Almost Optimum Secret Sharing Schemes Secure against Cheating for Arbitrary Secret Distribution

Satoshi Obana and Toshinori Araki

NEC Corporation
{obana@bx,t-araki@ek}.jp.nec.com

Abstract. We consider the problem of cheating in secret sharing schemes, cheating in which individuals submit forged shares in the secret reconstruction phase in an effort to make another participant reconstruct an invalid secret. We introduce a novel technique which uses universal hash functions to detect such cheating and propose two efficient secret sharing schemes that employ the functions. The first scheme is nearly optimum with respect to the size of shares; that is, the size of shares is only one bit longer than its existing lower bound. The second scheme possesses a particular merit in that the parameter for the probability of successful cheating can be chosen without regard to the size of the secret. Further, the proposed schemes are proven to be secure regardless of the probability distribution of the secret.

1 Introduction

A secret sharing scheme is a cryptographic primitive in which a secret is divided into shares and distributed among participants in such a way that only a qualified set of participants can recover the secret. It is a fundamental building block for many cryptographic protocols and is often used in the general composition of secure multiparty computations. While seminal papers were presented by Shamir [10] and Blakley [1] more than a quarter century ago, because of its importance in cryptography, it is still being studied actively today.

Tompa and Woll have pointed out that in Shamir's k-out-of-n threshold secret sharing scheme, even a single user can fool other participants by submitting invalid shares at the secret reconstruction phase. They also proposed a scheme which can detect the fact of cheating when invalid shares are submitted at that point. Ogata, Kurosawa and Stinson also have presented an efficient scheme for detecting cheating [8]. While the size of shares in their scheme is proven to be optimum, the scheme is proven to be secure only if the secret is uniformly distributed, and the size of the secret will restrict possible value for the successful cheating probability.

In this paper, we propose two efficient k-out-of-n threshold secret sharing schemes which are secure regardless of the probability distribution of the secret. The first scheme is nearly optimum with respect to the size of shares; that is, the size of shares is only one bit longer than its existing bound. In the second scheme, the size of shares is somewhat larger than the first scheme, but the second scheme possesses a particular merit in that the successful cheating probability can be chosen without regard to the size of the secret. This is not the case in either the first scheme or the scheme in [8]. The size of shares in the second scheme is much smaller than that in the scheme by Tompa and Woll, which is also secure for arbitrary secret distribution and whose successful cheating probability can be also chosen without regard to the size of the secret. The size of shares in the second scheme will be even smaller than that in [8] when $\epsilon > |\mathcal{S}|^{-1/2}$, where ϵ denotes the successful cheating probability and $\mathcal{S}$ denotes the set of secrets[1]. This interesting phenomenon results from inflexibility of parameter values in [8]. Note that the condition $\epsilon > |\mathcal{S}|^{-1/2}$ is quite reasonable since ϵ is usually required to be 2^{-128} or 2^{-256}, whereas the the size of the secret can be as large as $|\mathcal{S}| = 2^{1024}$ or more.

The main idea of the proposed schemes is to use universal hash functions (more precisely, a variant of ASU_2, an *almost strongly universal* class of hash functions) for cheating detection. Here, the key for the universal hash functions is distributedly shared together with the share of the secret. In reconstructing the secret, both the secret and the key are reconstructed, and each participant verifies that the secret and the hash value are consistent. We additionally provide some techniques to reduce the size of shares and to prevent the hash value from revealing any information about the secret.

The rest of the paper is organized as follows. In Section 2, we briefly review models of secret sharing schemes capable of detecting cheating, and we discuss previous works done on them. In Section 3, we introduce a novel technique for detecting cheating via a universal hash family, and we present efficient schemes based on it. In Section 4, we describe two generalizations of the schemes presented in Section 3. In Section 5, we introduce new models which deal with more powerful cheaters than those in existing models, and we present schemes secure in the new models. In Section 6, we summarize our work.

2 Preliminaries

2.1 Secret Sharing Schemes

In secret sharing schemes, there are n participants $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a dealer D. The set of participants who are allowed to reconstruct the secret is characterized by an *access structure* $\Gamma \subseteq 2^{\mathcal{P}}$; that is, participants $P_{i_1}, \ldots, P_{i_k}$ are allowed to reconstruct the secret if and only if $\{P_{i_1}, \ldots, P_{i_k}\} \in \Gamma$ (for instance, the access structure of a k-out-of-n threshold secret sharing scheme is defined by $\Gamma = \{\mathcal{A} \mid \mathcal{A} \in 2^{\mathcal{P}}, |\mathcal{A}| \geq k\}$.) A model consists of two algorithms: $\mathsf{ShareGen}$ and $\mathsf{Reconst}$. Share generation algorithm $\mathsf{ShareGen}$ takes a secret $s \in \mathcal{S}$ as input and outputs a list $(v_1, v_2, \ldots, v_n)$. Each $v_i \in \mathcal{V}_i$ is called a *share* and is given to a participant P_i. Ordinarily, $\mathsf{ShareGen}$ is invoked by the dealer. Secret reconstruction algorithm $\mathsf{Reconst}$ takes a list of shares and outputs a secret $s \in \mathcal{S}$.

[1] Throughout the paper, the cardinality of the set $\mathcal{X}$ is denoted by $|\mathcal{X}|$.

A secret sharing scheme is called *perfect* if the following two conditions are satisfied for the output $(v_1, \ldots, v_n)$ of $\mathsf{ShareGen}(\hat{s})$ where the probabilities are taken over the random tape of $\mathsf{ShareGen}$.

1. if $\{P_{i_1}, \ldots, P_{i_k}\} \in \Gamma$ then $\Pr[\mathsf{Reconst}(v_{i_1}, \ldots, v_{i_k}) = \hat{s}] = 1$,
2. if $\{P_{i_1}, \ldots, P_{i_k}\} \notin \Gamma$ then $\Pr[\mathcal{S} = s \mid \mathcal{V}_{i_1} = v_{i_1}, \ldots, \mathcal{V}_{i_k} = v_{i_k}] = \Pr[\mathcal{S} = s]$ for any $s \in \mathcal{S}$.

2.2 Secret Sharing Schemes Secure against Cheating

A secret sharing schemes capable of detecting cheating was first presented by Tompa and Woll [12]. They considered the scenario in which cheaters who do not belong to the access structure submit forged shares in the secret reconstruction phase. Such cheaters will succeed if another participants in the reconstruction accepts an incorrect secret[2]. There are two different models for secret sharing schemes capable of detecting such cheating. Carpentieri, De Santis and Vaccaro [3] first considered a model in which cheaters who *know* the secret try to make another participant reconstruct an invalid secret. We call this model the *"CDV model."* Recently, Ogata, Kurosawa and Stinson [8] introduced a model with weaker cheaters who *do not* know the secret in forging their shares. We call this model the *"OKS model."*

As in ordinary secret sharing schemes, each of these models consists of two algorithms. A share generation algorithm ShareGen is the same as that in the ordinary secret sharing schemes. A secret reconstruction algorithm Reconst is slightly changed: it takes a list of shares as input and outputs either a secret or the special symbol $\perp$ ($\perp \notin \mathcal{S}$.) Reconst outputs $\perp$ if and only if cheating has been detected. To formalize the models, we define the following simple game for any (k, n) threshold secret sharing scheme $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ and for any (not necessarily polynomially bounded) Turing machine $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$, where A represents cheaters $P_{i_1}, \ldots, P_{i_{k-1}}$ who try to cheat P_{i_k}. Please note that in this section and the next we will focus on the (k, n) threshold type access structure. A more general access structure will be discussed in Section 4.

```
Game(SS, A)
    $s \leftarrow \mathcal{S}$;    // according to the probability distribution over $\mathcal{S}$.
    $(v_1, \ldots, v_n) \leftarrow \mathsf{ShareGen}(s)$;
    $(i_i, \ldots, i_{k-1}) \leftarrow \mathsf{A}_1(X)$;
    // set $X = s$ for the CDV model, $X = \emptyset$ for the OKS model.
    $(v'_{i_1}, \ldots, v'_{i_{k-1}}, i_k) \leftarrow \mathsf{A}_2(v_{i_1}, \ldots, v_{i_{k-1}}, X)$;
```

The advantage of cheaters is expressed as $Adv(\mathbf{SS}, \mathsf{A}) = \Pr[s' \in \mathcal{S} \wedge s' \neq s]$, where $s' = \mathsf{Reconst}(v'_{i_1}, v'_{i_2}, \ldots, v'_{i_{k-1}}, v_{i_k})$ and the probability is taken over the distribution of $\mathcal{S}$, and over the random tapes of ShareGen and A.

[2] Please note that here we focus on the problem of *detecting* the fact of cheating with unconditional security. Neither secret sharing schemes which *identify* cheaters [2, 6] nor *verifiable secret sharing schemes* [9, 4] are within the scope of this paper.

Definition 1. *A (k, n) threshold secret sharing scheme* $\mathbf{SS}$ *is called a* (k, n, ϵ)-secure secret sharing scheme *if* $Adv(\mathbf{SS}, \mathsf{A}) \leq \epsilon$ *for any adversary* A.

2.3 Previous Work

In this subsection, we briefly review the known bounds and constructions of (k, n, ϵ)-secure secret sharing schemes. A lower bound for the size of shares in the CDV model is described as follows:

Proposition 1. [3] *In the CDV model, the size of shares for* $(k, n, \epsilon_{\mathsf{CDV}})$*-secure secret sharing schemes is lower bounded by* $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|}{\epsilon_{\mathsf{CDV}}}$.

Ogata *et al.* improved this bound when the secret is uniformly distributed:

Proposition 2. [8] *In the CDV model, if the secret is uniformly distributed, then the size of shares* $|\mathcal{V}_i|$ *for* $(k, n, \epsilon_{\mathsf{CDV}})$*-secure secret sharing schemes is lower bounded by* $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|-1}{\epsilon^2_{\mathsf{CDV}}} + 1$.

Ogata *et al.* also presented the lower bound for the size of shares for $(k, n, \epsilon_{\mathsf{OKS}})$-secure secret sharing scheme in the OKS model as follows.

Proposition 3. [8] *In the OKS model, the size of shares for* $(k, n, \epsilon_{\mathsf{OKS}})$*-secure secret sharing schemes is lower bounded by* $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|-1}{\epsilon_{\mathsf{OKS}}} + 1$.

The following corollary may be seen to be straightforward from Proposition 2 since it has to hold for a uniformly distributed secret.

Corollary 1. *In the CDV model, the size of shares for* $(k, n, \epsilon_{\mathsf{CDV}})$*-secure secret sharing schemes which satisfy the following two conditions is lower bounded by* $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|-1}{\epsilon^2_{\mathsf{CDV}}} + 1$*. (1) Successful cheating probability is upper bounded by* ϵ *regardless of the probability distribution of the secret. (2) Share generation is independent of the secret distribution (i.e.* ShareGen *does not need to know the secret distribution.)*

Because it is in general difficult to determine exact probability distributions, we do not consider here situations in which the share generation algorithm *knows* the secret distribution and shares are generated according to the distribution[3].

Within the OKS model, Ogata *et al.* have proposed an elegant $(k, n, \epsilon_{\mathsf{OKS}})$-secure secret sharing schemes that satisfies the bound of Proposition 3 with equality [8]. The construction is summarized by the following proposition (please refer to [7] for the definition of *difference set.*)

Proposition 4. [8] *If there exists an* (N, ℓ, λ) *difference set then there exists a* $(k, n, \epsilon_{\mathsf{OKS}})$*-secure secret sharing scheme in the OKS model which satisfies the lower bound of Proposition 3 with equality. The scheme is secure if the secret is uniformly distributed.*

[3] As mentioned in [8], an example exists in which the size of shares is smaller than the bound of Proposition 2 when the secret is not uniformly distributed and shares are generated according to the distribution.

However, there are two drawbacks in the scheme of [8]. The first is that the scheme is proven to be secure only if the secret is uniformly distributed. This drawback comes from the property of the scheme that the share of the target participant can be uniquely determined from the shares of $k-1$ cheaters and the secret. Therefore, if there exists a secret which occurs with high probability then cheaters can guess the share of the target participant also with high probability, which causes the successful cheating probability larger than what is expected when the secret is uniformly distributed. The second drawback is that the successful cheating probability is uniquely determined from the size of the secret; that is, ϵ_{OKS} is determined to be $\epsilon_{\mathsf{OKS}} = 1/|\mathcal{S}|$ in [8]. On the other hand, the scheme by Tompa and Woll [12] which is secure in the CDV model is proven to be secure for arbitrary secret distribution and the successful cheating probability can be chosen without regard to the size of the secret. However, the size of shares is as large as $|\mathcal{V}_i| = (\frac{(|\mathcal{S}|-1)(k-1)}{\epsilon_{\mathsf{CDV}}} + k)^2$.

3 Proposed Schemes

In this section, we propose two efficient $(k, n, \epsilon_{\mathsf{CDV}})$-secure secret sharing schemes in the CDV model which are proven to be secure for any secret distribution. The first scheme is nearly optimum with respect to the size of shares; that is, the size of shares is $|\mathcal{V}_i| = |\mathcal{S}|/\epsilon_{\mathsf{CDV}}^2$ which is only one bit longer than the bound of Corollary 1. The size of shares in the second scheme is $|\mathcal{V}_i| = |\mathcal{S}|(\log|\mathcal{S}|)^2/\epsilon_{\mathsf{CDV}}^2$. Though the size of share is larger than the first scheme, the second scheme possesses a particular merit in that the size of the secret and the successful cheating probability can be chosen independently.

The underlying (and yet naive) idea of the schemes is to use *almost strongly universal hash functions* ϵ_{CDV}-ASU_2 for cheating detection. A family of hash functions $H : \mathcal{A} \to \mathcal{B}$ with the properties (1) and (2) below is called an ϵ-ASU_2. (1) For any $x \in \mathcal{A}$ and $y \in \mathcal{B}$, $|\{h_e \in H \mid h_e(x) = y\}| = |H|/|\mathcal{B}|$. (2) For any $x_1, x_2 (\neq x_1) \in \mathcal{A}$ and $y_1, y_2 \in \mathcal{B}$, $\{h_e \in H \mid h_e(x_1) = y_1, h_e(x_2) = y_2\}| = \epsilon|H|/|\mathcal{B}|$. where h_e denotes the element of H indexed by the *key* $e \in \mathcal{E}$ (clearly $|H| = |\mathcal{E}|$ holds.)

Now, consider the secret sharing scheme in which a randomly chosen key $e \in \mathcal{E}$ of H (where $H : \mathcal{S} \to \mathcal{B}$ is ϵ_{CDV}-ASU_2) is shared as well as the secret $s \in \mathcal{S}$ using the Shamir's (k, n) threshold secret sharing scheme and hash value $b = h_e(s)$ is open to the public. In the reconstruction phase, a secret $\hat{s}$ and a key $\hat{e}$ are reconstructed and $\mathsf{Reconst}$ outputs $\hat{s}$ as the valid secret if and only if $h_{\hat{e}}(\hat{s}) = b$ holds. Intuitively, the scheme seems to be $(k, n, \epsilon_{\mathsf{CDV}})$-secure in the CDV model since knowledge of the secret s does not help cheaters to compute $\hat{s}(\neq s)$ such that $h_{\hat{e}}(\hat{s}) = b$ with probability better than ϵ_{CDV}.

However, we must be careful about the following problems. The first problem is that the key $\hat{e} \in \mathcal{E}$ reconstructed from the shares is not always same as the original one since cheaters can forge the shares of the key for the hash functions. Therefore, we cannot prove the security of the above scheme directly from the properties of ϵ-ASU_2. The second problem is that public (and unforgeable) storage to store the hash value $b = h_e(s)$ is not always available. If the public storage is not available then the hash value has to be included in the share of each participant, which makes the size of shares larger. Further, we must ensure that the hash value $b = h_e(s)$ does not reveal any information about the secret since the scheme is no longer perfect if it is not the case. To overcome the first problem, we choose the specific ϵ-ASU_2 which can ensure security even when the key for the hash function is forged[4]. To overcome the second and the third problem, we fix the hash value $b = h_e(s)$ to be the constant (e.g. 0,) by which we can eliminate the public storage or additional shares without any loss of security.

We use two families of hash functions to construct the schemes. The first scheme is based on the well known $\frac{1}{p}$-ASU_2 such that $H = \{h_{e_0,e_1} \mid h_{e_0,e_1}(s) = e_0 - s \cdot e_1,\ e_i \in \mathit{GF}(p)\}$ (e.g. [11].) The second scheme is generalization of the first scheme and is based on the hash family $H = \{h_{e_0,e_1} \mid h_{e_0,e_1}(s_1, \ldots, s_N) = e_0 - \sum_{j=1}^{N} s_j \cdot e_1^j,\ e_i \in \mathit{GF}(p)\}$ which is proven to be $\frac{N}{p}$-ASU_2 [5].

3.1 Almost Optimum Scheme

The share generation algorithm ShareGen and the share reconstruction algorithm Reconst of the first scheme is described as follows where p is a prime power.

Share Generation: On input a secret $s \in \mathit{GF}(p)$, the share generation algorithm ShareGen outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Choose random $e_0, e_1 \in \mathit{GF}(p)$ such that $e_0 - s \cdot e_1 = 0$.
2. Generate random polynomials $f_s(x)$, $f_{e_0}(x)$, $f_{e_1}(x) \in \mathit{GF}(p)[X]$ of degree $k-1$ such that $f_s(0) = s$, $f_{e_0}(0) = e_0$ and $f_{e_1}(0) = e_1$.
3. Compute $v_i = (f_s(i), f_{e_0}(i), f_{e_1}(i))$ and output $(v_1, \ldots, v_n)$.

Secret Reconstruction and Validity Check: On input a list of k shares $(v_{i_1}, \ldots, v_{i_k})$, the secret reconstruction algorithm Reconst outputs a secret s or $\perp$ as follows:

1. Reconstruct $\hat{s}, \hat{e}_0$ and $\hat{e}_1$ from $v_{i_1}, \ldots, v_{i_k}$ using Lagrange interpolation.
2. Output s if $\hat{e}_0 - \hat{s} \cdot \hat{e}_1 = 0$ holds. Otherwise Reconst outputs $\perp$.

The properties of the first scheme is summarized by the following theorem.

Theorem 1. *The scheme of §3.1 is (k, n, ϵ)-secure secret sharing schemes in the CDV model with parameters $|\mathcal{S}| = p, \epsilon = 1/p$ and $|\mathcal{V}_i| = p^3$ $(= |\mathcal{S}|/\epsilon^2)$. Further, the scheme is secure for arbitrary secret distribution.*

The size of shares in the first scheme is only one bit longer than the lower bound of Proposition 2 since $\frac{|\mathcal{S}|}{\epsilon^2} < 2(\frac{|\mathcal{S}|-1}{\epsilon^2} + 1)$ holds for $|\mathcal{S}| \geq 2$.

3.2 A Scheme with Flexible Parameters

In the first scheme, the successful cheating probability is uniquely determined from the size of the secret. On the other hand, the successful cheating probability can be chosen without regard to the size of the secret in the second scheme. The second scheme can be described as follows.

[4] Formal requirements for the family of hash functions are given in Section 4.

Share Generation: On input a secret $s = (s_1, \dots, s_N) \in GF(p)^N$, the share generation algorithm $\mathsf{ShareGen}$ outputs a list of shares $(v_1, \dots, v_n)$ according to the following procedure. Please note that we sometimes regard $s = (s_1, \dots, s_N)$ as an element of $GF(p^N)$ instead of $GF(p)^N$.

1. Choose random $e_0, e_1 \in GF(p)$ such that $e_0 - \sum_{j=1}^{N} s_j e_1^j = 0$.
2. Generate a random polynomials $f_s(x) \in GF(p^N)[X]$ and $f_{e_0}(x)$, $f_{e_1}(x) \in GF(p)[X]$ of degree $k-1$ such that $f_s(0) = s$, $f_{e_0}(0) = e_0$ and $f_{e_1}(0) = e_1$.
3. Compute $v_i = (f_s(i), f_{e_0}(i), f_{e_1}(i))$ and output $(v_1, \dots, v_n)$.

Secret Reconstruction and Validity Check: On input a list of k shares $(v_{i_1}, \dots, v_{i_k})$, the secret reconstruction algorithm $\mathsf{Reconst}$ outputs a secret s or $\perp$ as follows:

1. Reconstruct $\hat{s}, \hat{e}_0$ and $\hat{e}_1$ from $v_{i_1}, \dots, v_{i_k}$ using Lagrange interpolation.
2. Output s if $\hat{e}_0 - \sum_{j=1}^{N} \hat{s}_j \hat{e}_1^j = 0$ holds. Otherwise $\mathsf{Reconst}$ outputs $\perp$.

The following theorem holds for the second scheme. Note that the successful cheating probability ϵ can be chosen flexibly by choosing the prime power p.

Theorem 2. *The scheme of §3.2 is (k, n, ϵ)-secure secret sharing schemes in the CDV model with parameters $|\mathcal{S}| = p^N, \epsilon = N/p, |\mathcal{V}_i| = p^{N+2} (= |\mathcal{S}|(\log_p |\mathcal{S}|)^2/\epsilon^2)$. Further, the scheme is secure for arbitrary secret distribution.*

Proof. Without loss of generality, we can assume $P_1, \dots, P_{k-1}$ are cheaters and they try to cheat P_k by forging their shares $v_i = (v_{s,i}, v_{e_0,i}, v_{e_1,i})$ $(1 \le i \le k-1.)$

We consider two cases depending on whether the cheaters know the secret. In the first case, suppose that the cheaters *know* the secret. The cheaters obtain the following information about e_0 and e_1 from their shares $v_1, \dots, v_{k-1}$ and the secret $s \in \mathcal{S}$: $e_\ell = L_k v_{e_\ell,k} + \sum_{j=1}^{k-1} L_j v_{e_\ell,j}$ (for $\ell = 0, 1$,), $e_0 - \sum_{j=1}^{N} s_j \cdot e_1^j = 0$ where $v_{e_0,k}$ and $v_{e_1,k}$ are unknown to the cheaters and each L_j is a Lagrange coefficient. For simplicity, we will rewrite e_i by $e_i = L_k v_{e_i,k} + C_i$ (for $i = 0, 1$) where $C_i = \sum_{j=1}^{k-1} L_j v_{e_i,j}$ are known to the cheaters. Then we have

$$L_k v_{e_0,k} + C_0 = \sum_{j=1}^{N} s_j \cdot (L_k v_{e_1,k} + C_1)^j \ . \tag{1}$$

Now suppose that the cheaters try to cheat P_k by forging their shares to $v'_i = (v'_{s,i}, v'_{e_0,i}, v'_{e_1,i})$ (for $1 \le i \le k-1$.) They succeed in cheating P_k if $e'_0 - \sum_{j=1}^{N} s'_j \cdot e'^j_1 = 0$ holds where e'_0, e'_1 and $s' (\neq s)$ are computed by $e'_0 = L_k v_{e_0,k} + \sum_{j=1}^{k-1} L_j v'_{e_0,j}$, $e'_1 = L_k v_{e_1,k} + \sum_{j=1}^{k-1} L_j v'_{e_1,j}$ and $s' = L_k v_{s,k} + \sum_{j=1}^{k-1} L_j v'_{s,j}$. Let $C'_i = \sum_{j=1}^{k-1} L_j v'_{e_i,j}$ (for $i = 0, 1$) then the cheaters succeed in cheating if the following equality holds (please note that the cheater can control the values of C'_0, C'_1 and s' as they want by adjusting their shares.[5])

$$L_k v_{e_0,k} + C'_0 = \sum_{j=1}^{N} s'_j \cdot (L_k v_{e_1,k} + C'_1)^j \tag{2}$$

[5] The cheaters can control s' since they can compute $v_{s,k}$ from their shares and s.

The successful cheating probability ϵ is computed as follows:

$$\epsilon = \Pr[s' \in \mathcal{S} \wedge s' \neq s] = \Pr[\text{eq. (1) and eq. (2) hold} \mid \text{eq. (1) holds}] = N/p\,.$$

We will show the above equation. The condition "eq. (1) and eq. (2) hold" is equivalent to "eq. (1) and eq. (3) hold" where eq. (3) is described as follows:

$$\textstyle\sum_{j=1}^{N} s_j \cdot (L_k v_{e_1,k} + C_1)^j - C_0 = \sum_{j=1}^{N} s'_j \cdot (L_k v_{e_1,k} + C'_1)^j - C'_0\,. \qquad (3)$$

Now let J be the largest number such that $s_J \neq s'_J$, then eq. (3) can be rewritten as the univariate equation $(s_J - s'_J)L_k^J \cdot v_{e_1,k}^J + \sum_{j=0}^{J-1} a_j \cdot v_{e_1,k}^j = 0$ of degree J with the variable $v_{e_1,k}$ where all the coefficients can be arbitrarily controlled by the cheaters except that $(s_J - s'_J)L_k^J \neq 0$. This equation has at most J $(\leq N)$ roots and for each root $v_{e_1,k}$, there exists a unique $v_{e_0,k}$ that satisfies eq. (1). Since the share generation algorithm $\mathsf{ShareGen}$ chooses actual $(v_{e_0,k}, v_{e_1,k})$ uniformly and randomly from the p pairs of $(v_{e_0,k}, v_{e_1,k})$ which satisfy eq. (1), we see that the successful cheating probability of the cheaters is upper bounded by N/p.

Now we consider the second case in which the cheaters *do not* know the secret. In this case the successful cheating probability of the cheaters who forge their shares from $v_i = (v_{s,i}, v_{e_0,i}, v_{e_1,i})$ to $v'_i = (v'_{s,i}, v'_{e_0,i}, v'_{e_1,i})$, where at least one $v'_{s,i}$ must satisfy $v'_{s,i} \neq v_{s,i}$, is computed as follows:

$$\begin{aligned}\epsilon &= \textstyle\sum_{s\in\mathcal{S}} \Pr[\mathcal{S} = s] \Pr[s' \in \mathcal{S} \wedge s' \neq s] \\ &= \textstyle\sum_{s\in\mathcal{S}} \Pr[\mathcal{S} = s] \Pr[\text{eq. (1) and eq. (2) hold} \mid \text{eq. (1) holds}] \;=\; N/p\,.\end{aligned}$$

The above equality holds since $\Pr[\text{eq. (1) and eq. (2) hold} \mid \text{eq. (1) holds}] = N/p$ holds for any $s \in \mathcal{S}$. □

Note that the above proof includes the proof for Theorem 2 since the first scheme is achieved by setting $N = 1$ in the second scheme.

4 Generalization

In this section, we present more general results on the access structures and on the class of hash functions used to detect cheating.

Though the schemes presented in Section 3 only deal with (k, n) threshold type access structure, we can show that the proposed technique can be applied to any *linear secret sharing schemes.* A linear secret sharing scheme is a class of secret sharing schemes with the following properties: (1) The secret s is an element of a finite field $\mathbb{F}$. (2) The shares $(v_1, \ldots, v_n)$ are generated by $(v_1, v_2, \ldots, v_n) = (s, r_1, \ldots, r_{t-1})M$ where M is a fixed $t \times n$ matrix over $\mathbb{F}$ and each $r_i \in \mathbb{F}$ is chosen randomly. (3) For a set of participants $\mathcal{P} = \{P_{i_1}, \ldots, P_{i_j}\} \in \Gamma$ and their shares $(v_{i_1}, \ldots, v_{i_j})$, the secret s is computed by $s = \sum_{k=1}^{j} c_{\mathcal{P},j} \cdot v_{i_j}$ where each $c_{\mathcal{P},j} \in \mathbb{F}$ is a constant uniquely determined from $\mathcal{P}$.

We can also generalize the class of hash function used to detect cheating. To characterize such class of hash function, we define a new class of hash function called *strongly key-differential universal* (ϵ-SKDU_2 for short) as follows:

Definition 2. *A family of hash functions $H : \mathcal{A} \to \mathcal{B}$ is called a strongly key-differential universal ϵ-SKDU_2 if there exists $\hat{b} \in \mathcal{B}$ such that for any distinct $a, a' \in \mathcal{A}$ and for any $c \in \mathcal{E}$,*

$$\frac{|\{h_e \mid e \in \mathcal{E},\ h_e(a) = \hat{b},\ h_{e+c}(a') = \hat{b}\}|}{|\{h_e \mid e \in \mathcal{E},\ h_e(a) = \hat{b}\}|} \leq \epsilon. \tag{4}$$

Further, ϵ-SKDU_2 is called an "efficiently samplable" *if there exists an efficient (i.e. polynomial time) algorithm to choose $e \in \mathcal{E}$ randomly from the set $\{e \in \mathcal{E} \mid h_e(a) = \hat{b}\}$ for any $a \in \mathcal{A}$.*

The following theorem shows that we can construct secret sharing scheme capable of detecting cheating in the CDV model from any linear secret sharing schemes over $\mathcal{S}$ and over $\mathcal{E}$, and any efficiently samplable ϵ-SKDU_2 with the domain $\mathcal{S}$.

Theorem 3. *If there exist linear secret sharing schemes over $\mathcal{S}$ and $\mathcal{E}$ for a common access structure Γ and an efficiently samplable ϵ-SKDU_2 $H : \mathcal{S} \to \mathcal{B}$, then there exists a secret sharing scheme capable of detecting cheating for the access structure Γ in the CDV model such that the successful cheating probability is equal to ϵ for arbitrary secret distribution.*

Proof. Let $\mathcal{S}$ and $\mathcal{E}$ be a set of the secrets and the set of keys for ϵ-SKDU_2, respectively and let $\mathbf{SS}_1 = (\mathsf{ShareGen}_1, \mathsf{Reconst}_1)$ and $\mathbf{SS}_2 = (\mathsf{ShareGen}_2, \mathsf{Reconst}_2)$ be linear secret sharing schemes over $\mathcal{S}$ and over $\mathcal{E}$ for the same access structure Γ, respectively. We construct a secret sharing scheme secure against cheaters $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ as follows.

Share Generation: On input a secret $s \in \mathcal{S}$, the share generation algorithm $\mathsf{ShareGen}$ outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Choose a random $e \in \mathcal{E}$ such that $h_e(s) = \hat{b}$, which can be computed efficiently since the efficiently samplable ϵ-SKDU_2 is used.
2. Generate $(v_{s,1}, \ldots, v_{s,n}) \leftarrow \mathsf{ShareGen}_1(s)$ and $(v_{e,1}, \ldots, v_{e,n}) \leftarrow \mathsf{ShareGen}_2(e)$.
3. Compute the share $v_i = (v_{s,i}, v_{e,i})$ of each P_i and output $(v_1, \ldots, v_n)$.

Secret Reconstruction and Validity Check: On input t shares $(v_{i_1}, \ldots, v_{i_t})$ such that $\{P_{i_1}, \ldots, P_{i_t}\} \in \Gamma$, the secret reconstruction algorithm $\mathsf{Reconst}$ outputs a secret $s \in \mathcal{S}$ or $\perp$ as follows:

1. Compute $\hat{s} \leftarrow \mathsf{Reconst}_1(v_{s,i_1}, \ldots, v_{s,i_t})$ and $\hat{e} \leftarrow \mathsf{Reconst}_2(v_{e,i_1}, \ldots, v_{e,i_t})$.
2. Output s if $h_{\hat{e}}(\hat{s}) = \hat{b}$. Otherwise $\mathsf{Reconst}$ outputs $\perp$.

Now we show that $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ constructed above is ϵ-secure. Without loss of generality we can assume that $\mathcal{P} = \{P_1, \ldots, P_t\}$ is an element of Γ and that $P_1, \ldots, P_{t-1}$ are cheaters who try to cheat P_t. There are two cases to consider. In the first case, suppose that the cheaters *know* the secret.

Let $v_i = (v_{s,i}, v_{e,i})$ be the share of P_i. Since the cheaters know their shares $v_1, \ldots, v_{t-1}$ and the secret s and that $\mathbf{SS}_1$ and $\mathbf{SS}_2$ are the linear secret sharing schemes, the cheaters know $h_e(s) = \hat{b}$ holds where e is computed by $e = c_{\mathcal{P},t}v_{e,t} + \sum_{j=1}^{t-1} c_{\mathcal{P},j}v_{e,j}$ for a constant $c_{\mathcal{P},i}$. Now suppose the cheaters try to cheat P_t by forging their shares to $v'_i = (v'_{s,i}, v'_{e,i})$ (for $1 \leq i \leq t-1$.) They succeed in cheating P_t if $h_{e'}(s') = \hat{b}$ holds for e' and $s'(\neq s)$ computed by $e' = c_{\mathcal{P},t}v_{e,t} + \sum_{j=1}^{t-1} c_{\mathcal{P},j}v'_{e,j}$, $s' = c_{\mathcal{P},t}v_{s,t} + \sum_{j=1}^{t-1} c_{\mathcal{P},j}v'_{s,j}$. Since $e' = e + \sum_{j=1}^{t-1} c_{\mathcal{P},j}(v'_{e,j} - v_{e,j})$ holds, we see that the cheaters succeed in cheating if $h_{e+C}(s') = \hat{b}$ holds where $C = \sum_{j=1}^{t-1} c_{\mathcal{P},j}(v'_{e,j} - v_{e,j})$ is known to the cheaters. Therefore, the successful cheating probability ϵ is computed as follows.

$$\begin{aligned}
&\Pr[s' \in \mathcal{S} \wedge s' \neq s] \\
&\ = \Pr[h_e(s) = \hat{b} \text{ and } h_{e+C}(s') = \hat{b} \mid h_e(s) = \hat{b}] \\
&\ = \frac{\Pr[h_e(s) = \hat{b} \text{ and } h_{e+C}(s') = \hat{b}]}{\Pr[h_e(s) = \hat{b}]} = \frac{|\{h_e \mid h_e(s) = \hat{b},\ h_{e+C}(s') = \hat{b}\}|}{|\{h_e \mid h_e(s) = \hat{b}\}|} \leq \epsilon
\end{aligned}$$

where the last equation directly follows from eq. (4).

It can be proven that the successful cheating probability is upper bounded by ϵ when the cheaters *do not* know the secret by the same technique used in Theorem 2. □

It is easily checked that the families of hash function used in the proposed schemes of Section 3 meet the requirements of efficiently samplable ϵ-SKDU_2.

Constructions of ϵ-SKDU_2 other than those used in the proposed schemes will be of independent interest. The following theorem shows that an ϵ-SKDU_2 (and therefore, a secret sharing scheme capable of detecting cheating) can be constructed from an ϵ-ASU_2 with additional properties.

Theorem 4. *If a family of hash functions $H : \mathcal{A} \to \mathcal{B}$ is an ϵ-ASU_2 with the properties (1) and (2) below then H is an efficiently samplable ϵ-SKDU_2.*

(1) H is constructed from $H_\Delta : \mathcal{A} \to \mathcal{B}$ of ϵ-$\mathsf{A\Delta U}_2$ as follows, where ϵ-$\mathsf{A\Delta U}_2$ is a family of hash functions such that $|\{h_e \in H_\Delta \mid h_e(a) - h_e(a') = b\}| = \epsilon|H|$ for any distinct $a, a' \in \mathcal{A}$ and for any $b \in \mathcal{B}$.

$$H = \{h_{e_0,e_1} \mid h_{e_0,e_1}(a) = h'_{e_0}(a) + e_1,\ \ h'_{e_0} \in H_\Delta, e_1 \in \mathcal{B}\}$$

(2) H_Δ is linear with respect to the key; that is, $h'_{e+e'}(a) = h'_e(a) + h'_{e'}(a)$ holds for any $e, e' \in \mathcal{E}$ and for any $a \in \mathcal{A}$.

Proof. It is well known that the family of hash functions H constructed as above is ϵ-ASU_2 (please refer to [11] for the proof.) Let $\hat{b}$ be an arbitrary element of $\mathcal{B}$ then we will show that H satisfies the conditions of an efficiently samplable ϵ-SKDU_2. First, it is easy to see that e_0 and e_1 such that $h_{e_0,e_1}(a) = \hat{b}$ is efficiently samplable by choosing $e_0 \in \mathcal{E}$ randomly and by computing $e_1 = \hat{b} - h_{e_0}(a)$. Next, we show that eq. (4) holds for H. Since H is constructed based on H_Δ with the property $h'_{e+e'}(a) = h'_e(a) + h'_{e'}(a)$ for any $h' \in H_\Delta$, $h_{e_0+c_0,e_1+c_1}(a) = h'_{e_0+c_0}(a) + (e_1 + c_1) = (h'_{e_0}(a) + e_1) + (h'_{c_0}(a) + c_1) = h_{e_0,e_1}(a) + h_{c_0,c_1}(a)$ holds for any $a \in \mathcal{A}$ and for any $(e_0, e_1), (c_0, c_1) \in \mathcal{E} \times \mathcal{B}$. Therefore, the following equation holds.

$$\begin{aligned}
&|\{h_{e_0,e_1} \in H \mid h_{e_0,e_1}(a) = \hat{b},\ h_{e_0+c_0,e_1+c_1}(a') = \hat{b}\}| \\
&= |\{h_{e_0,e_1} \in H \mid h_{e_0,e_1}(a) = \hat{b},\ h_{e_0,e_1}(a') = \hat{b} - h_{c_0,c_1}(a')\}| \\
&= |\{h_{e_0,e_1} \in H \mid h_{e_0,e_1}(a) = \hat{b},\ h_{e_0,e_1}(a') = \hat{b}'\}| \ = \ \epsilon|H|/|\mathcal{B}|
\end{aligned}$$

where the last equation follows from the second condition of ϵ-ASU_2. Combining the above equation and the first property of ϵ-ASU_2: $|\{h_{e_0,e_1} \in H \mid h_{e_0,e_1}(a) = \hat{b}\}| = |H|/|\mathcal{B}|$, we have $\frac{|\{h_{e_0,e_1} \in H | h_{e_0,e_1}(a) = \hat{b},\ h_{e_0+c_0,e_1+c_1}(a') = \hat{b}\}|}{|\{h_{e_0,e_1} \in H | h_{e_0,e_1}(a) = \hat{b}\}|} = \epsilon$ for any distinct $a, a' \in \mathcal{A}$ and for any $(c_0, c_1) \in \mathcal{E} \times \mathcal{B}$. □

Please note that the family of hash function used in the first scheme is constructed based on Theorem 4, whereas the family of hash function used in the second scheme is not. Therefore, we see that SKDU_2 can be constructed by other means than Theorem 4.

5 Coping with More Powerful Cheaters

In this section, we consider the models with more powerful cheaters than those in the OKS and the CDV models and we present secure schemes against them.

In the OKS model and the CDV model, the secret reconstruction algorithm Reconst is defined to take only a list of share $(v_{i_1}, \ldots, v_{i_k})$ as input. In actual schemes, however, the identities of the owners $i_1, \ldots, i_k$ are usually required to reconstruct the secret. This means that we implicitly assume there exist means to know the *correct* identities of share holders in the secret reconstruction phase of both the OKS and the CDV models. In the real life, however, it is very difficult to realize an identification scheme secure against adversaries with unlimited computational power. Therefore, it is highly desired to construct secret sharing schemes capable of detecting cheating without relying on secure identification.

To this end, we define new models: the OKS$^+$ model and the CDV$^+$ model which are slight modifications of the OKS model and the CDV model, respectively. In both new models, we modify a secret reconstruction algorithm Reconst and a game Game$^+$ of cheaters $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ against $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ as follows. The secret reconstruction algorithm Reconst takes a list $((i_1, v_{i_1}), (i_2, v_{i_2}), \ldots, (i_k, v_{i_k}))$ of pairs of an identity i_ℓ and a share v_{i_ℓ} of P_{i_ℓ}. Cheaters in the new models are allowed to forge their identities as well as their shares. To characterize such cheaters, a game Game$^+$ is defined as follows.

```
Game$^+$($\mathbf{SS}$, A)
    $s \leftarrow \mathcal{S}$;    // according to the probability distribution over $\mathcal{S}$.
    $(v_1, \ldots, v_n) \leftarrow \mathsf{ShareGen}(s)$;
    $(i_i, \ldots, i_{k-1}) \leftarrow \mathsf{A}_1(X)$;
    // set $X = s$ for the CDV$^+$ model, $X = \emptyset$ for the OKS$^+$ model.
    $((i'_1, v'_{i'_1}), \ldots, (i'_{k-1}, v'_{i'_{k-1}}), i_k) \leftarrow \mathsf{A}_2(v_{i_1}, \ldots, v_{i_{k-1}}, X)$;
```

The advantage of cheaters is redefined by $Adv(\mathbf{SS}, \mathsf{A}) = \Pr[s' \in \mathcal{S} \wedge s' \neq s]$, where $s' = \mathsf{Reconst}((i'_1, v'_{i'_1}), (i'_2, v'_{i'_2}), \ldots, (i'_{k-1}, v'_{i'_{k-1}}), (i_k, v_{i_k}))$ and the probability is taken over the distribution of $\mathcal{S}$, and over the random tapes of $\mathsf{ShareGen}$ and A.

Note that all the bounds for the OKS model (resp., the CDV model) (e.g. Propositions 1–3 and Corollary 1) are also valid for OKS$^+$ model (resp., the CDV$^+$ model) since a scheme secure in the OKS$^+$ model (resp., the CDV$^+$ model) are also secure in the OKS model (resp., the CDV model.)

Though the schemes secure in the OKS model (resp., the CDV model) are not necessarily secure in the OKS$^+$ model (resp., the CDV$^+$ model,) the scheme presented in [8] can be proven to be secure in the OKS$^+$ model and the scheme presented in [12] can be proven to be secure in the CDV$^+$ model. With respect to the proposed schemes, the first scheme can be shown to be secure in the CDV$^+$ model. However, the second scheme is *not* secure in the CDV$^+$ model. This is because the security proof of the second scheme strongly relies on the fact that the cheaters can not manipulate the Lagrange coefficient L_k, which is not the case in the CDV$^+$ model. When cheaters can manipulate the Lagrange coefficient as they want, they will succeed in cheating with probability one, which is possible by forging the Lagrange coefficient L_k to $L'_k(\neq L_k)$ in eq. (2) and by adjusting s'_j, C'_0 and C'_1 to make eq. (2) equivalent to eq. (1).

The good news is that the second scheme secure can be made secure in CDV$^+$ model by slight modification. The main idea of the modified scheme is to introduce a *constant padding* to a hash function. Specifically, we choose a key e of a hash families with which $h_e(s_1, \ldots, s_N, 1, 1, 0, 1) = 0$ instead of choosing a key such that $h_e(s_1, \ldots, s_N) = 0$ as in the second scheme. In this modified scheme, we can show that cheaters cannot make eq. (2) equivalent to eq. (1) unless they leave the Lagrange coefficient L_k and the secret $s = (s_1, \ldots, s_N)$ unchanged. The modified scheme can be described as follows.

Share Generation: On input a secret $s = (s_1, \ldots, s_N) \in GF(p)^N$, the share generation algorithm $\mathsf{ShareGen}$ outputs a list of shares $(v_1, \ldots, v_n)$ according to the following procedure. Please note that we sometimes regard $s = (s_1, \ldots, s_N)$ as an element of $GF(p^N)$ instead of $GF(p)^N$.

1. Choose random $e_0, e_1 \in GF(p)$ such that $e_0 - (e_1^{N+4} + e_1^{N+2} + e_1^{N+1} + \sum_{j=1}^{N} s_j e_1^j) = 0$.
2. Generate random polynomials $f_s(x) \in GF(p^N)[X]$ and $f_{e_0}(x), f_{e_1}(x) \in GF(p)[X]$ of degree $k-1$ such that $f_s(0) = s$, $f_{e_0}(0) = e_0$ and $f_{e_1}(0) = e_1$.
3. Compute $v_i = (f_s(i), f_{e_0}(i), f_{e_1}(i))$ and output $(v_1, \ldots, v_n)$.

Secret Reconstruction and Validity Check: On input a list of k pair of identities and shares $((i_1, v_{i_1}), \ldots, (i_k v_{i_k}))$, the secret reconstruction algorithm $\mathsf{Reconst}$ outputs a secret s or $\perp$ according to the following procedure.

1. Reconstruct $\hat{s}, \hat{e}_0$ and $\hat{e}_1$ from $v_{i_1}, \ldots, v_{i_k}$ using Lagrange interpolation.
2. Output s if $\hat{e}_0 - (\hat{e}_1^{N+4} + \hat{e}_1^{N+2} + \hat{e}_1^{N+1} + \sum_{j=1}^{N} \hat{s}_j \hat{e}_1^j) = 0$ holds. Otherwise $\mathsf{Reconst}$ outputs $\perp$.

Security of the modified scheme is summarized by the following theorem.

Theorem 5. *The modified scheme presented above is* (k, n, ϵ)*-secure secret sharing schemes in the CDV*$^{+}$ *model with the following parameters:* $|\mathcal{S}| = p^N, \epsilon = (N+4)/p$ *and* $|\mathcal{V}_i| = p^{N+2} (= |\mathcal{S}|(\log_p |\mathcal{S}|+4)^2/\epsilon^2)$. *Further, the scheme is secure for arbitrary secret distribution.*

Proof. The proof is similar to that of Theorem 2. Let P_j $(1 \leq j \leq k-1)$ be cheaters who try to cheat P_k by forging their identities j to $i_j (\neq k)$ and corresponding shares to $v'_{i_j} = (v'_{s,i_j}, v'_{e_0,i_j}, v'_{e_1,i_j})$ $(1 \leq j \leq k-1.)$

As in the proof of Theorem 2, we consider two cases depending on whether the cheaters know the secret. In the first case, suppose that the cheaters *know* the secret. The cheaters obtain the following information about e_0 and e_1 from their shares $v_1, \ldots, v_{k-1}$ and the secret $s \in \mathcal{S}$: $e_\ell = L_k v_{e_\ell,k} + \sum_{j=1}^{k-1} L_j v_{e_\ell,j}$ $(\ell = 0, 1)$, $e_0 - (e_1^{N+4} + e_1^{N+2} + e_1^{N+1} + \sum_{j=1}^{N} s_j \cdot e_1^j) = 0$ where $v_{e_0,k}$ and $v_{e_1,k}$ are unknown to the cheaters and each L_j is a Lagrange coefficient. For simplicity, we will rewrite e_i by $e_i = L_k v_{e_i,k} + C_i$ (for $i = 0, 1$) where $C_i = \sum_{j=1}^{k-1} L_j v_{e_i,j}$ is known to the cheaters. Then we have the following equality.

$$L_k v_{e_0,k} + C_0 = \textstyle\sum_{j \in \{1,2,4\}} (L_k v_{e_1,k} + C_1)^{N+j} + \sum_{j=1}^{N} s_j \cdot (L_k v_{e_1,k} + C_1)^j \quad (5)$$

Now suppose the cheaters P_j $(1 \leq j \leq k-1)$ try to cheat P_k by forging their identities to i_j and by forging corresponding shares to $v'_{i_j} = (v'_{s,i_j}, v'_{e_0,i_j}, v'_{e_1,i_j})$. They succeed in cheating P_k if $e'_0 - (\sum_{j=\{1,2,4\}} e_1'^{N+j} + \sum_{j=1}^{N} s'_j \cdot e_1'^j) = 0$ holds where e'_0, e'_1 and $s' (\neq s)$ are computed by $e'_\ell = L'_k v_{e_\ell,k} + \sum_{j=1}^{k-1} L'_{i_j} v'_{e_\ell,i_j}$ (for $\ell = 0, 1$), $s' = L'_k v_{s,k} + \sum_{j=1}^{k-1} L'_{i_j} v'_{s,i_j}$. Let $C'_\ell = \sum_{j=1}^{k-1} L'_{i_j} v'_{e_\ell,i_j}$ (for $\ell = 0, 1$) then the cheaters succeed in cheating if the following equality holds (as in Theorem 2, the cheaters can control the values of C'_0, C'_1 and s' as they want.)

$$L'_k v_{e_0,k} + C'_0 = \textstyle\sum_{j \in \{1,2,4\}} (L'_k v_{e_1,k} + C'_1)^{N+j} + \sum_{j=1}^{N} s'_j \cdot (L'_k v_{e_1,k} + C'_1)^j \quad (6)$$

The successful cheating probability ϵ is computed by $\epsilon = \Pr[s' \in \mathcal{S} \wedge s' \neq s] = \Pr[\text{eq. (5) and eq. (6) hold} \mid \text{eq. (5) holds}]$. We will show that $\epsilon = (N+4)/p$. First, assume that eq. (5) is not equivalent to eq. (6) (i.e. $L'_k \times$ eq. (5) is not identical to $L_k \times$ eq. (6).) In this case, ϵ is proven to be $(N+4)/p$ by similar discussion to the proof of Theorem 2. Next, we will show that if the cheaters make eq. (6) equivalent to eq. (5) then successful cheating probability becomes 0. This can be proven by showing that eq. (5) is equivalent to eq. (6) only if the $L'_k = L_k$, $C'_i = C_i$ (for $i = 0, 1$) and $s_j = s'_j$ (for $1 \leq j \leq N$) since the cheaters succeed in cheating only when P_k accepts s' such that $s' \neq s$. Suppose $L_k \times$ eq. (5) and $L'_k \times$ eq. (6) are identical then their coefficients of $v_k^{N+4}, v_k^{N+3}, v_k^{N+2}$ and v_k^{N+1} must be identical. Therefore, we have the following equations.

$$L'_k L_k^{N+4} = L_k L_k'^{N+4} \quad (7)$$

$$\tbinom{N+4}{1} C_1 L'_k L_k^{N+3} = \tbinom{N+4}{1} C'_1 L_k L_k'^{N+3} \quad (8)$$

$$\left(\binom{N+4}{2}C_1^2+1\right)L'_kL_k^{N+2}=\left(\binom{N+4}{2}C_1'^2+1\right)L_kL_k'^{N+2} \tag{9}$$

$$\left(\binom{N+4}{3}C_1^3+\binom{N+2}{1}C_1+1\right)L'_kL_k^{N+1}=\left(\binom{N+4}{3}C_1'^3+\binom{N+2}{1}C'_1+1\right)L_kL_k'^{N+1} \tag{10}$$

From eq. (7) and eq. (8) we have $L_k^{N+3}=L_k'^{N+3}$ and $C_1/L_k=C'_1/L'_k$. Using these relations eq. (7)–eq. (10) can be rewritten as follows.

$$L_k^{N+3}=L_k'^{N+3},\quad C_1/L_k=C'_1/L'_k,\quad L_k^{N+1}=L_k'^{N+1},\quad L_k^N=L_k'^N$$

The above equalities holds if and only if $L_k=L'_k$ and $C_1=C'_1$. Further, $s_j=s'_j$ (for $1\leq j\leq N$) can be also derived from the condition that the coefficients of v_k^j in eq. (5) and eq. (6) are identical. Finally, $C_0=C'_0$ is derived from the condition that the constant terms of eq. (5) and eq. (6) are identical.

Now we consider the second case in which the cheaters *do not* know the secret. In this case the successful cheating probability of the cheaters who forge their identities and corresponding shares from $(j,(v_{s,j},v_{e_0,j},v_{e_1,j}))$ to $(i_j,(v'_{s,i_j},v'_{e_0,i_j},v'_{e_1,i_j}))$ is computed as follows:

$$\begin{aligned}\epsilon &= \textstyle\sum_{s\in\mathcal{S}}\Pr[\mathcal{S}=s]\Pr[s'\in\mathcal{S}\wedge s'\neq s]\\ &= \textstyle\sum_{s\in\mathcal{S}}\Pr[\mathcal{S}=s]\Pr[\text{eq. (5) and eq. (6) hold}\mid\text{eq. (5) holds}] \;=\; (N+4)/p\,.\end{aligned}$$

The above equality holds since $\Pr[\text{eq. (5) and eq. (6) hold}\mid\text{eq. (5) holds}]=(N+4)/p$ holds for any $s\in\mathcal{S}$. □

The following theorem gives a generalized result analogous to Theorem 3.

Theorem 6. *If there exist linear secret sharing schemes over $\mathcal{S}$ and $\mathcal{E}$ for a common access structure Γ and a family of hash functions $H:\mathcal{S}\to\mathcal{B}$ which satisfies the conditions (1)–(3) below, then there exists a secret sharing scheme capable of detecting cheating for the access structure Γ in the CDV$^+$ model such that the successful cheating probability equals ϵ for arbitrary secret distribution.*

1. *Addition and (scalar) multiplication over the set of keys $\mathcal{E}$ of H are defined.*
2. *There exists $\hat{b}\in\mathcal{B}$ such that for any distinct $a,a'\in\mathcal{A}$ and for any c_0 and $c_1\in\mathcal{E}$, $\dfrac{|\{h_e\mid e\in\mathcal{E},\ h_e(a)=\hat{b},\ h_{c_0e+c_1}(a')=\hat{b}\}|}{|\{h_e\mid e\in\mathcal{E},\ h_e(a)=\hat{b}\}|}\leq\epsilon$. holds.*
3. *There exists an efficient (i.e. polynomial time) algorithm to choose $e\in\mathcal{E}$ randomly from the set $\{e\in\mathcal{E}\mid h_e(a)=\hat{b}\}$ for any $a\in\mathcal{A}$.*

Proof. The proof is similar to that of Theorem 3. Let $\mathcal{S}$ and $\mathcal{E}$ be a set of the secrets and the set of keys for a function family H, respectively. Further, let $\mathbf{SS}_1=(\mathsf{ShareGen}_1,\mathsf{Reconst}_1)$ and $\mathbf{SS}_2=(\mathsf{ShareGen}_2,\mathsf{Reconst}_2)$ be linear secret sharing schemes over $\mathcal{S}$ and over $\mathcal{E}$ for the same access structure Γ, respectively. The share generation algorithm $\mathsf{ShareGen}$ and $\mathsf{Reconst}$ are identical to those defined in the proof of Theorem 3 except that the family of hash functions used here meets the condition 1–3 of Theorem 6.

Now we show that the above $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is ϵ-secure even when the cheaters forge their identities as well as their shares. Without loss of generality we can assume that $\mathcal{P} = \{P_1, \ldots, P_t\}$ is an element of Γ and that $P_1, \ldots, P_{t-1}$ are cheaters who try to cheat P_t. There are two cases to consider. In the first case, suppose that the cheaters *know* the secret. Let $v_i = (v_{s,i}, v_{e,i})$ be the share of P_i. Since the cheaters know their shares $v_1, \ldots, v_{t-1}$ and the secret s and that $\mathbf{SS}_1$ and $\mathbf{SS}_2$ are the linear secret sharing schemes, the cheaters know $h_e(s) = \hat{b}$ holds where e is computed by $e = c_{\mathcal{P},t}v_{e,t} + \sum_{j=1}^{t-1} c_{\mathcal{P},j}v_{e,j}$ for a constant $c_{\mathcal{P},i}$. Now suppose the cheaters try to cheat P_t by forging their identities from j to i_j (for $1 \leq j \leq t-1$) and corresponding shares to $v'_{i_j} = (v'_{s,i_j}, v'_{e,i_j})$ (for $1 \leq j \leq t-1$.) They succeed in cheating P_t if $h_{e'}(s') = \hat{b}$ holds for e' and $s'(\neq s)$ computed by $e' = c'_{\mathcal{P},t}v_{e,t} + \sum_{j=1}^{t-1} c'_{\mathcal{P},i_j} v'_{e,i_j}$, $s' = c'_{\mathcal{P},t}v_{s,t} + \sum_{j=1}^{t-1} c'_{\mathcal{P},i_j} v'_{s,i_j}$. Since $e' = (\frac{c'_{\mathcal{P},t}}{c_{\mathcal{P},t}})e + \sum_{j=1}^{t-1}(c'_{\mathcal{P},i_j} v'_{e,i_j} - \frac{c_{\mathcal{P},t}c'_{\mathcal{P},i_j}}{c'_{\mathcal{P},t}} \cdot v_{e,j})$ holds, we see that the cheaters succeed in cheating if $h_{C_0 \cdot e + C_1}(s') = \hat{b}$ holds where $C_0 = c'_{\mathcal{P},t}/c_{\mathcal{P},t}$ and $C_1 = \sum_{j=1}^{t-1}(c'_{\mathcal{P},i_j} v'_{e,i_j} - \frac{c_{\mathcal{P},t}c'_{\mathcal{P},i_j}}{c'_{\mathcal{P},t}} \cdot v_{e,j})$ are known to the cheaters. Therefore, the successful cheating probability ϵ is computed as follows.

$$\begin{aligned}\Pr[s' \in \mathcal{S} \wedge s' \neq s] &= \Pr[h_e(s) = \hat{b} \ \text{and} \ h_{C_0 \cdot e + C_1}(s') = \hat{b} \mid h_e(s) = \hat{b}] \\ &= \frac{|\{h_e \mid h_e(s) = \hat{b}, \ h_{C_0 \cdot e + C_1}(s') = \hat{b}\}|}{|\{h_e \mid h_e(s) = \hat{b}\}|} \ \leq \ \epsilon\end{aligned}$$

where the last equation directly follows from the condition (2) of Theorem 6.

It can be proven that the successful cheating probability is upper bounded by ϵ when the cheaters *do not* know the secret by the same technique used in Theorem 5. □

6 Conclusion

In this paper, we proposed two efficient $(k, n, \epsilon_{\mathsf{CDV}})$-secure secret sharing schemes in the CDV model which are proven to be secure for arbitrary secret distribution. The first scheme is nearly optimum with respect to the size of shares; that is, the size of share is only one bit longer than the lower bound of Corollary 1. In the second scheme, the size of share is larger than that in the first scheme. However, the second scheme possesses a particular merit in that the successful cheating probability can be chosen without regard to the size of the secret. Table 1 below compares the bit length of shares in the three schemes for the various security parameters where the secret is 1024 bit and the access structure considered is 3-out-of-5 threshold type access structure. Compared to the scheme of [12] the size of shares in the proposed scheme (the second scheme) is smaller for all security parameters. It is interesting to note that, when $\varepsilon > |\mathcal{S}|^{-1/2}$, the size of the share in the proposed scheme is even smaller than that in [8] which is proven to be secure only in the OKS model. This is because ϵ is determined to be $\epsilon = 2^{-1024}$ when the secret is 1024 bit in the scheme of [8]. Therefore, ϵ is forced to be 2^{-1024}

ϵ	Proposed Scheme	Tompa and Woll	Ogata *et al.*
2^{-128}	1286	2306	2048
2^{-256}	1540	2562	2048
2^{-512}	2050	3074	2048
2^{-1024}	3072	4098	2048

Table 1. Comparison table of the bit length of the shares (for the secret of 1024 bit)

in [8] even when we only require the security level of $\epsilon = 2^{-128}$ or $\epsilon = 2^{-256}$, which makes the size of share larger than that in the proposed scheme when ϵ is relatively large (please note that $\epsilon = 2^{-128}$ or $\epsilon = 2^{-256}$ will be secure enough in most settings.)

It will be a future study to find $(k, n, \epsilon_{\mathsf{CDV}})$-secure secret sharing schemes in the CDV model which are secure for arbitrary secret distribution and the bound of Corollary 1 is satisfied with equality.

References

1. G. R. Blakley, "Safeguarding cryptographic keys," Proc. AFIPS 1979, National Computer Conference, vol. 48, pp. 313–137, 1979. vol. 4, no. 4, pp. 502–510, 1991.
2. M. Carpentieri, "A Perfect Threshold Secret Sharing Scheme to Identify Cheaters," Designs, Codes and Cryptography, vol. 5, no. 3, pp. 183–187, 1995.
3. M. Carpentieri, A. De Santis and U. Vaccaro, "Size of Shares and Probability of Cheating in Threshold Schemes," Proc. Eurocrypt'93, Lecture Notes in Computer Science, vol. 765, Springer Verlag, pp. 118–125, 1993.
4. R. Cramer, I. Damgård and U. M. Maurer, "General Secure Multi-party Computation from any Linear Secret-Sharing Scheme," Proc. Eurocrypt'00, Lecture Notes in Computer Science, vol. 1807, Springer Verlag, pp. 316–334, 2000.
5. B. den Boer, "A Simple and Key-Economical Unconditional Authentication Scheme," Journal of Computer Security, vol. 2, pp. 65–71, 1993.
6. K. Kurosawa, S. Obana and W. Ogata, "t-Cheater Identifiable (k, n) Secret Sharing Schemes," Proc. Crypto'95, Lecture Notes in Computer Science, vol. 963, Springer Verlag, pp. 410–423, 1995.
7. F. MacWilliams and N. Sloane, "The Theory of Error Correcting Codes," North Holland, Amsterdam, 1977.
8. W. Ogata, K. Kurosawa and D. R. Stinson, "Optimum Secret Sharing Scheme Secure against Cheating," SIAM Journal on Discrete Mathematics, vol. 20, no. 1, pp. 79–95, 2006.
9. T. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," Proc. Crypto'91, Lecture Notes in Computer Science, vol 576, Springer Verlag, pp. 129–149, 1991.
10. A. Shamir, "How to Share a Secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.
11. D. R. Stinson, "On the Connections between Universal Hashing, Combinatorial Designs and Error-Correcting Codes," Congressus Numerantium 114, pp. 7–27, 1996.
12. M. Tompa and H. Woll, "How to Share a Secret with Cheaters," Journal of Cryptology, vol. 1, no. 3, pp. 133–138, 1989.