# Efficient $(k, n)$ Threshold Secret Sharing Schemes Secure Against Cheating from $n - 1$ Cheaters

Toshinori Araki

NEC Corporation
t-araki@ek.jp.nec.com

**Abstract.** In $(k, n)$ threshold secret sharing scheme, Tompa and Woll consider a problem of cheaters who try to make another participant reconstruct invalid secret. Later, the model of such cheating is formalized in some researches. Some schemes secure against cheating of these models are proposed. However, in these models, the number of colluding participants is restricted to $k - 1$ or less. In this paper, we consider $k$ or more colluding participants. Of course, secrecy is not maintained to such participants. However, if considering detecting the fact of cheating, we need to consider a cheating from $k$ or more colluding participants. In this paper, we propose a $(k, n)$ threshold secret sharing scheme that is capable of detecting the fact of cheating from $n - 1$ or less colluding participants. A scheme proposed by Tompa and Woll can be proven to be a $(k, n)$ threshold secret sharing scheme that is capable of detecting the fact of cheating from $n - 1$ or less colluding participants. However, our proposed scheme is much more efficient with respect to the size of shares.

## 1 Introduction

*Background.* A $(k, n)$ threshold secret sharing scheme [1,10] is a cryptographic primitive used to distribute a secret $s$ to $n$ participants in such a way that a set of $k$ or more participants can recover the secret $s$ and a set of $k - 1$ or less participants cannot obtain any information about $s$. A piece of information held by participant is called a share.

Various problems in $(k, n)$ threshold secret sharing schemes are considered. Above all, the problem of cheaters in threshold schemes is considered in various researches.

Tompa and Woll [11] considered the following cheating scenario. Suppose that colluding participants want to cheat another participant by submitting forged shares in the reconstruction. They succeed if the reconstructed value is different from the original secret. Later, a model of such cheating is formalized in [3,8]. Some schemes secure against cheating of these models are proposed [2,7,8,11].

*Our Contribution.* In the models of [3,8], the number of colluding participants is restricted to $k - 1$ or less. However, we can consider $k$ or more colluding

J. Pieprzyk, H. Ghodosi, and E. Dawson (Eds.): ACISP 2007, LNCS 4586, pp. 133–142, 2007.

participants. Of course, secrecy is not maintained to such participants. However, if considering detecting the fact of cheating, we need to consider a cheating from $k$ or more colluding participants. In this paper, we construct a $(k, n)$ threshold secret sharing scheme that is capable of detecting the fact of cheating from $n-1$ or less colluding participants.

Schemes in [2,7,8] are not capable of detecting the fact of cheating from $k$ or more colluding participants. Scheme in [11] is capable of detecting the fact of cheating from $n-1$ colluding participants. However our proposed scheme is much more efficient with respect to the size of shares. Particularly, the size of the share in the proposed scheme is a few bit longer than lower bound of [7] when parameter $k$,$n$ are small and $|\mathcal{S}|$[1] is smaller than $1/\epsilon$ , where $\epsilon$ denotes the successful probability of cheating and $\mathcal{S}$ denotes the set of secrets.

*Organization.* The rest of the paper is organized as follows. In Section 2, we briefly review the models of secret sharing schemes capable of detecting cheating, and we discuss previous works done on them. In Section 3, we introduce a new model of cheating from $n-1$ or less colluding cheaters. In Section 4, we present an efficient scheme secure in the new model. In Section 5, we consider the problem of forged reconstruction result. In Section 6, we summarize our work.

## 2 Preliminaries

### 2.1 $(k, n)$ Threshold Scheme

In secret sharing schemes, there are $n$ participants $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a dealer $D$.

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 $v_i$ is called a *share* and is given to a participant $P_i$. Ordinarily, $\mathsf{ShareGen}$ is invoked by the $D$. Secret reconstruction algorithm $\mathsf{Reconst}$ takes a list of shares and outputs a secret $s \in \mathcal{S}$. In a $(k, n)$ threshold scheme [1,10], any $k$ or more participants can recover $s$ but no subset of less than $k$ participants can determine any partial information about $s$.

### 2.2 Secret Sharing Schemes Secure Against Cheating

A secret sharing scheme capable of detecting cheating was first presented by Tompa and Woll [11]. They considered the scenario that $k-1$ or less cheaters submit forged shares in the secret reconstruction phase. Such cheaters will succeed if another participant 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 [4] first considered a model

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

[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 [3,6] nor *verifiable secret sharing schemes* [9,5] are within the scope of this paper.

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 $\mathsf{ShareGen}$ is the same as that in the ordinary secret sharing schemes. A secret reconstruction algorithm $\mathsf{Reconst}$ is slightly changed: it takes a list of shares as input and outputs either a secret or the special symbol $\perp$ ($\perp \notin \mathcal{S}$.) $\mathsf{Reconst}$ 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 $\mathsf{A}$ represents cheaters $P_{i_1}, \ldots, P_{i_{k-1}}$ who try to cheat $P_{i_k}$.

```
Game($\mathbf{SS}$, $\mathsf{A}$)
    $s \leftarrow \mathcal{S}$;    // according to the probability distribution over $\mathcal{S}$.
    $(v_1, \ldots, v_n) \leftarrow \mathsf{ShareGen}(s)$;
    $(i_1, \ldots, i_{k-1}) \leftarrow \mathsf{A}_1(X)$;
    // 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 $s'$ is a secret reconstructed from $v'_{i_1}, v'_{i_2}, \ldots, v'_{i_{k-1}}, v_{i_k}$ and the probability is taken over the distribution of $\mathcal{S}$ and over the random tapes of $\mathsf{ShareGen}$ and $\mathsf{A}$.

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

### 2.3 Previous Work

In this subsection, we briefly review the known bounds and constructions of $(k, n, \epsilon)$-secure secret sharing schemes.

Tompa and Woll have proposed a scheme [11] that can be proven to be a $(k, n, \epsilon_{\mathsf{CDV}})$-secure secret sharing scheme in the CDV model. Where $\mathcal{V}_i$ denotes the set of shares, the size of share $|\mathcal{V}_i|$ is as large as $(\frac{(|\mathcal{S}|-1)(k-1)}{\epsilon_{\mathsf{CDV}}} + k)^2$.

A lower bound for the size of shares in the CDV model is described as follows:

**Proposition 1.** [4] *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* $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|-1}{\epsilon_{\mathsf{OKS}}} + 1$ *.*

Within the OKS model, Ogata *et al.* have proposed a $(k, n, \epsilon_{\mathsf{OKS}})$-secure secret sharing schemes that satisfies the bound of Proposition 3 with equality [8]. However, this scheme is proven to be secure only if the secret is uniformly distributed. Within the CDV model, Cabello *et al.* have proposed a $(k, n, \epsilon_{\mathsf{CDV}})$-secure secret sharing scheme [2]. The size of share is a little longer than the lower bound of Proposition 2. Further, the scheme is secure for arbitrary secret distribution, but , in this scheme, the successful cheating probability is uniquely determined from the size of the secret. Obana *et al.* have generalized this result in [7]. In this scheme, the successful cheating probability can be chosen without regard to the size of secret.

## 3 New Model of Secret Sharing Schemes Secure Against Cheating

Some kinds of cheating are not covered by the OKS(CDV) model. For example, cheaters who know $k$ or more shares are not considered. Schemes in [2,7,8] are proven to be secure in the CDV model or OKS model. However, if cheaters know $k$ or more shares, these schemes are not secure. The successful cheating probability is one.

Actually, cheating from $k$ or more colluding participants exists. Of course, secrecy is not maintained to such participants. However, if considering detecting the fact of cheating, we need to consider a cheating from $k$ or more colluding participants. Therefore, it is highly desired to construct secret sharing schemes capable of detecting cheating from $k$ or more colluding participants with unlimited computational power. To this end, we define new models : the OKS$^{n-1}$ model and the CDV$^{n-1}$ model which are slight modifications of the OKS model and the CDV model, respectively. Cheaters in the new models are allowed to know $n-1$ shares. To characterize such cheaters, a game is defined as follows.

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

The advantage of cheaters is redefined by $Adv(\mathbf{SS}, \mathsf{B}) = \Pr[s' \in \mathcal{S} \wedge s' \neq s]$ , where $s'$ is a secret reconstructed from $v'_{i_1}, v'_{i_2}, \ldots, v'_{i_{k-1}}, v_{i_n}$ and the probability is taken over the distribution of $\mathcal{S}$ and over the random tapes of $\mathsf{ShareGen}$ and $\mathsf{B}$. In CDV$^{n-1}$ model, $s$ seems to be non-valuable information for $\mathsf{B}_2$ , because $k$

or more colluding cheaters can reconstruct secret . However, in the case of $(n, n)$ threshold structure, $s$ is valuable for $\mathsf{B}_2$ .

Please note that the CDV$^{n-1}$ model is the most powerful model of cheating. Because, now, target participant's share is the only information that cheaters don't know. Besides, please note that all the bounds for the OKS (CDV) model (e.g. Propositions 1-3) are also valid for OKS$^{n-1}$ (CDV$^{n-1}$) since a scheme secure in the OKS$^{n-1}$ (CDV$^{n-1}$) model is also secure in the OKS (CDV) model.

However, the schemes secure in the OKS(CDV) model are not necessarily secure in the OKS$^{n-1}$(CDV$^{n-1}$) model. For example, the schemes presented in [2,7,8] are not secure in the OKS$^{n-1}$(CDV$^{n-1}$) model. In these schemes, $k$ or more cheaters can know any other participant's share $v_{i_n}$. So, they can adjust $v'_{i_1}, v'_{i_2}, \ldots, v'_{i_{k-1}}$ such that reconstructed result from $v'_{i_1}, v'_{i_2}, \ldots, v'_{i_{k-1}}, v_{i_n}$ is the value which they want.

However, the schemes presented in [11] can be proven to be secure in the CDV$^{n-1}$ model.

Next, we briefly review the scheme presented in [11].

### 3.1 The Tompa and Woll Scheme[11]

The share generation algorithm $\mathsf{ShareGen}$ and the share reconstruction algorithm $\mathsf{Reconst}$ is described as follows[3].

*Share Generation.* On input a secret $s \in \{0, \ldots, |\mathcal{S}| - 1\}$, the share generation algorithm $\mathsf{ShareGen}$ outputs a list of shares $(v_1, \ldots, v_n)$ as follows. Here, $q$ is a prime such that $q > (|\mathcal{S}| - 1)(k - 1)/\epsilon + n$:

1. Generate random polynomial $f(x)$ of degree $k-1$ over $Z_q$ such that $f(0) = s$.
2. Choose $n$ distinct elements $r_1, \ldots, r_n$ uniformly and randomly from $\{1, \ldots, q - 1\}$.
3. Compute $v_i = (f(r_i), r_i)$ 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 $\mathsf{Reconst}$ outputs a secret $s$ or $\perp$ as follows:

1. Reconstruct $f(\hat{0})$ from $v_{i_1}, \ldots, v_{i_k}$ using Lagrange interpolation.
2. Output $f(\hat{0})$ if $f(\hat{0}) < |\mathcal{S}|$ holds. Otherwise $\mathsf{Reconst}$ outputs $\perp$.

In this scheme, $k$ or more cheaters can't know any other participant's share $r_{i_n}$. This scheme can be proven to be a $(k, n, \epsilon)$-secure secret sharing scheme in the CDV$^{n-1}$ model, and the size of share $|V_i|$ is $q^2 = (\frac{(|\mathcal{S}|-1)(k-1)}{\epsilon} + n)^2$. Further, the scheme is secure for arbitrary secret distribution.

[3] We made slight modification to the parameter of [11]. Because, the parameters in [11] are the parameters considering at most $k-1$ cheaters. We change the parameters to the parameters considering at most $n - 1$ cheaters.

# 4 Proposed Scheme

Tompa and Woll scheme's *Validity Check algorithm* check whether reconstructed secret is in range. This is the reason why their scheme needs very large field for polynomial which distributes secret. In proposed scheme, we use one more polynomial for distributing secret. Comparing two reconstructed secret, proposed scheme's *Validity Check algorithm* can check whether reconstructed secret is a particular value. Then, the size of the field for polynomial can be made small. Consequently, though proposed scheme uses two polynomials, the size of the share is smaller than Tompa and Woll scheme.

In this section, we propose an efficient $(k, n, \epsilon)$-secure secret sharing scheme in the CDV$^{n-1}$ model that is proven to be secure for any secret distribution.

The share generation algorithm $\mathsf{ShareGen}$ and the share reconstruction algorithm $\mathsf{Reconst}$ are described as follows where $p$ is a prime power and $q$ is a prime power such that $q > \max\ ((k-1)/\epsilon + n, p)$.

*Share Generation.* On input a secret $s \in \{0, \ldots, p-1\}$, the share generation algorithm $\mathsf{ShareGen}$ outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Generate random polynomial $f(x)$ of degree $k-1$ over $GF(q)$ such that $f(0) = s$, and $g(x)$ of degree $k-1$ over $GF(p)$ such that $g(0) = s$.
2. Choose $n$ distinct elements $r_1, \ldots, r_n$ uniformly and randomly from $\{1, \ldots r\}$, $r \leq q-1$.
3. Compute $v_i = (f(r_i), g(i), r_i)$ 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 $\mathsf{Reconst}$ outputs a secret $s$ or $\perp$ as follows:

1. Reconstruct $f\hat{(0)}$ and $g\hat{(0)}$ from $v_{i_1}, \ldots, v_{i_k}$ using Lagrange interpolation.
2. Output $f\hat{(0)}$ if $f\hat{(0)} = g\hat{(0)}$ holds. Otherwise $\mathsf{Reconst}$ outputs $\perp$.

the properties of this scheme is summarized by the following theorem.

**Theorem 1.** *The scheme of §4 is a $(k, n, \epsilon)$-secure secret sharing scheme in the CDV$^{n-1}$ model with parameters $|\mathcal{S}| = p, \epsilon = (k-1)/(r-n+1)$ and $|\mathcal{V}_i| = p \cdot q \cdot r \simeq \max\ (|S|^2(\frac{k-1}{\epsilon} + n + 1), |S|(\frac{k-1}{\epsilon} + n + 1)^2)$. Further, the scheme is secure for arbitrary secret distribution.*

*Proof.* Without loss of generality, we can assume $P_1, \ldots, P_{n-1}$ are cheaters and they try to cheat $P_n$ who has $v_n = (f_n, g_n, r_n)$ by forging their shares $v_i = (f_i, g_i, r_i)$ (for $1 \leq i \leq k-1$.)

Now, suppose that cheaters try to cheat $P_n$ by forging their shares to $v_i = (f'_i, g'_i, r'_i)$(for $1 \leq i \leq k-1$.), $(r'_1, f'_1), \ldots, (r'_{k-1}, f'_{k-1}), (r_n, f_n)$ define a polynomial $\hat{f}$ and $(1, g'_1), \ldots, (k-1, g'_{k-1}), (n, g_n)$ define a polynomial $\hat{g}$. They succeed in cheating $P_n$ if $f\hat{(0)} = g\hat{(0)}$. In the other words, they succeed in cheating if $(r'_1, f'_1), \ldots, (r'_{k-1}, f'_{k-1}), (r_n, f_n), (0, g\hat{(0)}))$ are passing through the same polynomial $f'$ of degree $k-1$ such that $f'(0) = g\hat{(0)}(\neq s)$. The cheaters can obtain polynomial $g$ from $(0, s), (1, g_1), \ldots, (k-1, g_{k-1})$. We can rewrite $g\hat{(0)}$ by

$\hat{g(0)} = L_n g(n) + \sum_{j=1}^{k-1} L_j g'_j$ ($L_j$ is a Lagrange coefficient), so cheaters can control the value $\hat{g(0)}$ as they want by adjusting their shares. Now suppose a polynomial $f'$ that is passed by the points $(r'_1, f'_1), \dots, (r'_{k-1}, f'_{k-1}), (0, \hat{g(0)}(\neq s))$. The cheaters succeed in cheating if $f'(r_n) = f(r_n)$. The $f'$ is different polynomial from $f$, because $f'(0) = \hat{g(0)} \neq s = f(0)$ . So, $f'$ can intersect $f$ in at most $k-1$ points. Here, $r_n$ is a random element of $\{1, \dots, r\} - \{r_1, \dots, r_{n-1}\}$. Thus, the probability that $f'(r_n) = f(r_n)$ is at most $(k-1)/(r-n+1)$. So $\epsilon = (k-1)/(r-n+1)$. □

## 5 Validity Check of Reconstruction Result

In previous work, participants can identify the fact of cheating only when they participate in the reconstruction.

In some situation, participants want to verify whether there was cheating from only reconstruction result. In this section, we consider the scenario that cheaters forge the reconstruction result. Such cheaters will succeed if another participants accepts an incorrect secret.

We define new models for secret sharing schemes capable if detecting such cheating. These model consist of three algorithms: ShareGen, Reconst, and a validity checking algorithm Check. The 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 pair of secret $s$ and "check data" $c$ or the special symbol $\perp$ ($\perp \notin \mathcal{S}$.) Reconst outputs $\perp$ if and only if cheating has been detected. "check data" $c$ is a value for checking the validity of the reconstructed secret. Check takes a secret $s$, check data $c$, and one share $v_i$ and outputs either a secret $s$ or the special symbol $\perp$ ($\perp \notin \mathcal{S}$.) Check outputs $\perp$ if and only if cheating has been detected. To formalize the models, we define the following simple game for threshold secret sharing scheme $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst}, \mathsf{Check})$ and for any (not necessarily polynomially bounded) Turing machine $\mathsf{C} = (\mathsf{C}_1, \mathsf{C}_2)$, where $\mathsf{C}$ represents cheaters $P_{i_1}, \dots, P_{i_{n-1}}$ who try to cheat $P_{i_n}$.

```
Game($\mathbf{SS}$, C)
    $s \leftarrow \mathcal{S}$;    // according to the probability distribution over $\mathcal{S}$.
    $(v_1, \dots, v_n) \leftarrow \mathsf{ShareGen}(s)$;
    $(i_1, \dots, i_{n-1}) \leftarrow \mathsf{C}_1(X)$;
    // set $X = s$ for the CDV$^{n-1}$ model, $X = \emptyset$ for the OKS$^{n-1}$ model.
    $(s', c') \leftarrow \mathsf{C}_2(v_{i_1}, \dots, v_{i_{n-1}}, X)$;
```

The advantage of cheaters is expressed as $Adv(\mathbf{SS}, \mathsf{C}) = \Pr[s' \in \mathcal{S} \wedge s' \neq s]$ , where $s' = \mathsf{Check}(s', c', v_{i_n})$ and the probability is taken over the distribution of $\mathcal{S}$ and over the random tapes of ShareGen and C.

**Definition 2.** *A $(k, n)$ threshold secret sharing scheme* **SS** *is called a $(k, n,$ $\epsilon_1, \epsilon_2)$* -secure secret sharing scheme with Validity check of reconstruction result *if* $Adv(\mathbf{SS}, \mathsf{B}) \leq \epsilon_1$ *for any adversary* $\mathsf{B}$ *and* $Adv(\mathbf{SS}, \mathsf{C}) \leq \epsilon_2$ *for any adversary* $\mathsf{C}$.

Easily, we can construct a $(k, n, \epsilon_1, \epsilon_2)$-secure secret sharing scheme with Validity check of reconstruction result from the scheme of Section 4.

Using reconstruction algorithm which outputs all inputs as check data, all participants can check the validity of a reconstruction result by inputing $k - 1$ shares from check data and a share which they have to the reconstruction algorithm.

But, in this scheme, the size of check data is very large. However, by slight modification to the scheme of Section 4, we can construct more efficient scheme.

### 5.1 Modified Proposed Scheme

In this section, we propose a $(k, n, \epsilon_1, \epsilon_2)$-secure secret sharing scheme with Validity check of reconstruction result. This scheme is a slightly modified scheme of the scheme of Section 4 and the check data is much smaller than trivial scheme.

The share generation algorithm ShareGen, the share reconstruction algorithm Reconst, and the validity checking algorithm Check are described as follows where $p$ is a prime power and $q$ is a prime power such that $q > \max\ ((k-1)/\epsilon_l + n, p)$ (for $l = 1, 2$).

*Share Generation.* On input a secret $s \in \{0, \dots, p-1\}$, the share generation algorithm ShareGen outputs a list of shares $(v_1, \dots, v_n)$ as follows:

1. Generate random polynomial $f(x)$ of degree $k - 1$ over $GF(q)$ such that $f(0) = s$, and $g(x)$ of degree $k - 1$ over $GF(p)$ such that $g(0) = s$.
2. Choose $n$ distinct elements $r_1, \dots, r_n$ uniformly and randomly from $\{1, \dots r\}$ $r \leq q - 1$.
3. Compute $v_i = (f(r_i), g(i), r_i)$ 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 Reconst outputs a secret $s$ or $\perp$ as follows:

1. Reconstruct $\hat{f}$ and $\hat{g(0)}$ from $v_{i_1}, \dots, v_{i_k}$ using Lagrange interpolation.
2. Output $\hat{f(0)}$ as secret and $\hat{f}$ as check data if $\hat{f(0)} = \hat{g(0)}$ holds. Otherwise Reconst outputs $\perp$.

*Validity check of Reconstruction result.* On input a polynomial $f(x)$ of degree $k - 1$ over $GF(q)$ and a share $v_i = (f_i, g_i, r_i)$, the validity checking algorithm Check outputs a secret $s$ or $\perp$ as follows:

- Output $f(0)$ if $f(r_i) = f_i$ holds. Otherwise Reconst outputs $\perp$.

In this validity check algorithm, $f$ can be regarded not only as secret but also as check data.

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

**Theorem 2.** *The scheme of* §*5.1 is* $(k, n, \epsilon_1, \epsilon_2)$*-secure secret sharing scheme in the CDV*$^{n-1}$ *model with parameters* $|\mathcal{S}| = p, \epsilon_1 = \epsilon_2 = (k-1)/(r-n+1)$*, and* $|\mathcal{V}_i| = p \cdot q \cdot r \simeq max\ (|S|^2(\frac{k-1}{\epsilon_l} + n + 1), |S|(\frac{k-1}{\epsilon_1} + n + 1)^2)$ *. Further, the scheme is secure for arbitrary secret distribution.*

*Proof.* Firstly, $\epsilon_1$ is proven to be $(k-1)/(r-n+1)$ by similar discussion to the proof of Theorem 1. Next, we will show that $\epsilon_2 = (k-1)/(r-n+1)$. Without loss of generality, we can assume $P_1, \ldots, P_{n-1}$ are cheaters and they try to cheat $P_n$ who has $v_n = (f_n, g_n, r_n)$ by forging their check data to $f'$ such that $f'(0) \neq s$.

They succeed in cheating $P_n$ if $f'(r_n) = f_n$. In other words, they succeed in cheating $P_n$ if $f'(r_n) = f(r_n)$. The $f'$ is different polynomial from $f$, because $f'(0) \neq s$. Here, $r_n$ is a random element of $\{1, \ldots, r\} - \{r_1, \ldots, r_{n-1}\}$. Thus, the probability that $f'(r_n) = f(r_n)$ is at most $(k-1)/(r-n+1)$. So $\epsilon_2 = (k-1)/(r-n+1)$. □

In proposed scheme, the size of check data is only one polynomial representation of degree $k-1$ over $GF(q)$. This is much smaller than the check data of trivial scheme.

## 6 Conclusion

In this paper, we proposed an efficient $(k, n)$ threshold secret sharing scheme capable of detecting cheating from $n-1$ or less colluding participants.

Table 1 and Table 2 below compares the bit length of shares for the various security parameters where the access structure considered is 3-out-of-5 threshold access structure.

Compared to the scheme of [11] the size of the share in the proposed scheme is smaller for all the security parameters. When $|\mathcal{S}| < 1/\epsilon$ and $k$,$n$ are small,

**Table 1.** Comparison of the bit length of the shares (for $\epsilon = 2^{-128}$)

| $\lvert S \rvert$ | Known Bound | Proposed Scheme | Tompa and Woll |
|---|---|---|---|
| $2^{64}$ | 321 | 324 | 388 |
| $2^{128}$ | 385 | 388 | 516 |
| $2^{256}$ | 503 | 642 | 772 |
| $2^{512}$ | 769 | 1154 | 1284 |

**Table 2.** Comparison of the bit length of the shares (for $\epsilon = 2^{-256}$)

| $\lvert S \rvert$ | Known Bound | Proposed Scheme | Tompa and Woll |
|---|---|---|---|
| $2^{64}$ | 577 | 580 | 644 |
| $2^{128}$ | 641 | 644 | 772 |
| $2^{256}$ | 769 | 772 | 1026 |
| $2^{512}$ | 1025 | 1282 | 1540 |

the size of the share in the proposed scheme is a few bits longer than the lower bound of [7].

Finding more efficient $(k, n, \epsilon)$-secure secret sharing schemes in the CDV$^{n-1}$ model will be future work.

## Acknowledgement

We thank the anonymous referees for useful and datailed comments.

## References

1. Blakley, G.R.: Safeguarding cryptographic keys. In: Proc. AFIPS 1979, National Computer Conference, vol. 48, pp. 313–137 (1979)
2. Cabello, S., Padró, C., Sáez, G.: Secret Sharing Schemes with Detection of Cheaters for a General Access Structure. Designs, Codes and Cryptography 25(2), 175–188 (2002)
3. Carpentieri, M.: A Perfect Threshold Secret Sharing Scheme to Identify Cheaters. Designs, Codes and Cryptography 5(3), 183–187 (1995)
4. Carpentieri, M., De Santis, A., Vaccaro, U.: Size of Shares and Probability of Cheating in Threshold Schemes. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 118–125. Springer, Heidelberg (1994)
5. Cramer, R., Damgård, I., Maurer, U.M.: General Secure Multi-party Computation from any Linear Secret-Sharing Scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000)
6. Kurosawa, K., Obana, S., Ogata, W.: $t$-Cheater Identifiable $(k, n)$ Secret Sharing Schemes. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 410–423. Springer, Heidelberg (1995)
7. Obana, S., Araki, T.: Almost Optimum Secret Sharing Schemes Secure Against Cheating for Arbitrary Secret Distribution. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 364–379. Springer, Heidelberg (2006)
8. Ogata, W., Kurosawa, K., Stinson, D.R.: Optimum Secret Sharing Scheme Secure against Cheating. SIAM Journal on Discrete Mathematics 20(1), 79–95 (2006)
9. Pedersen, T.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–149. Springer, Heidelberg (1992)
10. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
11. Tompa, M., Woll, H.: How to Share a Secret with Cheaters. Journal of Cryptology 1(3), 133–138 (1989)