Computationally Secure Cheating Identifiable Multi-Secret Sharing for General Access Structure*

Partha Sarathi Roy[1], Angsuman Das[2], and Avishek Adhikari[1]

[1] Department of Pure Mathematics,
University of Calcutta, Kolkata, India
{royparthasarathi0,avishek.adh}@gmail.com
[2] Department of Mathematics,
St. Xavier's College, Kolkata, India
angsumandas054@gmail.com

Abstract. Secret sharing scheme is a key component of distributed cryptosystems. In its basic form, secret sharing schemes can tolerate honest but curious adversary. But, in modern open system environment, adversary can behave maliciously i.e., the adversary can do anything according to his available computational resources. To get rid of such adversary, cheating identifiable (multi) secret sharing scheme plays an important role. Informally, cheating identifiable (multi) secret sharing scheme can identify the cheating participants, who are under the control of malicious adversary, and recover the correct secret whenever possible. However, to achieve unconditional security against such adversary, share size should be at least equal to the size of the secret. As a result, the need for computational notion of security of such schemes, which can accommodate smaller share size, has been felt over the years, specially in case of multi-secret sharing schemes. In this paper, we propose a notion of security for computationally secure cheating identifiable multi-secret sharing scheme for general access structure along with a construction which is secure under this new notion.

Keywords: cheating identifiable secret sharing, general access structure, computational security.

1 Introduction

Secret sharing scheme is a corner stone of secure distributed cryptographic protocols. It is also an essential building block for *encryption* schemes (specially *identity based encryption scheme*). Informally speaking, a *secret sharing scheme* (SSS) allows a dealer $\mathcal{D}$ to split a secret s into different pieces, called *shares*, which are given to a set of players $\mathcal{P}$, such that only certain qualified subsets

* Research supported in part by National Board for Higher Mathematics, Department of Atomic Energy, Government of India (No 2/48(10)/2013/NBHM(R.P.)/R&D II/695).

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 278–287, 2015.

of players can recover the secret using their respective shares. The collection of those qualified set of players is called *access structure* Γ_s corresponding to the secret s.

Blakley [2] and Shamir [16], in 1979, independently, introduced the notion of secret sharing scheme with a construction for threshold access structure. Presently, there exists a rich literature of secret sharing schemes with advanced features like general access structures (where qualified subsets are not all of same size t), multiple secrets (when number of secrets to be shared is more than one), verifiability, multi-usability (reconstruction of one secret does not endanger the security of the other secrets). But, some important issues remain open after extensive work of last three decades. In this paper, we deal with one of them. We consider cheating identifiable multi-secret sharing for general access structure which is an enhanced version of multi-secret sharing for general access that can tolerate any number of malicious participants and capture more realistic scenarios. In this scenario, the dealer is assumed to be honest and the goal is to identify the cheaters and to recover the correct secret whenever possible. In this work, we focus on public cheater identification, where reconstruction of the secret and cheater identification can be performed by a third party.

Most of the cheating identifiable secret sharing schemes proposed and analysed so far enjoy unconditional (or information-theoretic) security. Though there is an advantage that information theoretically secure scheme can tolerate computationally unbounded adversary, but there are some crucial drawbacks, such as requirement of honest majority, large amount of secret information. An alternative solution can be relying on computational security, by which tolerance of arbitrary number of dishonest participants is possible with lower share size (secret information), that serves well in practical purposes.

1.1 Related Work

The idea and construction of computationally secure secret sharing schemes came into existence with various proposals [1,8,6,15,4,3,11]. In 1994, He-Dawson [8] proposed a multi-stage (t, n) threshold secret sharing scheme. In 2007, Geng *et al.* [6] proposed a multi-use threshold secret sharing scheme using one-way hash function and pointed out that the He-Dawson scheme was actually an one-time-use scheme and can not endure conspiring attacks. A SSS is said to be *multi-use* if even after a secret is reconstructed by some players, the share remain hidden from the adversary. Generally, to make a scheme multi-use, the players do not broadcast the original share but a shadow or image of that share, which is actually an entity that depends on the original share. This image or shadow is known as the *pseudo-share*. Multi-use multi-secret sharing for general access structure was first introduced in [15,4].

Herranz *et. al.* [9], [10] formalize the computational notion of security for multi-secret sharing schemes with a concrete construction. In [12], authors discussed formal security notion for cheating identifiable threshold (single) secret sharing scheme. But, up to the best of our knowledge, there does not exists any formal security notion for computationally secure cheating identifiable multi secret sharing scheme for general access structure.

1.2 Our Contribution

In this paper, we introduce a formal notion of security for computationally secure cheating identifiable multi secret sharing scheme for general access structure and propose a multi secret sharing scheme which is secure under the proposed notion in random oracle model. In this context, it is worth mentioning that there is a simple way (see [13]) to construct computationally secure cheating identifiable secret sharing scheme from secret sharing scheme by using signature of the dealer on the shares and thereby preventing any tampering of shares. But, this technique is not applicable for multi-use multi-secret sharing schemes, as in multi-use multi secret sharing schemes, secrets are reconstructed with the help of pseudo-shares which are generated by the participants. As a result, dealer's signature may not be useful any more.

1.3 Organization of the Paper

In Section 2, we describe the adversarial model and communication model on which our construction and analysis are based. The detailed construction is given in Section 3 and its security analysis is done in Section 3.1. Finally we conclude in Section 4.

2 Model and Definition

In this section, we specify the adversarial and communication model used in the rest of the paper. We also propose formal definitions of construction and security of cheating identifiable multi-secret sharing scheme for general access structure.

Adversarial Model. The dealer $\mathcal{D}$ is assumed to be honest. The dealer delivers the shares to respective players over point-to-point private channels. We assume that $\mathcal{A}$ is computationally bounded and malicious. Once a player P is corrupted, the adversary learns his share and internal state. Moreover from that point onwards, $\mathcal{A}$ has full control over P. By being *malicious*, we mean that $\mathcal{A}$ can deviate from the protocol in an arbitrary manner.

Communication Model. We assume synchronous network model. There are point to point secure channels among the dealer and the players. Moreover, all them have an access of a common broadcast channel.

Definition 1. *A Cheating Identifiable Multi Secret Sharing Scheme (CI-MSSS) Ω consists of three probabilistic polynomial time algorithms* $(\mathsf{Setup}, \mathsf{Dist}, \mathsf{Reconst})$ *as follows:*

1. *The setup protocol,* Setup*, takes as input a security parameter $\lambda \in \mathbb{N}$, the set of players $\mathcal{P}$ and the k access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$, where $\Gamma_i =$*

$\{A_{i1}, A_{i2}, \ldots, A_{it_i}\}$ is the access structure for the ith secret and A_{ij} is the jth qualified subset of the access structure for the ith secret s_i, and outputs some public and common parameters pms *for the scheme (such as the access structures and set of players, mathematical groups, hash functions, etc.). We implicitly assume that* pms *also contains the descriptions of $\mathcal{P}$ and the access structures.*

2. *The share distribution protocol,* Dist*, (run by the dealer $\mathcal{D}$) takes as input* pms *and the global secret $\vec{\mathbf{s}} = (s_1, s_2, \ldots, s_k)$ to be distributed, and produces the set of shares $\{x_\alpha\}_{P_\alpha \in \mathcal{P}}$, possibly some public output* out_{pub} *and a set of public verification values $\mathcal{V} = \{V_{\varphi(x_\alpha, A_{ij})} : P_\alpha \in A_{ij} \in \Gamma_i\}$. (Note: $\varphi(x_\alpha, A_{ij})$ is a public function used to generate pseudo-shares from the share x_α and the qualified set A_{ij}.)*
3. *The secret reconstruction protocol,* $\mathsf{Reconst}$*, takes as input* $\mathsf{pms}, \mathsf{out}_{pub}$*, $\mathcal{V}$ and the possible pseudo-shares $\{\varphi^*_\alpha\}_{P_\alpha \in A_{ij}}$ of the players belonging to some subset $A_{ij} \in \Gamma_i$ and outputs either a possible value of the secret s^*_i for the i-th secret or a special symbol $\perp$ along with a list of cheating participants* $\mathsf{CheatList} = \{P_\alpha \in A_{ij} : V_{\varphi(x_\alpha, A_{ij})} \neq V_{\varphi^*_\alpha}\}$.

For correctness, we require that, for any index $i \in \{1, 2, \ldots, k\}$ and any subset $A_{ij} \in \Gamma_i$, it holds

$$\mathsf{Reconst}(\mathsf{pms}, \mathsf{out}_{pub}, \{\varphi(x_\alpha, A_{ij})\}_{P_\alpha \in A_{ij}}) = s_i$$

if $\{x_\alpha\}_{P_\alpha \in A_{ij}} \subset \{x_\alpha\}_{P_\alpha \in \mathcal{P}}$ and $(\mathsf{out}_{pub}, \{x_\alpha\}_{P_\alpha \in \mathcal{P}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \vec{\mathbf{s}})$ *is a distribution of the secret $\vec{\mathbf{s}} = (s_1, \ldots, s_i, \ldots s_k)$ and the setup protocol has produced* $\mathsf{pms} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{P}, \{\Gamma_i\}_{1 \le i \le k})$.

The computational security and cheating identifiablity of CI-MSSS Ω is defined by the games described in Definition 2 and Definition 3 respectively.

Definition 2. (Indistinguishability of Shares against Chosen Secret Attack). *Indistinguishability of shares of a CI-MSSS under chosen secret attack* **(IND-CSA)** *is defined by the following game $\mathcal{G}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:*

1. *The adversary $\mathcal{A}$ publishes the set of players $\mathcal{P}$ and the k access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k \subset 2^{\mathcal{P}}$.*
2. *The challenger $\mathcal{C}$ runs* $\mathsf{pms} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{P}, \{\Gamma_i\}_{1 \le i \le k})$ *and sends* pms *to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs a subset $\tilde{B} \subset \mathcal{P}$ of unqualified players (unqualified means $\exists i \in \{1, 2, \ldots, k\}$ such that $\tilde{B} \notin \Gamma_i$) and two different global secrets $\vec{\mathbf{s}}^{(0)} \neq \vec{\mathbf{s}}^{(1)}$ with the restriction:*

$$s_i^{(0)} = s_i^{(1)}, \forall i \in \{1, 2, \ldots, k\}, \text{ such that } \tilde{B} \in \Gamma_i.$$

4. *The challenger $\mathcal{C}$ chooses at random a bit $b \in_R \{0, 1\}$, runs* $\mathsf{Dist}(\mathsf{pms}, \vec{\mathbf{s}}^{(b)}) \rightarrow (\mathsf{out}_{pub}, \mathcal{V}, \{x_\alpha\}_{P_\alpha \in \mathcal{P}})$ *and sends* $\left(\mathsf{out}_{pub}, \mathcal{V}, \{x_\alpha\}_{P_\alpha \in \tilde{B}}\right)$ *to $\mathcal{A}$.*
5. *Finally, $\mathcal{A}$ outputs a bit b'.*

The advantage of $\mathcal{A}$ in breaking the CI-MSSS Ω is defined as $\mathsf{Adv}_{\mathcal{A}}(\lambda) = |\mathsf{Pr}[b' = b] - \frac{1}{2}|$.

The scheme Ω is said to be computationally IND-CSA secure if $\mathsf{Adv}_{\mathcal{A}}(\lambda)$ *is negligible for all polynomial-time adversaries $\mathcal{A}$.*

Definition 3. (Cheating Identifiability). *Cheating Identifiability of a CI-MSSS Ω is defined by the following game $\mathcal{G}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:*

1. *The adversary $\mathcal{A}$ chooses the set of players $\mathcal{P}$, a secret vector $\vec{\mathbf{s}} = (s_1, s_2, \ldots, s_k)$ and the corresponding k access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k \subset 2^{\mathcal{P}}$. Then $\mathcal{A}$ runs* $\mathsf{pms} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{P}, \{\Gamma_i\}_{1\le i\le k})$ *and sends $(pms, \vec{\mathbf{s}})$ to $\mathcal{C}$.*
2. *The challenger $\mathcal{C}$ runs* $\mathsf{Dist}(\mathsf{pms}, \vec{\mathbf{s}}) \rightarrow (\mathsf{out}_{pub}, \mathcal{V}, \{x_\alpha\}_{P_\alpha \in \mathcal{P}})$ *and sends* $(\mathsf{out}_{pub}, \mathcal{V}, \{x_\alpha\}_{P_\alpha \in \mathcal{P}})$ *to $\mathcal{A}$.*
3. *For each secret s_i, $\mathcal{A}$ outputs one qualified subset of Γ_i and a corresponding set of pseudo-shares (may or may not be honestly generated) of each participant in that qualified set, i.e.,*

$$\forall i \in \{1, 2, \ldots, k\}, \mathcal{A} \textit{ outputs some } A_{ij} \in \Gamma_i \textit{ and } \{\varphi^*_\alpha\}_{P_\alpha \in A_{ij}}$$

.

4. *The challenger $\mathcal{C}$ runs* $\forall i \in \{1, 2, \ldots, k\}$

$$\mathsf{Reconst}(\mathsf{pms}, \mathsf{out}_{pub}, \mathcal{V}, \{\varphi^*_\alpha\}_{P_\alpha \in A_{ij}}, A_{ij} \in \Gamma_i) \rightarrow Out_i,$$

where $Out_i = \begin{cases} s_i, & \textit{if } V_{\varphi(x_\alpha, A_{ij})} = V_{\varphi^*_\alpha}, \forall P_\alpha \in A_{ij} \\ \{\perp, \mathsf{CheatList} = \{P_\alpha \in A_{ij} : V_{\varphi(x_\alpha, A_{ij})} \neq V_{\varphi^*_\alpha}\}\}, & \textit{otherwise} \end{cases}$

5. *If for any $i \in \{1, 2, \ldots, k\}$, for some $P_\alpha \in A_{ij}$, $\varphi(x_\alpha, A_{ij}) \neq \varphi^*_\alpha$, but $V_{\varphi(x_\alpha, A_i)} = V_{\varphi^*_\alpha}$, i.e., $P_\alpha \notin$* $\mathsf{CheatList}$*, the challenger $\mathcal{C}$ sets $b = 1$, else sets $b = 0$. Finally, $\mathcal{C}$ outputs the bit b.*

The scheme Ω is said to be computationally cheating identifiable if $\mathsf{Pr}[b = 1]$ *is negligible for all polynomial-time adversaries $\mathcal{A}$.*

3 A Cheating Identifiable Multi-Secret Sharing Scheme

In this section, we modify the MSSS for general access structure proposed by [15] and analyse its security in the computational model of IND-CSA and cheating identifiability. (It is worth mentioning that the scheme in [15] lacked formal security analysis.) The scheme $\Omega = (\mathsf{Setup}, \mathsf{Dist}, \mathsf{Reconst})$ consists of three basic phases,

1. **Setup:** On a input security parameter λ, the set of n players or participants $\mathcal{P} = \{P_\alpha : \alpha \in \{1, 2, \ldots, n\}\}$ and k-access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$ for k secrets, where $\Gamma_i = \{A_{i1}, A_{i2}, \ldots, A_{it_i}\}$ is the access structure for the i-th secret and A_{ij} is the jth qualified subset of the access structure of ith secret s_i and $|A_{ij}| = r_{ij}$,

(a) Choose a $q = q(\lambda)$-bit prime p.
(b) Choose a hash function $H : \{0,1\}^{q+l+m} \to \mathbb{Z}_p \subseteq \{0,1\}^q$, where $l = [log_2 k] + 1, m = [log_2 t] + 1$ such that $t = max\{t_1, t_2, \dots, t_k\}$.
(c) Choose distinct identifier $ID_\alpha \in_R \mathbb{Z}_p^*$ corresponding to each of the participant $P_\alpha, \alpha \in \{1, 2, \dots, n\}$
(d) Choose a hash function $G : \{0,1\}^q \to \{0,1\}^{u(\lambda)}$. [This is needed for cheating identifiability.]
(e) Set as $\mathsf{pms} = (p, q, k, l, m, H, G, ID_\alpha, \mathcal{P}, \Gamma_1, \Gamma_2, \dots \Gamma_k)$.

2. **Dist:** On input $\mathsf{pms} = (p, q, k, l, m, H, ID_\alpha, \mathcal{P}, \Gamma_1, \Gamma_2, \dots \Gamma_k)$ and k secrets $s_1, s_2, \dots, s_k \in \mathbb{Z}_p \subseteq \{0,1\}^q$,
   (a) Choose $x_\alpha \in_R \{0,1\}^q$, $\alpha = 1, 2, \dots, n$.
   (b) For A_{ij} where $i = 1, 2, \dots, k; j = 1, 2, \dots, t_i$, choose $d_1^{ij}, d_2^{ij}, \dots, d_{r_{ij}-1}^{ij} \in_R \mathbb{Z}_p \subseteq \{0,1\}^q$ and set

$$f_{ij}(x) = s_i + d_1^{ij} x + d_2^{ij} x^2 + \cdots + d_{r_{ij}-1}^{ij} x^{r_{ij}-1}$$

   (c) For each $P_\alpha \in A_{ij}$, compute
   - $\varphi(x_\alpha, A_{ij}) = H(x_\alpha || i_l || j_m)$ where i_l denotes the l-bit binary representation of i, j_m denotes the m-bit binary representation of j and '$||$' denotes the concatenation of two binary strings.
   - $\mathcal{B}_{ij}^\alpha = f_{ij}(ID_\alpha)$ and $\mathcal{M}_{ij}^\alpha = \mathcal{B}_{ij}^\alpha - \varphi(x_\alpha, A_{ij})$.
   - the public verification values $V_{\varphi(x_\alpha, A_{ij})} = G(\varphi(x_\alpha, A_{ij}))$. [needed for cheating identifiability]

   (d) Output $\{x_\alpha\}_{1 \le \alpha \le n}$ as shares, $\mathsf{out}_{pub} = \{\mathcal{M}_{ij}^\alpha : P_\alpha \in A_{ij}, 1 \le i \le k; 1 \le j \le t_i\}$ as public output.
   (e) Output $\mathcal{V} = \{V_{\varphi(x_\alpha, A_{ij})} : P_\alpha \in A_{ij}, 1 \le i \le k; 1 \le j \le t_i\}$ as public verification value. [needed for cheating identifiability]

3. **Reconst:**
   (a) **Participant Phase:** On input $\mathsf{pms}, \mathsf{out}_{pub}$ and $A_{ij} \in \Gamma_i$, each participant $P_\alpha \in A_{ij}$ computes and broadcast $\varphi(x_\alpha, A_{ij}) = H(x_\alpha || i_l || j_m)$, $\forall A_{ij} \in \Gamma_i$.

   (b) **Verification Phase:** On input $\mathcal{V}$ and $\{\varphi(x_\alpha, A_{ij}) : \forall P_\alpha \in A_{ij} \in \Gamma_i\}$,

   - participants check $G(\varphi(x_\alpha, A_{ij}) \overset{?}{=} V_{\varphi(x_\alpha, A_{ij})}, \forall P_\alpha \in A_{ij}$.
   - compute $\mathsf{CheatList} = \{P_\alpha : G(\varphi(x_\alpha, A_{ij}) \neq V_{\varphi(x_\alpha, A_{ij})}\}$.

   (c) **Secret Reconstruction Phase:**

   - if $\mathsf{CheatList} = \emptyset$, then compute $f_{ij}(ID_\alpha) = \mathcal{B}_{ij}^\alpha = \mathcal{M}_{ij}^\alpha + \varphi(x_\alpha, A_{ij})$, $\forall P_\alpha \in A_{ij}$. Then compute and output s_i from $\{f_{ij}(ID_\alpha) : P_\alpha \in A_{ij}\}$ using Lagrange's Interpolation.
   - otherwise, output $\{\perp, \mathsf{CheatList}\}$.

3.1 Security Analysis of Ω

Theorem 1. *Ω satisfies correctness condition.*
Proof : As correctness is considerable only when all the participants are honest, it is obvious that, using Lagrange's Interpolation, every qualified set of honest participants can reconstruct corresponding secret. □

Theorem 2. *Ω is IND-CSA secure CI-MSSS in random oracle model.*
Proof : Let $\mathcal{A}_\Omega$ be an adversary against IND-CSA security of Ω. Let $\mathcal{C}$ be the challenger of the security game. $\mathcal{A}_\Omega$ starts the game by choosing a set of participants $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ and k access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$. $\mathcal{C}$ runs Setup of Ω to generate pms and send everything in pms except the hash functions G, H to $\mathcal{A}_\Omega$.

$\mathcal{A}_\Omega$ outputs a set $\tilde{B} \subset \mathcal{P}$ of corrupted players and two different global secrets $\vec{\mathbf{s}}^{(0)} \neq \vec{\mathbf{s}}^{(1)}$ with the restriction:

$$s_i^{(0)} = s_i^{(1)}, \forall i \in \{1, 2, \ldots, k\}, \text{ such that } \tilde{B} \in \Gamma_i.$$

$\mathcal{C}$ chooses pairwise distinct $x_\alpha \in_R \{0,1\}^q$, $\alpha = 1, 2, \ldots, n$.

Simulation of H-queries: $\mathcal{C}$ starts with two empty lists namely H-list and R-list. When $\mathcal{A}_\Omega$ submits a hash query of the form $x||i||j$ (In this proof, for simplicity, we write i_l, j_m as i, j only.), $\mathcal{C}$ checks whether $x = x_\alpha$ for some $P_\alpha \in \mathcal{P}$.

```
If $x \neq x_\alpha, \forall \alpha \in \{1, 2, \ldots, n\}$
do {
    Choose $\gamma \in_R \{0,1\}^q$
    Add $(x||i||j, \gamma)$ to the R-list
    Return $\gamma$.
}
If $x = x_\alpha$ for some $\alpha$,
If $x = x_\alpha$ & $P_\alpha \in \tilde{B}$,
do {
    If $P_\alpha \in A_{ij} \in \Gamma_i$
        Choose $h_{\alpha,i,j} \in_R \{0,1\}^q$.
        Add $(x_\alpha||i||j, h_{\alpha,i,j})$ to H-list
        Return $h_{\alpha,i,j}$.
    If $P_\alpha \notin A_{ij} \in \Gamma_i$
        Choose $\gamma \in_R \{0,1\}^q$.
        Add $(x_\alpha||i||j, \gamma)$ to the R-list
        Return $\gamma$.
}
If $x = x_\alpha$ & $P_\alpha \notin \tilde{B}$,
do {
    If $P_\alpha \in A_{ij} \in \Gamma_i$
        Choose $h_{\alpha,i,j} \in_R \{0,1\}^q$.
        Add $(x_\alpha||i||j, h_{\alpha,i,j})$ to H-list
        Return $h_{\alpha,i,j}$.
    If $P_\alpha \notin A_{ij} \in \Gamma_i$
        Choose $\gamma \in_R \{0,1\}^q$.
        Add $(x_\alpha||i||j, \gamma)$ to the R-list
        Return $\gamma$.
}
```

If a hash query $x||i||j$ by $\mathcal{A}_\Omega$ is already in H or R-list, the stored value is sent back to $\mathcal{A}_\Omega$. It is to be noted that the entries in R-list are not required in the actual execution of the MSSS, whereas H-list will be used by the challenger $\mathcal{C}$ to simulate the out_{pub}.

Simulation of G-queries: $\mathcal{C}$ starts with two empty lists namely G-list and G'-list. When $\mathcal{A}_\Omega$ submits a hash query of the form h^*, $\mathcal{C}$ checks whether $h^* = h_{\alpha,i,j}$ for some $h^* \in H$-list.

$$\begin{array}{l||l} \text{If } h^* = h_{\alpha,i,j} \in H\text{-list}, & \text{If } h^* \notin H\text{-list}, \\ \text{do} \begin{cases} \text{Choose } V_{\alpha,i,j} \in_R \{0,1\}^u. \\ \text{Add } (h_{\alpha,i,j}, V_{\alpha,i,j}) \text{ to G-list} \\ \text{Return } V_{\alpha,i,j}. \end{cases} & \text{do} \begin{cases} \text{Choose } \eta \in_R \{0,1\}^u. \\ \text{Add } (h^*, \eta) \text{ to G'-list} \\ \text{Return } \eta. \end{cases} \end{array}$$

If a hash query h^* by $\mathcal{A}_\Omega$ is already in G or G'-list, the stored value is sent back to $\mathcal{A}_\Omega$. It is to be noted that it may happen that $\mathcal{A}_\Omega$ queries the hash function G with h^* such that at that stage $h^* \notin H$-list, but h^* was latter added to the H-list as some $h_{\alpha,i,j}$. In that case, the entry (h^*, η) is shifted from G'-list to G-list and renamed as $(h_{\alpha,i,j}, V_{\alpha,i,j})$. Observe that the entries in the final G'-list are not required in the actual execution of Dist algorithm. Only the entries in G-list are used by the challenger $\mathcal{C}$ to simulate the $\mathcal{V}$.

$\mathcal{C}$ chooses a bit $b \in_R \{0,1\}$ and do the following:

- $\forall A_{ij} \in \Gamma_i$ where $i = 1, 2, \ldots, k; j = 1, 2, \ldots, t_i$, choose $d_1^{ij}, d_2^{ij}, \ldots, d_{r_{ij}-1}^{ij} \in_R \mathbb{Z}_p \subseteq \{0,1\}^q$ and set
$$f_{ij}(x) = s_i + d_1^{ij} x + d_2^{ij} x^2 + \cdots + d_{r_{ij}-1}^{ij} x^{r_{ij}-1}$$
- For each $P_\alpha \in A_{ij}$, compute $\mathcal{B}_{ij}^\alpha = f_{ij}(ID_\alpha)$, $\mathcal{M}_{ij}^\alpha = \mathcal{B}_{ij}^\alpha - h_{\alpha,i,j}$.

The values of $h_{\alpha,i,j}$ are either recollected from H-list, if they exist, or they are chosen randomly from $\{0,1\}^q$. In the latter case, the entry is added to the H-list for answering further hash queries. Moreover, $\mathcal{C}$ generates a simulated set $\mathcal{V} = \{V_{\alpha,i,j} : P_\alpha \in A_{ij} \in \Gamma_i\}$ where $V_{\alpha,i,j}$'s are either collected from G-list, if they exists, or randomly chosen from $\{0,1\}^u$ and added in the G-list.

$\mathcal{C}$ returns the public output $\mathsf{out}_{pub} = \{\mathcal{M}_{ij}^\alpha : P_\alpha \in A_{ij}, 1 \leq i \leq k; 1 \leq j \leq t_i\}$, $\mathcal{V} = \{V_{\alpha,i,j} : P_\alpha \in A_{ij} \in \Gamma_i\}$ and the shares $\{x_\alpha : P_\alpha \in \tilde{B}\}$ of the corrupted participants to $\mathcal{A}_\Omega$. Finally, $\mathcal{A}_\Omega$ outputs its guess b' for b.

Therefore, to compute the probability that $\mathcal{A}_\Omega$ outputs the correct bit, we distinguish between two cases, depending on whether $\mathcal{A}_\Omega$ somehow manages to get the pseudo-share $h_{\alpha,i,j}$ for some non-corrupted participant $P_\alpha \notin \tilde{B}$ and $P_\alpha \in A_{ij} \in \Gamma_i$ or not. If $\mathcal{A}_\Omega$ gets $h_{\alpha,i,j}$ for some $P_\alpha \notin \tilde{B}$, say with probability δ, this is the best case for $\mathcal{A}_\Omega$ and he can correctly guess the secret bit. On the other hand, if $\mathcal{A}_\Omega$ is not able to output any pseudo-share corresponding to a non-corrupted participant, which happens with probability $1 - \delta$, then the probability of $\mathcal{A}_\Omega$ guessing the correct bit is exactly 1/2. Hence, in any case, the probability of $\mathcal{A}_\Omega$ guessing the correct bit is $\delta + \frac{1}{2}(1-\delta) = \frac{\delta}{2} + \frac{1}{2}$ i.e., $\mathsf{Adv}_{\mathcal{A}_\Omega}(\lambda) = |(\frac{\delta}{2} + \frac{1}{2}) - \frac{1}{2}| = \frac{1}{2}\delta$.

Now, let E_1 be the event that $\mathcal{A}_\Omega$ makes a hash query $x_\alpha||i||j$, where x_α is the share of $P_\alpha \in \mathcal{P} \setminus \tilde{B}$ and $P_\alpha \in A_{ij} \in \Gamma_i$ and $|\tilde{B}| = \tilde{t}$. The probability that a single H query leads to E_1 is $\dfrac{n-\tilde{t}}{2^q - \tilde{t}}$. Now, taking Q_H to be the total number of H-queries, we get

$$\mathsf{Pr}[E_1] = 1 - \left(1 - \frac{n-\tilde{t}}{2^q - \tilde{t}}\right)\left(1 - \frac{n-\tilde{t}}{2^q - \tilde{t} - 1}\right) \cdots \left(1 - \frac{n-\tilde{t}}{2^q - \tilde{t} - Q_H + 1}\right)$$

$$\leq 1 - \left(1 - \frac{n - \tilde{t}}{2^q - \tilde{t}}\right)^{Q_H} \approx \frac{Q_H(n - \tilde{t})}{2^q - \tilde{t}} \leq \frac{n \cdot Q_H}{2^q - \tilde{t}} \approx \frac{n \cdot Q_H}{2^q}$$

as $\tilde{t}, Q_H$ are negligible compared to 2^q. Let E_2 be the event that $\mathcal{A}_\Omega$ guesses the $h_{\alpha,i,j}$ for some $P_\alpha \notin \tilde{B}$ and $P_\alpha \in A_{ij} \in \Gamma_i$ from the publicly available $V_{\alpha,i,j}$. Since, $V_{\alpha,i,j}$ is randomly chosen and letting Q_G to be the total number of G-queries, we get,

$$\mathsf{Pr}[E_2] = 1 - \left(1 - \frac{1}{2^u}\right)^{Q_G} \approx \frac{Q_G}{2^u}$$

Now, $\delta = \mathsf{Pr}[E_1 \cup E_2] \leq \mathsf{Pr}[E_1] + \mathsf{Pr}[E_2] \approx \frac{n \cdot Q_H}{2^q} + \frac{Q_G}{2^u}$. Thus,

$$\mathsf{Adv}_{\mathcal{A}_\Omega}(\lambda) \approx \frac{1}{2}\left(\frac{n \cdot Q_H}{2^q} + \frac{Q_G}{2^u}\right).$$

□

Theorem 3. *Ω is cheating identifiable, if G is collision resistant.*

Proof : The adversary $\mathcal{A}$ chooses the set of players $\mathcal{P}$, a secret vector $\vec{\mathsf{s}} = (s_1, s_2, \ldots, s_k)$ and the corresponding k access structures $\Gamma_1, \Gamma_2, \ldots, \Gamma_k \subset 2^{\mathcal{P}}$. Then $\mathcal{A}$ runs $\mathsf{Setup}(1^\lambda, \mathcal{P}, \{\Gamma_i\}_{1 \leq i \leq k}) \to \mathsf{pms} = (p, q, k, l, m, H, G, ID_\alpha)$ and sends $(\mathsf{pms}, \vec{\mathsf{s}})$ to $\mathcal{C}$. The challenger $\mathcal{C}$ runs $\mathsf{Dist}(\mathsf{pms}, \vec{\mathsf{s}})$ to output the shares $\{x_\alpha\}_{P_\alpha \in \mathcal{P}}$, public outputs $\mathsf{out}_{pub} = \{\mathcal{M}_{ij}^\alpha : P_\alpha \in A_{ij}, 1 \leq i \leq k; 1 \leq j \leq t_i\}$ and public verification value $\mathcal{V} = \{V_{\varphi(x_\alpha, A_{ij})} : P_\alpha \in A_{ij}, 1 \leq i \leq k; 1 \leq j \leq t_i\}$ and sends $(\mathsf{out}_{pub}, \mathcal{V}, \{x_\alpha\}_{P_\alpha \in \mathcal{P}})$ to $\mathcal{A}$.

For each secret s_i, $\mathcal{A}$ outputs one qualified subset of Γ_i and a corresponding set of pseudo-shares (may or may not be honestly generated) of each participant in that qualified set, i.e.,

$$\forall i \in \{1, 2, \ldots, k\}, \mathcal{A} \text{ outputs some } A_{ij} \in \Gamma_i \text{ and } \{\varphi_\alpha^*\}_{P_\alpha \in A_{ij}}.$$

Finally, the challenger $\mathcal{C}$ runs $\forall i \in \{1, 2, \ldots, k\}$

$$\mathsf{Reconst}(\mathsf{pms}, \mathsf{out}_{pub}, \mathcal{V}, \{\varphi_\alpha^*\}_{P_\alpha \in A_{ij}}, A_{ij} \in \Gamma_i) \to Out_i,$$

where $Out_i = \begin{cases} s_i, & \text{if } V_{\varphi(x_\alpha, A_{ij})} = V_{\varphi_\alpha^*}, \forall P_\alpha \in A_{ij} \\ \{\bot, \mathsf{CheatList} = \{P_\alpha \in A_{ij} : V_{\varphi(x_\alpha, A_{ij})} \neq V_{\varphi_\alpha^*}\}\}, & \text{otherwise} \end{cases}$

Now, let us consider the case when $\mathcal{A}$ wins the game i.e., when $\mathcal{C}$ outputs $b = 1$. Note that $b = 1 \Rightarrow \exists$ at least one $i \in \{1, 2, \ldots, k\}$ such that $\exists P_\alpha \in A_{ij}$ with $\varphi(x_\alpha, A_{ij}) \neq \varphi_\alpha^*$, but $V_{\varphi(x_\alpha, A_i)} = V_{\varphi_\alpha^*}$, i.e.,

$$\varphi(x_\alpha, A_{ij}) \neq \varphi_\alpha^* \text{ but } G(\varphi(x_\alpha, A_{ij})) = G(\varphi_\alpha^*),$$

i.e., we find a collision for G.

Let us denote the event of finding collision for G by Col_G and let $\mathsf{Pr}[\mathsf{Col}_G] = \delta_G$. Thus, the adversary wins the game if Col_G occurs, i.e., $\mathsf{Pr}[b = 1] \leq \delta_G$.

Since, G is collision resistant, δ_G is negligible and as a result, $\mathsf{Pr}[b = 1]$ is negligible. □

4 Conclusion

In this paper, the notion of computational cheating identifiability for multi-secret sharing schemes for general access structure is established. We also provide construction and proofs of security of a cheating identifiable MSSS for general access structure. As a topic of future research, one can think of more efficient construction of cheating identifiable multi-secret sharing schemes for general access structure.

References

1. Bellare, M., Rogaway, P.: Robust computational secret sharing and a unified account of classical secret-sharing goals. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 172–184 (2007)
2. Blakley, G.R.: Safeguarding cryptographic keys. In: The National Computer Conference 1979. AFIPS, vol. 48, pp. 313–317 (1979)
3. Damgård, I., Jakobsen, T.P., Nielsen, J.B., Pagter, J.I.: Secure key management in the cloud. In: Stam, M. (ed.) IMACC 2013. LNCS, vol. 8308, pp. 270–289. Springer, Heidelberg (2013)
4. Das, A., Adhikari, A.: An efficient multi-use multi-secret sharing scheme based on hash function. Applied Mathematics Letters 23(9), 993–996 (2010)
5. Dehkordi, M.H., Mashhadi, S.: An efficient threshold verifiable multi-secret sharing. Computer Standards and Interfaces 30, 187–190 (2008)
6. Geng, Y.J., Fan, X.H., Hong, F.: A new multi-secret sharing scheme with multi-policy. In: The 9th International Conference on Advanced Communication Technology, vol. 3, pp. 1515–1517 (2007)
7. He, J., Dawson, E.: Multi-secret sharing scheme based on one-way function. Electronic Letters 31(2), 93–95 (1994)
8. He, J., Dawson, E.: Multi-stage secret sharing based on one-way function. Electronic Letters 30(19), 1591–1592 (1994)
9. Herranz, J., Ruiz, A., Saez, G.: New results and applications for multi-secret sharing schemes. In: Design, Codes and Cryptography, pp. 1–24. Springer (2013)
10. Herranz, J., Ruiz, A., Saez, G.: Sharing many secrets with computational provable security. Information Processing Letters 113, 572–579 (2013)
11. Huang, Z., Li, Q., Wei, R., Li, Z.: A Generalized Multi-secret Sharing Scheme to Identify Cheaters. Journal of the China Railway Society (July 2006)
12. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: Identifying cheaters without an honest majority. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 21–38. Springer, Heidelberg (2012)
13. Martin, K.M.: Challenging the adversary model in secret sharing schemes. Coding and Cryptography II. In: Proceedings of the Royal Flemish Academy of Belgium for Science and the Arts, pp. 45–63 (2008)
14. McEliece, R., Sarwate, D.: On sharing secrets and reed-solomon codes. Communications of the ACM 24(9), 583–584 (1981)
15. Roy, P.S., Adhikari, A.: Multi-Use Multi-Secret Sharing Scheme for General Access Structure. Annals of the University of Craiova, Mathematics and Computer Science Series 37(4), 50–57 (2010)
16. Shamir, A.: How to share a secret. Communications of the ACM 22, 612–613 (1979)