathscr{G}_2$ (i.e. $\mathscr{C}$) runs $(\mathtt{params}, P_{zt}) \leftarrow \mathtt{InstGen}(1^\lambda, k)$ to get a description of a k-graded encoding scheme $\mathrm{GES}(R, \mathscr{S})$.
2. For $j = 1, \ldots, k+1$, the challenger $\mathscr{C}$ firstly runs $a_j \leftarrow \mathtt{Samp}(\mathtt{params})$ to get level-zero encodings $a_j \in S_0^{(\alpha_j)}$, where $\alpha_1, \ldots, \alpha_{k+1} \in R$ are random and nearly uniform elements. Then, $\mathscr{C}$ performs the following three steps:
   - $\mathscr{C}$ runs $u_j \leftarrow \mathtt{Enc}(\mathtt{params}, 1, a_j)$ to get level-1 encodings $u_j \in S_1^{(\alpha_j)}$.
   - According to the multiplication procedure, $\mathscr{C}$ computes $\tilde{a} = a_1 \times \cdots \times a_{k+1} \in S_0^{\left(\prod_{j=1}^{k+1} \alpha_j\right)}$
   - $\mathscr{C}$ runs $\tilde{u} \leftarrow \mathtt{Enc}(\mathtt{params}, k, \tilde{a})$ to get the level-k encoding $\tilde{u} \in S_k^{\left(\prod_{j=1}^{k+1} \alpha_j\right)}$.
3. The challenger $\mathscr{C}$ runs $\hat{a} \leftarrow \mathtt{Samp}(\mathtt{params})$ to get $\hat{a} \in S_0^{(\alpha)}$ which is a level-zero encoding, where $\alpha \in R$ is a random and nearly uniform element. Then
   - $\mathscr{C}$ runs $\hat{u} \leftarrow \mathtt{Enc}(\mathtt{params}, k, \hat{a})$ to get level-k encoding $\hat{u} \in S_k^{(\alpha)}$.
4. Consequently, the challenger $\mathscr{C}$ considers two distributions $D_0 = \{(\mathtt{params}, P_{zt}, \{u_j\}_{j=0}^{k+1}, \tilde{u})\}$ in which $\tilde{u}$ is the level-k encoding of the right product $\prod_{j=1}^{k+1} \alpha_j$, and $D_1 = \{(\mathtt{params}, P_{zt}, \{u_j\}_{j=0}^{k+1}, \hat{u})\}$, where $\hat{u}$ is the level-k encoding of a random element α. Then, $\mathscr{C}$ randomly selects a bit $\beta \in \{0,1\}$ and sends $D_\beta = \{(\mathtt{params}, P_{zt}, \{u_j\}_{j=0}^{k+1}, u_\beta)\}$ to $\mathscr{B}$ (u_β is either identical to $\tilde{u}$ or $\hat{u}$).
5. The set of participants $\mathscr{P}$ and also an access structure Γ are broadcast by the adversary $\mathscr{A}$ (the start of the game $\mathscr{G}_3$).
6. $\mathscr{B}$ acts as a challenger for $\mathscr{A}$ in the secret sharing scheme. So simulator $\mathscr{B}$ runs the multiplication procedure $u_c := a_2 \times \cdots \times a_{k+1} \in S_k^{\left(\prod_{j=2}^{k+1} \alpha_j\right)}$ to get the level-k encoding u_c. Next, $\mathscr{B}$ considers the (unknown) level-zero encoding $a_1 \in S_0^{(\alpha_1)} \subset \{0,1\}^*$ as an element of the field $\mathbb{Z}_q^*$ in which q is a sufficiently large prime number to distribute the (unknown) secret
$$S^0 = \mathtt{Ext}(\mathtt{params}, P_{zt}, a_1 \times u_c) = \mathtt{Ext}\left(\mathtt{params}, P_{zt}, S_k^{\left(\prod_{j=1}^{k+1} \alpha_j\right)}\right)$$
using the general access structure Γ.
7. The 'adaptive' adversary $\mathscr{A}$ publishes a subset $B_0 \subset \mathscr{P}$ which is the corrupted participants at the beginning of the game $\mathscr{G}_3$.

Next, $\mathscr{B}$ (as the challenger of this game) guesses the set of all corrupted participants B for which $B_0 \subset B$ and $B \notin \Gamma$ (for simplicity, we assume that $|B| = t$).

8. In order to share the (unknown) secret S^0, the simulator $\mathscr{B}$ uses $(\mathbb{F} = \mathbb{Z}_q, \boldsymbol{M}, \psi)$ with target vector $\boldsymbol{\xi}$ which is an MSP for access structure Γ:

   - $\mathscr{B}$ broadcasts public parameters $\mathtt{pms} = (\mathtt{GES}(R, \mathcal{S}), u_c, q, \boldsymbol{M})$.
   - Using the integers $\rho_2, \ldots, \rho_l \in \mathbb{F}$ which are chosen uniformly at random, the challenger $\mathscr{B}$ constructs a vector $\vec{\rho} = (0, \rho_2, \ldots, \rho_l)^T$.
   - As previously discussed in Section 2, there exists a sweeping vector $\boldsymbol{k} = (k_1, k_2, \ldots, k_l)^T \in \mathbb{F}^l$ with $k_1 = 1$ such that for the unauthorised set B ($B \notin \Gamma$), it holds that $\boldsymbol{M}_B \boldsymbol{k} = \boldsymbol{0} \in \mathbb{F}^t$. Using the vector $\boldsymbol{k}$, the distribution vector is computed as $\boldsymbol{\rho}' = \boldsymbol{\rho} + a_1 \boldsymbol{k} = (a_1, \rho_2', \ldots, \rho_l')^T$.
   - For each participant $P_i \in B$, the simulator $\mathscr{B}$ secretly transmits the shares $z_i = \boldsymbol{M}_i \boldsymbol{\rho}$, that is because for $P_i \in B$ we have

$$z_i = \boldsymbol{M}_i \boldsymbol{\rho}' = \boldsymbol{M}_i(\boldsymbol{\rho} + a_1 \boldsymbol{k}) = \boldsymbol{M}_i \boldsymbol{\rho} + a_1 \boldsymbol{M}_i \boldsymbol{k} = \boldsymbol{M}_i \boldsymbol{\rho} + 0 = \boldsymbol{M}_i \boldsymbol{\rho}.$$

   - For the rest participants $P_i \in \mathscr{P} \setminus B$, the simulator $\mathscr{B}$ secretly transmits a uniformly random element z_i of $\mathbb{F}$ as their shares.

   Thus, $\mathscr{B}$ acts as a challenger for $\mathscr{A}$ in distribution phase $(\{z_i\}_{P_i \in \mathscr{P}}) \leftarrow \mathtt{Dist}(\mathtt{pms}, S^0)$. Then, $\mathscr{B}$ sends $S^\beta = \mathtt{Ext}(\mathtt{params}, P_{zt}, u_\beta)$ to $\mathscr{A}$.
9. Suppose that $B_1 \subset \mathscr{P} \setminus B_0$, be the set of all participants corrupted by the 'adaptive' adversary $\mathscr{A}$ such that $|B_0 \cup B_1| \leq t$. If it holds that $B_1 \nsubseteq B \setminus B_0$, then $\mathscr{B}$ leaves the game $\mathscr{G}_2$ and randomly gives a bit as final output. Otherwise $\mathscr{B}$ sends the share z_i of each $P_i \in B_1$ to $\mathscr{A}$. Then,

   - The adversary $\mathscr{A}$ outputs $b \in \{0, 1\}$ as follows: (game $\mathscr{G}_3$)
     - If $S^\beta = S^0$, $\mathscr{A}$ outputs $b = 0$.
     - Otherwise, $\mathscr{A}$ outputs $b = 1$.
10. $\mathscr{B}$ outputs $\beta' = b$ as output of the game $\mathscr{G}_2$.

Now, suppose that $\mathtt{Succ}_{\mathscr{B}}$, $\mathtt{Succ}_{\mathscr{A}}$ and $\mathtt{Fail}$ be the event that $\beta' = \beta$, the event that $b = \beta$ and the event that $\mathscr{B}$ leaves the game $\mathscr{G}_2$ at step 9, respectively. For the events $\mathtt{Fail}$ and $\mathtt{Succ}_{\mathscr{B}}$, note that if $\mathtt{Fail}$ occurs, then the probability of $\mathtt{Succ}_{\mathscr{B}}$ will be 1/2. Thus

$$\begin{aligned}
\mathtt{Pr}[\mathtt{Succ}_{\mathscr{B}}] &= \mathtt{Pr}[\mathtt{Succ}_{\mathscr{B}}]\mathtt{Pr}[\mathtt{Fail}] + \mathtt{Pr}[\mathtt{Succ}_{\mathscr{A}}]\mathtt{Pr}[\neg\mathtt{Fail}] \\
&= \frac{1}{2}\mathtt{Pr}[\mathtt{Fail}] + \left(\frac{1}{2} + \varepsilon_{\mathscr{A}}\right)\mathtt{Pr}[\neg\mathtt{Fail}] \\
&= \frac{1}{2}\mathtt{Pr}[\mathtt{Fail}] + \frac{1}{2}\mathtt{Pr}[\neg\mathtt{Fail}] + \mathtt{e}_{\mathscr{A}}\mathtt{Pr}[\neg\mathtt{Fail}] \\
&= \frac{1}{2}(\mathtt{Pr}[\mathtt{Fail}] + \mathtt{Pr}[\neg\mathtt{Fail}]) + \mathtt{e}_{\mathscr{A}}\mathtt{Pr}[\neg\mathtt{Fail}] \\
&= \frac{1}{2} + \varepsilon_{\mathscr{A}}\mathtt{Pr}[\neg\mathtt{Fail}]
\end{aligned}$$

and so $\varepsilon_{\mathscr{B}} = \varepsilon_{\mathscr{A}}\mathtt{Pr}[\neg\mathtt{Fail}]$. For any possible choice of B_1, the probability of $\neg\mathtt{Fail}$ is computed as

$$\mathtt{Pr}[\neg\mathtt{Fail}] = \frac{\binom{t - |B_0|}{|B_1|}}{\binom{n - |B_0|}{|B_1|}}$$

which ranges from 1 if $B_1 = \varnothing$ to $\binom{n}{t}^{-1}$ if $B_0 = \varnothing$ and $|B_1| = t$. Consequently, it holds that $\varepsilon_{\mathscr{B}} \leq \varepsilon_{\mathscr{A}}$.

5.3 Instantiating the used graded encoding scheme

In this paper, we have presented a general VMSS scheme whose security is based on a graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$. To use the proposed VMSS scheme, the graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ must be instantiated. The instantiation of $\mathrm{GES}(R, \mathcal{S})$ can be done by selecting one of the existing graded encoding schemes [13, 37, 38, 35] or ones that are offered in the future. Here, we discuss how $\mathrm{GES}(R, \mathcal{S})$ can be instantiated to ensure the VMSS scheme's security.

As we saw, the security of the proposed VMSS scheme is based on the GDDH problem. So, in the graded encoding scheme that is chosen, the GDDH problem must be hard.

In [39], Hu and Jia provided their cryptanalysis of GGH13 that led to breaking the GGH13-GDDH problem. So, instantiating $\mathrm{GES}(R, \mathcal{S})$ using GGH13 cannot be a good choice. Very recently, Farshim *et al.* [35] showed how to construct a graded encoding scheme from obfuscation. They proved that in their proposed graded encoding scheme, the GDDH problem is hard, if the used components are secure. So, $\mathrm{GES}(R, \mathcal{S})$ should be instantiated by this graded encoding scheme which leads to the security of the proposed VMSS scheme.

6 VMSS scheme

Based on the scheme proposed in Section 4, in this section, we describe a technique to construct a verifiable multi-secret sharing scheme from k-graded encoding schemes.

6.1 Description of the scheme

Suppose that $\mathscr{P} = \{P_1, \ldots, P_n\}$ be a set of n participants. The dealer D firstly chooses a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ as defined in Definition 5. Next, D runs $\mathtt{s}_1, \ldots, \mathtt{s}_m, c, b \leftarrow \mathtt{Samp}(\mathtt{params})$ to get level-zero encodings $\mathtt{s}_1 \in S_0^{(\alpha_1)} \subset \{0,1\}^*, \ldots, \mathtt{s}_m \in S_0^{(\alpha_m)} \subset \{0,1\}^*$, $c \in S_0^{(\gamma)}$ and $b \in S_0^{(\beta)}$, where $\alpha_1, \ldots, \alpha_m, \gamma, \beta \in R$ are random and nearly uniform elements. Then, D considers $s_j \in \{0,1\}^*$ as an element of the field $\mathbb{Z}_q^*$ in which q is a sufficiently large prime number and distributes the secret $S_j = \mathtt{Ext}\left(\mathtt{params}, P_{zt}, S_k^{(\alpha_j \cdot \gamma)}\right)$ according to a given general access structure Γ_j as follows ($1 \leq j \leq m$).

6.1.1 Setup: $\mathtt{Stp}(1^\lambda, \mathscr{P}, \Gamma_1, \ldots, \Gamma_m)$*:* The dealer runs the encoding procedures $\mathtt{Enc}(\mathtt{params}, k, c)$ and $\mathtt{Enc}(\mathtt{params}, k, b)$ to get the level-k encodings $u_c \in S_k^{(\gamma)}$ and $u_b \in S_k^{(\beta)}$. The dealer also runs the multiplication procedures $\mathtt{Mul}(\mathtt{params}, k, u_b, 0, c) = u_b \times c \in S_k^{(\beta \cdot \gamma)}$ and $\mathtt{Mul}(\mathtt{params}, 0, b, k, u_c) = b \times u_c \in S_k^{(\beta \cdot \gamma)}$. Next, the dealer chooses an MSP $(\mathbb{F} = \mathbb{Z}_q, \boldsymbol{M}, \psi)$ with target vectors $\boldsymbol{\xi}_j$ for access structures Γ_j, where $1 \leq j \leq m$. The public parameters are $\mathtt{pms} = (\mathtt{GES}(R, \mathcal{S}), u_c, u_b \times c, q, \boldsymbol{M})$.

6.1.2 Distribution: $\mathtt{Dist}(\mathtt{pms}, S_j)$ *:* The dealer shares the secret $S_j = \mathtt{Ext}\left(\mathtt{params}, P_{zt}, S_k^{(\alpha_j \cdot \gamma)}\right)$, $1 \leq j \leq m$ among n participants as follows:

- Suppose that $\boldsymbol{M} = [m_{ij}] \in \mathbb{F}^{n \times l}$ be the distribution matrix. Choose a distribution vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_l)^T \in \mathbb{F}^l$ where $\rho_j = \mathtt{s}_j$ and ρ_i's, $m + 1 \leq i \leq l$ are uniformly random chosen elements in $\mathbb{F}$ ($1 \leq j \leq m$) and then compute $\boldsymbol{M}\boldsymbol{\rho} = (z_1, z_2, \ldots, z_n)^T$.
- By using the bijection function $\psi: \mathscr{P} \to \{1, \ldots, n\}$, secretly transmit z_i to the participant $P_{\psi(i)}$ where $1 \leq i \leq n$. For

simplicity, we assume that $\psi(i) = i$. Thus, the share of each participant P_i is z_i.

- For $1 \le i \le l$, run the addition procedure to compute $\rho_i \cdot (b \times u_c) = (b \times u_c) + \cdots + (b \times u_c) \in S_k^{(\rho_i(\beta \cdot \gamma))}$.
- The public outputs of the distribution phase are $\text{proof} = \{\rho_i \cdot (b \times u_c)\}_{i=1}^{l}$.

6.1.3 Verification of shares: $\texttt{Ver}(\texttt{pms}, \texttt{proof}, z_1, \ldots, z_n)$ *:* Participant P_i runs the addition procedure to compute the values $X_i = \sum_{h=1}^{l} (m_{ih}(\rho_h \cdot (b \times u_c))) \in S_k^{\left(\sum_{h=1}^{l} (m_{ih} \cdot (\rho_h \cdot (\beta \cdot \gamma)))\right)}$ and $z_i \cdot (u_b \times c) = (u_b \times c) + \cdots + (u_b \times c) \in S_k^{(z_i(\beta \cdot \gamma))}$. Next, he can verify whether his share is valid by checking the equation

$$\texttt{Ext}(\texttt{params}, P_{zt}, X_i) = \texttt{Ext}(\texttt{params}, P_{zt}, z_i \cdot (u_b \times c)).$$

6.1.4 Reconstruction: $\texttt{Rec}(\texttt{pms}, z_1, \ldots, z_n)$ *:* The participants of an authorised set $A \in \Gamma$ can reconstruct the secret S_j as follows:

(i) Firstly, each $P_i \in A$ participant chooses a vector $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{lj})^T \in \mathbb{F}^l$ such that $\langle \boldsymbol{\xi}_j, \boldsymbol{r}_j \rangle = 0$ and computes $\boldsymbol{M}\boldsymbol{r}_j = (t_{1j}, t_{2j}, \ldots, t_{nj})^T$.
(ii) Next, all participants pool their pseudo shares. The pseudo share of the participant $P_i \in A$ is $z_i + t_{ij}$.
(iii) Verifiability in the reconstruction: Suppose that $P_i \in A$. He runs the addition procedure to compute and broadcast $Y_{ij} = t_{ij} \cdot (u_b \times c) = (u_b \times c) + \cdots + (u_b \times c) \in S_k^{(t_{ij}(\beta \cdot \gamma))}$. Now, any participant $P' \in A$ can verify the P_i's pseudo share by checking the following equation:

$$\texttt{Ext}(\texttt{params}, P_{zt}, (X_i + Y_{ij})) = \texttt{Ext}(\texttt{params}, P_{zt}, (z_i + t_{ij}) \cdot (u_b \times c)).$$

(iv) According to the previous discussion in Section 2, for the authorised set A, there exist a vector $\boldsymbol{w}_{Aj} = (w_{1j}, \ldots, w_{|A|j})$ such that $\boldsymbol{\xi}_j = \boldsymbol{w}_{Aj} M_A$, where $|A|$ denotes the number of elements of A. Thus, the participants of A can get the level-zero encoding $\mathrm{s}_j \in S_0^{(\alpha_j)}$ by the following equation:

$$\sum_{P_i \in A} (w_{ij} \cdot (z_i + t_{ij})) = \mathrm{s}_j \in S_0^{(\alpha_j)}. \tag{2}$$

Finally, the participants of A can recover the secret S_j by firstly running the multiplication procedure $\texttt{Mul}(\texttt{params}, 0, \mathrm{s}_j, k, u_c) = \mathrm{s}_j \times u_c \in S_k^{(\alpha_j \cdot \gamma)}$ and then running the extraction procedure to get the secret

$$S_j = \texttt{Ext}(\texttt{params}, P_{zt}, \mathrm{s}_j \times u_c) = \texttt{Ext}(\texttt{params}, P_{zt}, S_k^{(\alpha_j \cdot \gamma)}).$$

6.2 Correctness

Here we show the correctness of our scheme. If the dealer and the participants behave honestly, the participants of the authorised set $A \in \Gamma$ can get the secret S_j by firstly computing (2) as

$$\begin{aligned} \sum_{P_i \in A} (w_{ij} \cdot (z_i + t_{ij})) &= \sum_{P_i \in A} (w_{ij} z_i + w_{ij} t_{ij}) \\ &= \boldsymbol{w}_{Aj} M_A \boldsymbol{\rho} + \boldsymbol{w}_{Aj} M_A \boldsymbol{r}_j \\ &= \boldsymbol{\xi}_j \boldsymbol{\rho} + \boldsymbol{\xi}_j \boldsymbol{r}_j = \mathrm{s}_j \end{aligned}$$

and then running the multiplication procedure $\texttt{Mul}(\texttt{params}, 0, \mathrm{s}_j, k, u_c) = \mathrm{s}_j \times u_c \in S_k^{(\alpha_j \cdot \gamma)}$. Finally, they run the extraction procedure to get the secret

$$S_j = \texttt{Ext}(\texttt{params}, P_{zt}, \mathrm{s}_j \times u_c) = \texttt{Ext}(\texttt{params}, P_{zt}, S_k^{(\alpha_j \cdot \gamma)}).$$

6.3 Verifiability

Now, we determine the verifiability of our scheme using the following theorems:

Theorem 2: If a participant P_i accepts his share, then there exists a unique value z_i for which $\texttt{Ext}(\texttt{params}, P_{zt}, X_i) = \texttt{Ext}(\texttt{params}, P_{zt}, z_i \cdot (u_b \times c))$.

Proof: Suppose that the share of P_i is computed as $z_i = \boldsymbol{M}_i \boldsymbol{\rho}$ and the dealer sends the value z_i' to P_i. If P_i accepts z_i', then

$$\begin{aligned} \texttt{Ext}(\texttt{params}, P_{zt}, X_i) &= \texttt{Ext}(\texttt{params}, P_{zt}, z_i \cdot (u_b \times c) \in S_k^{(z_i(\beta \cdot \gamma))}) \\ &= \texttt{Ext}(\texttt{params}, P_{zt}, z'_i \cdot (u_b \times c) \in S_k^{(z'_i(\beta \cdot \gamma))}) \end{aligned} \tag{3}$$

where the first equality is due to the fact that

$$X_i = \sum_{h=1}^{l} (m_{ih}(\rho_h \cdot (b \times u_c))) = z_i \cdot (b \times u_c)$$

and $z_i \cdot (u_b \times c)$ are in $S_k^{(z_i(\beta \cdot \gamma))}$. Thus, the second equality of (3) leads to $z_i = z_i'$. □

Theorem 3: If a participant P_i accepts P_j's pseudo share, then there exists a unique value $z_i + t_{ij}$ for which $\texttt{Ext}(\texttt{params}, P_{zt}, (X_i + Y_{ij})) = \texttt{Ext}(\texttt{params}, P_{zt}, (z_i + t_{ij}) \cdot (u_b \times c))$.

Proof: This proof is similar to that of Theorem 2. Suppose that P_j sends $(z_i + t_{ij})'$ in the reconstruction phase. If P_i accepts P_j's value $(z_i + t_{ij})'$, then

$$\begin{aligned} \texttt{Ext}(\texttt{params}, P_{zt}, (X_i + Y_{ij})) &= \texttt{Ext}(\texttt{params}, P_{zt}, (z_i \cdot (u_b \times c) \\ &\quad + t_{ij} \cdot (u_b \times c)) \in S_k^{((z_i + t_{ij})(\beta \cdot \gamma))}) \\ &= \texttt{Ext}(\texttt{params}, P_{zt}, (z_i + t_{ij}) \cdot (u_b \times c) \\ &\in S_k^{((z_i + t_{ij})(\beta \cdot \gamma))}) \\ &= \texttt{Ext}(\texttt{params}, P_{zt}, (z_i + t_{ij})' \cdot (u_b \times c) \\ &\in S_k^{((z_i + t_{ij})'(\beta \cdot \gamma))}) \end{aligned} \tag{4}$$

where the first equality is due to the fact that

$$X_i = \sum_{h=1}^{l} (m_{ih}(\rho_h \cdot (b \times u_c))) = z_i \cdot (b \times u_c)$$

and $z_i \cdot (u_b \times c)$ are in $S_k^{(z_i(\beta \cdot \gamma))}$. Thus, the third equality of (4) leads to $(z_i + t_{ij}) = (z_i + t_{ij})'$. □

7 Conclusion

In this paper, we proposed the first VMSS scheme based on graded encoding schemes. More precisely, the security of the proposed VMSS scheme is based on the GDDH problem. As we mentioned earlier, there have already been verifiable (multi)-secret sharing schemes based on multilinear maps (the maps which do not really exist) [2, 31]. So our VMSS scheme is the first realisation of such verifiable schemes.

Here, we first propose a comparison of the basic properties in the new VMSS scheme with (t, n) threshold verifiable (multi)-secret sharing schemes in the literature [2, 31]. For easiness, the abbreviations S_1 and S_2 are used for our first and second schemes, respectively. The results of this comparison are summarised in Table 1.

Table 1 Basic properties of the schemes

Property	Peng and Tian [2]	Peng and Tian [31]	S_1	S_2
have indistinguishability against				
chosen secret attacks	yes	yes	yes	—
are secure against active attack	no	no	yes	—
are secure against adaptive attack	no	no	yes	—
assumption	MCDH	MCDH	GDDH	—
method	interpolation	interpolation	MSP	MSP
general access structure	no	no	yes	yes
have verification property for dealer	yes	yes	no	yes
have verification property for participants	yes	yes	no	yes
distribute multi-secret	yes	no	no	yes
the share of each participant is reusable	no	—	—	yes
it is possible that all secrets are not reconstructed simultaneously	no	—	—	yes

Table 2 Comparison of the computational complexities

Step	Peng and Tian [2]	Peng and Tian [31]	S_2
setup	nT_m	nT_m	$(m+6)T_{ep}$
distribution	$((nm+1)(t-1)+n)T_m$	$((n+1)t+n-1)T_m$	nlT_m
	$+(n(m-1)(t-1))T_a$	$+n(t-1)T_a$	$+n(l-1)T_a$
	$+(t+n)T_{map}$	$+nT_{map}$	$+lT_{ep}$
verification	$t^2T_m+2T_{map}$	$(2t-1)T_a+2T_{map}$	$(2l+1)T_{ep}$
reconstruction	$(2t(m-1))T_m$	t^2T_m	$\|A\|(nl+1)T_m$
	$+(t-1)T_a$		$+(\|A\|n(l-1)+2\|A\|-1)T_a$
			$+2T_{ep}$

Table 3 Comparison of the number of public values

	Peng and Tian [2]	Peng and Tian [31]	S_2
number of public values	$(m+1)n+t$	$4n+m+t-1$	$(n+1)l+3$

Now, we use the following notations to analyse the number of public values and also complexities of the proposed VMSS scheme:

- T_m: The time required to execute a multiplication operation.
- T_a: The time required to execute an addition operation.
- T_{map}: The time required to execute the used multilinear map.
- T_{ep}: The time required to execute one of the efficient procedures of the used k-graded encoding scheme.

In Tables 2 and 3, we give a comparison of the schemes mentioned above. From the comparison in the tables, we mention the main properties of the new proposed VMSS scheme:

- It is the first VMSS scheme based on the GDDH problem which has indistinguishability against chosen secret attacks. Furthermore, our VMSS scheme is secure against active and adaptive adversaries.
- Every participant P_i uses his pseudo share z_i+t_{ij} for reconstructing the secret S_j. So, the P_i's real share z_i will not be disclosed even after reconstruction of all secrets and therefore the proposed scheme is a multi-use secret sharing scheme.
- It is impossible for dealers to cheat since every participant can check the reality of his share. It is also impossible for every participant to cheat since everyone can check the reality of pseudo shares that they give.
- It is very practical in the cases that all of the participants do not have the same level power for reconstructing the secrets.

8 References

[1] Ghasemi, R., Safi, A., Dehkordi, M.H.: 'Efficient multi-secret sharing scheme using new proposed computational security model', *Int. J. Commun. Syst.*, 2017, **31**, (1), p. e3399

[2] Peng, Q., Tian, Y.: 'Publicly verifiable secret sharing scheme and its application with almost optimal information rate', *Sec. Commun. Netw.*, 2016, **9**, (18), pp. 6227–6238

[3] Qin, H., Dai, Y., Wang, Z.: 'A secret sharing scheme based on (t, n) threshold and adversary structure', *Int. J. Inf. Secur.*, 2009, **8**, (5), pp. 379–385

[4] Liu, Y.: 'Linear (k, n) secret sharing scheme with cheating detection', *Sec. Commun. Netw.*, 2016, **9**, (13), pp. 2115–2121

[5] Asmuth, C., Bloom, J.: 'A modular approach to key safeguarding', *IEEE Trans. Inf. Theory*, 1983, **29**, (2), pp. 208–210

[6] Liu, Y., Harn, L., Chang, C.-C.: 'A novel verifiable secret sharing mechanism using theory of numbers and a method for sharing secrets', *Int. J. Commun. Syst.*, 2015, **28**, (7), pp. 1282–1292

[7] Cramer, R., Damgård, I., Maurer, U.: 'General secure multi-party computation from any linear secret-sharing scheme'. Advances in Cryptology-EUROCRYPT, 2000, pp. 316–334

[8] Dehkordi, M.H., Mashhadi, S., Oraei, H.: 'A proactive multi stage secret sharing scheme for any given access structure', *Wirel. Pers. Commun.*, 2019, **104**, (1), pp. 491–503

[9] Karchmer, M., Wigdersonm, A.: 'On span programs'. Structure in Complexity Theory Conf., 1993, pp. 102–111

[10] Liu, M., Xiao, L., Zhang, Z.: 'Linear multi-secret sharing schemes based on multi-party computation', *Finite Fields Appl.*, 2006, **12**, (4), pp. 704–713

[11] Hsu, C.F., Cheng, Q., Tang, X., *et al.*: 'An ideal multi-secret sharing scheme based on MSP', *Inf. Sci.*, 2011, **181**, (7), pp. 1403–1409

[12] Boneh, D., Silverberg, A.: 'Applications of multilinear forms to cryptography', *Contemp. Math.*, 2003, **324**, (1), pp. 71–90

[13] Garg, S., Gentry, C., Halevi, S.: 'Candidate multilinear maps from ideal lattices'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Berlin, Heidelberg, 2013, pp. 1–17

[14] Lin, H., Tessaro, S.: 'Indistinguishability obfuscation from trilinear maps and block-wise local PRGs'. Annual Int. Cryptology Conf., Cham, 2017, pp. 630–660

[15] Ananth, P., Sahai, A.: 'Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Cham, 2017, pp. 152–181

[16] Garg, S., Gentry, C., Halevi, S., *et al.*: 'Candidate indistinguishability obfuscation and functional encryption for all circuits', *SIAM J. Comput.*, 2016, **45**, (3), pp. 882–929
[17] Ananth, P., Jain, A., Sahai, A.: 'Robust transforming combiners from indistinguishability obfuscation to functional encryption'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Cham, 2017, pp. 91–121
[18] Garg, S., Gentry, C., Sahai, A., *et al.*: 'Witness encryption and its applications'. Proc. of the Forty-fifth Annual ACM Symp. on Theory of Computing, ACM, 2013, pp. 467–476
[19] Gentry, C., Lewko, A., Waters, B.: 'Witness encryption from instance independent assumptions'. Int. Cryptology Conf., Berlin, Heidelberg, 2014, pp. 426–443
[20] Garg, S., Gentry, C., Halevi, S., *et al.*: 'On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input', *Algorithmica*, 2017, **79**, (4), pp. 1353–1373
[21] Ruiz, A., Villar, J.L.: 'Publicly verifiable secret sharing from Paillier's cryptosystem', *WEWoRC*, 2005, **74**, pp. 98–108
[22] Heidarvand, S., Villar, J.L.: 'Public verifiability from pairings in secret sharing schemes'. Int. Workshop on Selected Areas in Cryptography, Berlin, Heidelberg, 2009, pp. 294–308
[23] Jhanwar, M.P.: 'A practical (non-interactive) publicly verifiable secret sharing scheme'. Int. Conf. on Information Security Practice and Experience, Berlin, Heidelberg, 2011, pp. 273–287
[24] Wu, T.Y., Tseng, Y.M.: 'A pairing-based publicly verifiable secret sharing scheme', *J. Syst. Sci. Complex.*, 2011, **24**, (1), pp. 186–194
[25] Gan, Y., Wang, L., Wang, L., *et al.*: 'Publicly verifiable secret sharing scheme with provable security against chosen secret attacks', *Int. J. Distrib. Sens. Netw.*, 2013, **9**, (2), pp. 1–9
[26] Wu, T.Y., Tseng, Y.M.: 'Publicly verifiable multi-secret sharing scheme from bilinear pairings', *IET Inf. Sec.*, 2013, **7**, (3), pp. 239–246
[27] Herranz, J., Ruiz, A., Saez, G.: 'Sharing many secrets with computational provable security', *Inf. Process. Lett.*, 2013, **113**, (14), pp. 572–579
[28] Mashhadi, S.: 'Computationally-secure multiple secret sharing: models, schemes, and formal security analysis', *ISC Int. J. Inf. Sec.*, 2015, **7**, (2), pp. 91–99
[29] Rajabi, B., Eslami, Z.: 'A verifiable threshold secret sharing scheme based on lattices', *Inf. Sci.*, 2018, https://doi.org/10.1016/j.ins.2018.11.004
[30] Shen, J., Liu, D., Sun, X., *et al.*: 'Efficient cloud-aided verifiable secret sharing scheme with batch verification for smart cities', *Future Gener. Comput. Syst.*, 2018, https://doi.org/10.1016/j.future.2018.10.049
[31] Peng, Q., Tian, Y.: 'A publicly verifiable secret sharing scheme based on multilinear Diffie-Hellman assumption', *Int. J. Netw. Sec.*, 2016, **18**, (6), pp. 1192–1200
[32] Hsu, C.F., Cui, G.H., Cheng, Q., *et al.*: 'A novel linear multi-secret sharing scheme for group communication in wireless mesh networks', *J. Netw. Comput. Appl.*, 2011, **34**, (2), pp. 464–468
[33] Hsu, C.F., Harn, L., Cui, G.: 'An ideal multi-secret sharing scheme based on connectivity of graphs', *Wirel. Pers. Commun.*, 2014, **77**, (1), pp. 383–394
[34] Xiao, L., Liu, M.: 'Linear multi-secret sharing schemes', *Sci. China F: Inf. Sci.*, 2005, **48**, (1), pp. 125–136
[35] Farshim, P., Hesse, J., Hofheinz, D., *et al.*: 'Graded encoding schemes from obfuscation'. IACR Int. Workshop on Public Key Cryptography, Cham, 2018, pp. 371–400
[36] Garg, S.: 'Candidate Multilinear Maps'. PhD thesis, University of California Los Angeles, 2013
[37] Langlois, A., StehlÃ©, D., Steinfeld, R.: 'GGHLite: more efficient multilinear maps from ideal lattices'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Berlin, Heidelberg, 2014, pp. 239–256
[38] Coron, J.-S., Lepoint, T., Tibouchi, M.: 'New multilinear maps over the integers'. Annual Cryptology Conf., Berlin, Heidelberg, 2015, pp. 267–286
[39] Hu, Y., Jia, H.: 'Cryptanalysis of GGH map'. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, Berlin, Heidelberg, 2016, pp. 537–565