Almost Optimal Cheating-Detectable (2, 2, n) Ramp Secret Sharing Scheme

上松, 知貴 / Agematsu, Tomoki

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
15
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2020-03-24
(URL)
https://doi.org/10.15002/00022721

Almost Optimal Cheating-Detectable $(2, 2, n)$ Ramp Secret Sharing Scheme
不正検知可能な準最適 $(2, 2, n)$ ランプ型秘密分散

上松 知貴 *
Tomoki Agematsu
法政大学大学院 情報科学研究科 情報科学専攻
Email: 18t0002@cis.k.hosei.ac.jp

***Abstract*—In this research, we consider a strong ramp secret sharing scheme that can detect cheating. A cheating-detectable (k, L, n) ramp secret sharing scheme has been studied so far, and a strong ramp secret sharing scheme which achieves lower bounds on the size of shares and random number used in encoding (i. e., share generation), and the success probability of impersonation attack has been presented. Now a challenging task is to achieve the lower bound on the success probability of substitution attack.**

In this paper, we present a strong $(2, 2, n)$ ramp secret sharing scheme that almost achieves the lower bound on the success probability of substitution attack. The proposed scheme is the first to almost achieve the lower bound. Moreover the proposed scheme also achieves other lower bounds such as those on the size of shares and random number used in encoding, and the success probability of impersonation attack. We take a unique strategy to construct the scheme. Most existing works present generic type verification functions which can detect cheating for any linear and strong (k, L, n) ramp scheme. On the other hand, our proposed verification function (one of those which we call limited type verification functions) can detect cheating when used with a linear and strong $(2, 2, n)$ ramp scheme satisfying a certain property.

1. Introduction

In secret sharing schemes (SSSs for short), a secret is divided into multiple shares in a way that only qualified sets of shares can recover the secret. Therefore, secret sharing plays an important role for preventing information leakage. In addition, the risk of information loss can be reduced because the secret can be recovered from remaining shares in case some shares are lost. Further, secret sharing draws a lot of attention as a building block for secure multiparty computation. In (k, n) SSSs [1],[2], a secret is divided into n shares in a way that any k shares uniquely determine the secret, while less than k shares obtain no information concerning the secret. An important variant of (k, n) SSSs is a (k, L, n) ramp SSS [3] which gradually reveals information concerning a secret from $k - t$ $(1 \leq t \leq L - 1)$ shares. The merit of (k, L, n) ramp SSSs is in its small share size. The size of each share is $\frac{1}{L}$ of the size of a secret while the size of each share is the same as that of the secret in (k, n) SSSs.

In SSSs, when some shares are forged, a secret recovered from them becomes an incorrect value. Such attacks can be classified into two types, impersonation attacks that generate a forged share without knowing a correct share and substitution attacks that generate a forged share using a correct share. SSSs that can detect these attacks have been studied extensively so far.

In (k, n) SSSs, Cabello et al. [4] have proposed methods which modify any linear SSS to cheating-detectable schemes. Ogata et al. [5] have derived the lower bound on the size of shares for a given success probability of substitution attack and have proposed a cheating-detectable (k, n) SSS which achieves the lower bound on the size of shares. Also, Cramer et al. [6] have introduced algebraic manipulation detection (AMD) codes. By applying AMD codes to an arbitrary linear SSS, it is converted to a cheating-detectable scheme. Their construction flexibly accommodates arbitrary choices of security level and the cardinality of space of a secret. In (k, L, n) ramp SSSs, Nakamura et al. [7], [8] have proposed a cheating-detectable strong (k, L, n) ramp SSS. Their proposed scheme achieves lower bounds on the size of shares and random number used in encoding, and the success probability of impersonation attack. However, when the number of forged shares is less than k, the success probability of substitution attack is nearly L times the lower bound, therefore, there is room to improve the scheme.

In this paper, we present a cheating-detectable strong $(2, 2, n)$ ramp SSS which achieves lower bounds on the size of shares, the size of random number used in encoding, and the success probability of impersonation attack, and almost achieves the lower bound on the success probability of substitution attack. Our scheme can be applied to a secret $S^2 \in GF(p^m)^2$, where p is a prime number. We suppose that an attacker can forge up to $k-1$ shares, computation power of an attacker is unbounded, and an attacker does not know a secret S^2 (OKS model). A well-known technique to achieve cheating detection is to introduce a verification function with which the correctness of a recovered secret is verified. The most unique part of our research is that we introduce a notion of "limited type" verification functions. A limited type verification function is a function which can detect attack when a generator matrix of a ramp SSS satisfying certain conditions is used. In our scheme, $S_1 \cdot S_2$ is used as a verification function and a generator matrix of a strong $(2, 2, n)$ ramp SSS satisfying a certain condition (shown in Section 3) is used. We also show that such generator matrices exist, and that most generator matrices satisfy the condition, when $p^m \gg 2n - 1$ holds. To the best of our knowledge, our scheme is the first to almost achieve the lower bound on the success probability of substitution attack and also the first to introduce a notion of limited type verification functions.

* Supervisor: Prof. Satoshi Obana

With the notion, functions that could not be used as verification functions in a conventional notion of verification functions can be used as verification functions if there are conditions which functions can guarantee security against attack. However, applying our strategy to more generalized parameters is probably difficult. We discuss it in the full version of this paper.

2. Preliminaries

In this paper, for a subset $\mathcal{J}=\{i_1,...,i_j\}\subseteq\{1,...,n\}$, $X_{\mathcal{J}}$ denotes $(X_{i_1},...,X_{i_j})$. $H_p(\cdot)$ denotes entropy with base p in logarithm (the base p of $H_p(\cdot)$ is omitted for simplicity). Also, $I_p(\cdot;\cdot)$ denotes mutual information with base p in logarithm. Furthermore, $i_\ell\neq i_{\hat{\ell}}$, for $\ell\neq\hat{\ell}$ in $\{i_1,\ldots,i_k\}$ is assumed.

2.1. (k,L,n) Ramp SSSs

We describe (k,L,n) ramp SSSs. Let $S^L=S_1S_2...S_L$ be a secret, and all of them $(S_j,1\leq j\leq L)$ are mutually independent. Further, these have the same probability distribution P_S over a finite set $\mathcal{S}$. We introduce the encoder and the decoder. The encoder ϕ which generates shares is defined as a function $\phi:\mathcal{S}^L\times\mathcal{R}\to\mathcal{V}_1\times\mathcal{V}_2\times\cdots\times\mathcal{V}_n$, namely, $(V_1,...,V_n)=\phi(S^L,R)$. Here, $\mathcal{V}_i$ is the range of the share V_i and R is a uniform random number over a finite set $\mathcal{R}$. The decoder ψ_K is defined as a function $\psi_K:\mathcal{V}_{i_1}\times\cdots\times\mathcal{V}_{i_k}\to\mathcal{S}^L\cup\{\perp\}$ for each $K=\{i_1,...,i_k\}\subseteq\{1,...,n\}$ (in this paper, we consider the case that a secret is recovered from just k shares, and to treat a cheating-detectable scheme, we introduce a symbol $\perp$ that means "detect forgery"). K of ψ_K is omitted for simplicity. If (ϕ,ψ) satisfies the following conditions (i) and (ii), it is called a (k,L,n) ramp SSS. When $L=1$, (ϕ,ψ) is a (k,n) SSS. On the other hand, (ϕ,ψ) satisfies all of following conditions, it is called a strong (k,L,n) ramp SSS (if (ϕ,ψ) satisfies (i) and (ii) but does not satisfy (iii), it is called a weak (k,L,n) ramp SSS).

(i) For any $\{i_1,...,i_k\}\subseteq\{1,...,n\}$, the following holds.
$$\psi(V_{i_1},...,V_{i_k})=S^L$$

(ii) For any $t\in\{1,...,L\}$ and any $\mathcal{I}\subseteq\{1,...,n\}$ $(|\mathcal{I}|=k-t)$, the following holds.
$$H(S^L|V_{\mathcal{I}})=\frac{t}{L}H(S^L)$$

(iii) For any $t\in\{1,...,L\}$, any $\mathcal{I}\subseteq\{1,...,n\}$ $(|\mathcal{I}|=k-t)$, and any $\mathcal{J}\subseteq\{1,...,L\}$ $(|\mathcal{J}|=t)$, the following holds.
$$H(S_{\mathcal{J}}|V_{\mathcal{I}})=H(S_{\mathcal{J}})$$

In particular, from [3], shares are obtained as follows for a secret $S^L\in GF(p^m)^L$, where p^m satisfies $(k<p^m,\ n\leq p^m-L+1)$ or $(n=k\geq p^m,L=1)$.
$$[V_1\cdots V_n]=[S_1\cdots S_L\ R_1\cdots R_{k-L}]A$$
$A\in GF(p^m)^{k\times n}$ is called a generator matrix of a (k,L,n) ramp SSS. If A is a generator matrix of a strong (k,L,n) ramp SSS, any k columns $\{\boldsymbol{c}_1\cdots\boldsymbol{c}_k\}$ chosen from $\{\boldsymbol{e}_1\cdots\boldsymbol{e}_L\ \boldsymbol{a}_1\cdots\boldsymbol{a}_n\}$, where $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n$ are n columns of A and $\boldsymbol{e}_1\cdots\boldsymbol{e}_k$ are k columns of $k\times k$ identity matrix, satisfy $rank[\boldsymbol{c}_1\cdots\boldsymbol{c}_k]=k$.

2.2. Successful Cheating Probabilities and Lower Bounds

We show the definition of successful cheating probabilities of cheating-detectable schemes. Let a $(1\leq a\leq k-1)$ be the number of forged shares, let $\bar{V}_{\mathcal{O}}$ and $V_{\mathcal{O}},\mathcal{O}=\{i_1,...,i_a\}$ be forged shares and corresponding correct shares, respectively, and let $V_{\mathcal{I}},\mathcal{I}=\{i_{a+1},...,i_k\}$ be remaining shares satisfying $|\mathcal{I}|=k-a$ and $\mathcal{O}\cap\mathcal{I}=\emptyset$. An impersonation attack is an attack that an attacker generates $\bar{V}_{\mathcal{O}}$ without knowing $V_{\mathcal{O}}$, namely, $\bar{V}_{\mathcal{O}}$ is independent of $(V_{\mathcal{O}},V_{\mathcal{I}})$. A substitution attack is an attack that an attacker generates $\bar{V}_{\mathcal{O}}$ using $V_{\mathcal{O}}$, that is, $V_{\mathcal{I}}$, $V_{\mathcal{O}}$ and $\bar{V}_{\mathcal{O}}$ make a Markov chain in this order. In impersonation attacks, there are two definitions of the success of attack. One is $\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\neq\perp$, and the other is $\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\notin\{S^L,\perp\}$. On the other hand, in substitution attacks, $\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\notin\{S^L,\perp\}$ is the only meaningful definition. The success probabilities of impersonation attack $\left(P_{imp^*(a)},P_{imp(a)}\right)$ and substitution attack $\left(P_{sub(a)}\right)$ are as follows.

Definition 1. The successful cheating probabilities
$$P_{imp^*(a)}=\max_{\substack{\mathcal{O},\mathcal{I}\subseteq\{1,...,n\}:\\|\mathcal{O}|=a,|\mathcal{I}|=k-a,\\\mathcal{O}\cap\mathcal{I}=\emptyset}}\ \max_{P_{\bar{V}_{\mathcal{O}}}}\Pr\{\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\neq\perp\}$$
$$P_{imp(a)}=\max_{\substack{\mathcal{O},\mathcal{I}\subseteq\{1,...,n\}:\\|\mathcal{O}|=a,|\mathcal{I}|=k-a,\\\mathcal{O}\cap\mathcal{I}=\emptyset}}\ \max_{P_{\bar{V}_{\mathcal{O}}}}\Pr\{\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\notin\{S^L,\perp\}\}$$
$$P_{sub(a)}=\max_{\substack{\mathcal{O},\mathcal{I}\subseteq\{1,...,n\}:\\|\mathcal{O}|=a,|\mathcal{I}|=k-a,\\\mathcal{O}\cap\mathcal{I}=\emptyset}}\ \max_{v_{\mathcal{O}}\in\mathcal{V}_{\mathcal{O}}}\max_{P_{\bar{V}_{\mathcal{O}}|V_{\mathcal{O}}}}\Pr\{\psi(\bar{V}_{\mathcal{O}},V_{\mathcal{I}})\notin\{S^L,\perp\}|V_{\mathcal{O}}=v_{\mathcal{O}}\}$$

Next, we describe the definition of correlation level.

Definition 2. Correlation level
The correlation level of $(V_1,...,V_n)$ is defined as $(l_1,...,l_{k-1})_p$ if for any $j\in\{2,...,k\}$ and any $\{i_1,...,i_j\}\subseteq\{1,...,n\}$, it holds that
$$I_p(V_{i_1};V_{i_2}|V_{i_3},...,V_{i_j})=l_{j-1}.$$
For $j=2$, $I_p(V_{i_1};V_{i_2})=l_1$.

Nakamura et al. [8] have derived lower bounds of (k,L,n) ramp SSSs.

Proposition 1. For any (k,L,n) ramp SSS with correlation level $(l_1,...,l_{k-1})$,
$$\log|\mathcal{V}_i|\geq\frac{1}{L}H(S^L)+\sum_{j=1}^{k-1}l_j,\ i=1,...,n,$$
$$\log|\mathcal{R}|\geq\frac{k-L}{L}H(S^L)+\sum_{j=1}^{k-1}jl_j,$$
$$\log P_{imp^*(a)}\geq-\sum_{j=1}^{a}\sum_{j'=1}^{k-a}l_{j+j'-1},\ a=1,...,k-1,$$
$$\log P_{imp(a)}\geq-\sum_{j=1}^{a}\sum_{j'=1}^{k-a}l_{j+j'-1}+\log(1-Q_{max,L}),$$
$$a=L,...,k-1,(Q_{max,L}:=\max_{S^L\in\mathcal{S}^L}P_{S^L}(s^L)).$$

In addition, in a strong (k, L, n) ramp SSS,

$$\log P_{imp(a)} \geq -\sum_{j=1}^{a}\sum_{j'=1}^{k-a} l_{j+j'-1} + \log\left(1-(Q_{max})^a\right),$$
$$a = 1, ..., L-1, (Q_{max} := \max_{S\in\mathcal{S}} P_S(s)).$$

Furthermore, when S_j is uniformly distributed over $\mathcal{S}$, the following holds in a strong (k, L, n) ramp SSS.

$$P_{sub(a)} \geq \frac{|\mathcal{S}|-1}{|\mathcal{V}_i|}, a = 1, ..., k-1, \text{ for any } i\in\{1, \ldots, n\} \quad (1)$$

However, we note that the bound (1) is not tight. Proposition 1 holds for any base $p > 1$ of logarithm.

Next, we show a cheating-detectable strong (k, L, n) ramp SSS proposed by Nakamura et al. [8].

2.3. Cheating-Detectable Strong (k, L, n) Ramp SSS

Their scheme can be applied to a secret S^L and can detect substitution attacks of up to $k-1$ shares. Also, OKS model is supposed. Their verification function is $h(S_1, S_2, \ldots, S_L) = \sum_{j=1}^{L}(S_j)^{j+1}$.

Suppose that each S_j is uniformly distributed over $\mathcal{S} = GF(p^m)$. Here, m is a positive integer and p is a prime number that satisfies $p \geq L+2$. Let $l\in\{1, ..., m\}$. Then, assume that the followings hold.

$$(k < p^m,\ n \leq p^m - L + 1) \text{ or } (n = k \geq p^m, L = 1)$$
$$(k < p^l,\ n \leq p^l) \text{ or } (n = k \geq p^l)$$

Let f be a surjective linear mapping ($f : GF(p^m)\to GF(p^l)$). It satisfies following two properties.

$$\forall x_1, x_2\in GF(p^m),\ f(x_1 + x_2) = f(x_1) + f(x_2) \quad (2)$$

$$\forall y\in GF(p^l),\ |\{x\in GF(p^m) : f(x) = y\}| = p^{m-l} \quad (3)$$

Encoding is performed as follows. Shares are defined as $V_i := (W_i, U_i)$ $(1\leq i\leq n)$ where $W_i\in GF(p^m)$ is a share of S^L obtained by a linear strong (k, L, n) ramp SSS, and $U_i\in GF(p^l)$ is a share of $f(\sum_{j=1}^{L}(S_j)^{j+1})$ obtained by a linear (k, n) SSS. In particular, shares are given by

$$\begin{bmatrix} W_1 & \cdots & W_n \end{bmatrix} = \begin{bmatrix} S_1 & \cdots & S_L & R_1 & \cdots & R_{k-L} \end{bmatrix} A,$$

$$\begin{bmatrix} U_1 & \cdots & U_n \end{bmatrix} = \begin{bmatrix} f(\sum_{j=1}^{L}(S_j)^{j+1}) & R'_1 & \cdots & R'_{k-1} \end{bmatrix} B.$$

Here, $(R_1, ..., R_{k-L}, R'_1, ..., R'_{k-1})$ is a uniform random number over $GF(p^m)^{k-L}\times GF(p^l)^{k-1}$, $A\in GF(p^m)^{k\times n}$ is a generator matrix of a strong (k, L, n) ramp SSS, and $B\in GF(p^l)^{k\times n}$ is a generator matrix of a (k, n) SSS.

Decoding is performed as follows. Let $\hat{V}_{i_1} = (\hat{W}_{i_1}, \hat{U}_{i_1}), ..., \hat{V}_{i_k} = (\hat{W}_{i_k}, \hat{U}_{i_k})$ be the input of the decoder, let $\boldsymbol{a}_1, ..., \boldsymbol{a}_n$ be columns of A, and let $\boldsymbol{b}_1, ..., \boldsymbol{b}_n$ be columns of B. Then, define $C\in GF(p^m)^{k\times k}$ and $D\in GF(p^l)^{k\times k}$ as

$$C = (c_{ij}) := \begin{bmatrix} \boldsymbol{a}_{i_1} & \cdots & \boldsymbol{a}_{i_k} \end{bmatrix}^{-1},$$

$$D = (d_{ij}) := \begin{bmatrix} \boldsymbol{b}_{i_1} & \cdots & \boldsymbol{b}_{i_k} \end{bmatrix}^{-1}.$$

From the encoding procedure, the followings hold for correct shares $(W_{i_1}, U_{i_1}), \ldots, (W_{i_k}, U_{i_k})$.

$$\begin{bmatrix} S_1 & \cdots & S_L & R_1 & \cdots & R_{k-L} \end{bmatrix} = \begin{bmatrix} W_{i_1} & \cdots & W_{i_k} \end{bmatrix} C$$

$$\begin{bmatrix} f(\sum_{j=1}^{L}(S_j)^{j+1}) & R'_1 & \cdots & R'_{k-1} \end{bmatrix} = \begin{bmatrix} U_{i_1} & \cdots & U_{i_k} \end{bmatrix} D$$

Thus, the decoder checks whether

$$f\left(\sum\nolimits_{j=1}^{L}\left(\sum\nolimits_{\ell=1}^{k} c_{\ell j}\hat{W}_{i_\ell}\right)^{j+1}\right) = \sum\nolimits_{\ell=1}^{k} d_{\ell 1}\hat{U}_{i_\ell}$$

holds or not. If it holds, the decoder outputs $\hat{S}^L$ where

$$\hat{S}_j = \sum\nolimits_{\ell=1}^{k} c_{\ell j}\hat{W}_{i_\ell}, j = 1, ..., L.$$

If it is not satisfied, the decoder outputs $\perp$.

Proposition 2. Their proposed scheme is a strong (k, L, n) ramp SSS with correlation level $(0, ..., 0, l)_p$, and the size of shares, the size of random number used in encoding, and successful cheating probabilities are as follows.

$$\log_p |\mathcal{V}_i| = m + l,\ i = 1, ..., n,$$
$$\log_p |\mathcal{R}| = (k-L)m + (k-1)l,$$
$$P_{imp^*(a)} = p^{-l},\ a = 1, ..., k-1,$$
$$P_{imp(a)} = p^{-l}(1 - p^{-m\cdot\min\{a,L\}}),\ a = 1, ..., k-1,$$
$$P_{sub(a)} \leq Lp^{-l},\ a = 1, ..., k-1.$$

For any correlation level $(0, ..., 0, l)_p$, their scheme achieves the lower bounds of (k, L, n) ramp SSSs (or strong (k, L, n) ramp SSSs) on the size of shares, the size of random number used in encoding, $P_{imp^*(a)}$, and $P_{imp(a)}$. In addition, $P_{sub(a)}$ is nearly L times the lower bound of strong (k, L, n) ramp SSSs.

3. Proposed Scheme

We propose a cheating-detectable strong $(2, 2, n)$ ramp SSS which almost achieves the lower bound on $P_{sub(a)}$. We employ $S_1\cdot S_2$ as a verification function, which is a limited type. We define a limited type verification function as a function which can guarantee security against attack, when using a generator matrix of a (k, L, n) ramp SSS satisfying certain conditions. On the other hand, we define a generic type verification function as a function which can guarantee security against attack for arbitrary generator matrices of (k, L, n) ramp SSSs. The verification function used in [8] is the generic type in our definition. Limited types are the same as generic types, except that applicable generator matrices are "limited".

First, we show the condition of a generator matrix of a strong $(2, 2, n)$ ramp SSS with which our verification function detects substitution attacks. Second, we show that there are generator matrices satisfying the condition and show the number of such generator matrices. Finally, we show our scheme.

3.1. Condition of Strong $(2,2,n)$ Ramp SSSs

We show the condition of a generator matrix. To achieve cheating detection, the verification function $h(S_1, S_2) = S_1 \cdot S_2$ is used, and also $b = S_1 \cdot S_2$ is divided into shares by using a $(2,n)$ SSS. Furthermore, in verification, the decoder checks whether $\hat{S}_1 \cdot \hat{S}_2 = \hat{b}$ holds or not, where $\hat{S}_i,\ i \in \{1,2\}$ is a recovered secret and $\hat{b}$ is a recovered b. From the fact that $S_1 \cdot S_2 = b$ holds, an attacker needs to satisfy $\hat{S}_1 \cdot \hat{S}_2 - S_1 \cdot S_2 = \hat{b} - b$ with a forged value. In this research, an attacker can know and forge one share. Since a $(2,n)$ SSS is used to divide b, an attacker can manipulate the value of $\hat{b} - b$. Thus, to detect forgery, $\hat{S}_1 \cdot \hat{S}_2 - S_1 \cdot S_2$ must be a function of a share (of a secret) which is unknown to an attacker (i.e., an attacker cannot manipulate the value of it). Now, we consider a strong $(2,2,n)$ ramp SSS which can be applied to a secret $S^2 \in GF(p^m)^2$ (p is a prime number and m is a positive integer). Here, $2 < p^m$ and $n \leq p^m - 1$ hold. In particular, shares are given by

$$[W_1\ W_2\ \cdots\ W_n] = [S_1\ S_2]X \tag{4}$$

where $X \in GF(p^m)^{2\times n}$ is a generator matrix of a strong $(2,2,n)$ ramp SSS. From (4), the following holds

$$[S_1\ S_2] = [W_{i_1}\ W_{i_2}][\boldsymbol{x}_{i_1}\ \boldsymbol{x}_{i_2}]^{-1}$$

for any two correct shares W_{i_1} and W_{i_2} ($\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are columns of X).

Assume that an attacker forges W_{i_1}. Let $\bar{W}_{i_1} = W_{i_1} + \delta_1$ ($\delta_1 \in GF(p^m)$) and W_{i_2} be the input of the decoder. Then,

$$\bar{S}_1 = x'_{11}(W_{i_1} + \delta_1) + x'_{21}W_{i_2}$$
$$\bar{S}_2 = x'_{12}(W_{i_1} + \delta_1) + x'_{22}W_{i_2}$$

hold, where $X' = (x'_{ij}) := [\boldsymbol{x}_{i_1}\ \boldsymbol{x}_{i_2}]^{-1}$. In addition, the following is obtained.

$$\begin{aligned}&\bar{S}_1 \cdot \bar{S}_2 - S_1 \cdot S_2 \\ &= (x'_{11}x'_{22} + x'_{12}x'_{21})\delta_1 W_{i_2} + x'_{11}x'_{12}\delta_1(2W_{i_1} + \delta_1)\end{aligned} \tag{5}$$

When $(x'_{11}x'_{22} + x'_{12}x'_{21}) = 0$ holds, (5) is not a function of W_{i_2}, and an attacker can manipulate the value of (5). On the other hand, if $(x'_{11}x'_{22} + x'_{12}x'_{21}) \neq 0$ holds, (5) becomes a linear polynomial in W_{i_2}. Thus, it is the condition of a generator matrix of a strong $(2,2,n)$ ramp SSS with which $S_1 \cdot S_2$ detects forgery (of course, in the case $\delta_1 = 0$, the term $(x'_{11}x'_{22} + x'_{12}x'_{21})\delta_1 W_{i_2}$ becomes 0 even if $(x'_{11}x'_{22} + x'_{12}x'_{21}) \neq 0$ holds, however there is no forgery). Naturally, the condition is the same in the case W_{i_2} is forged. In this paper, we call the generator matrices satisfying the condition $(x'_{11}x'_{22} + x'_{12}x'_{21} \neq 0)$ "generator matrices for securing $S_1 \cdot S_2$".

3.2. Number of Generator Matrices of Strong $(2,2,n)$ Ramp SSSs for Securing $S_1 \cdot S_2$

We show the number of generator matrices of strong $(2,2,n)$ ramp SSSs satisfying the condition (for securing $S_1 \cdot S_2$) through the process of constructing such matrices. From [3, Theorem 2, 3] and the condition, in order for $2\times n$ matrix $X \in GF(p^m)^{2\times n}$ (where p is a prime number, m is a positive integer, and $n \geq 2$) to be a generator matrix satisfying the condition, the following conditions must be satisfied.

(c1) All elements of X are not 0.

(c2) An arbitrary 2×2 matrix $M = (m_{ij})$ which is consisted of any two columns of X has an inverse matrix. In other words, $m_{11}m_{22} - m_{12}m_{21} \neq 0$ holds.

(c3) The inverse matrix of M (we denote it $M' = (m'_{ij})$) satisfies the condition for securing $S_1 \cdot S_2$, namely, $m'_{11}m'_{22} + m'_{12}m'_{21} \neq 0$ holds.

Obviously, $(c1) \wedge (c2)$ is the necessary and sufficient condition for X to be a generator matrix of a strong $(2,2,n)$ ramp SSS. Further, $(c1) \wedge (c2) \wedge (c3)$ is the necessary and sufficient condition for X to be a generator matrix of a strong $(2,2,n)$ ramp SSS for securing $S_1 \cdot S_2$.

We show the following theorem about the number of matrices (which are elements of $GF(p^m)^{2\times n}$) satisfying the above conditions.

Theorem 1. If $2n - 1 < p^m$ (p is a prime number greater than or equal to 3) holds, there are

$$\prod_{t=1}^{n}\{(p^m - 1)^2 - 2(t-1)(p^m - 1)\}$$

$2\times n$ matrices satisfying conditions (c1) to (c3). Further, in the case $p = 2$, if $n < 2^m$ holds, there are

$$\prod_{t=1}^{n}\{(2^m - 1)^2 - (t-1)(2^m - 1)\}$$

$2\times n$ matrices satisfying conditions (c1) to (c3). On the other hand, if these are not satisfied, there is no $2\times n$ matrix satisfies conditions (c1) to (c3).
Moreover, if $n < p^m$ (p is a prime number) holds, there are

$$\prod_{t=1}^{n}\{(p^m - 1)^2 - (t-1)(p^m - 1)\}$$

$2\times n$ matrices satisfying conditions (c1) and (c2). On the other hand, if it is not satisfied, there is no $2\times n$ matrix satisfies conditions (c1) and (c2).

We show the proof of Theorem 1 in the full version of this paper. Now, we show a simple example of X which satisfies (c1) to (c3).

A simple example. Such X can be easily obtained by constructing the following matrix.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix} \in GF(p^m)^{2\times n} \tag{6}$$

Here, (6) satisfies that x_1 to x_n are non-zero, $x_i \neq x_j$ (for $i \neq j$), and any x_i ($i \in \{1, ..., n\}$) is not additive inverse of any x_j ($j \in \{1, ..., n\} \backslash \{i\}$) over $GF(p^m)$. When $p \geq 3$, if $2n - 1 < p^m$ holds, such a matrix exists, obviously. When $p = 2$, it exists if $n < 2^m$ holds.

From Theorem 1, when $p \geq 3$, the ratio of the number of generator matrices of strong $(2,2,n)$ ramp SSSs for

securing $S_1 \cdot S_2$ to the number of generator matrices of strong $(2,2,n)$ ramp SSSs is

$$\prod_{t=1}^{n} \frac{p^m-(2t-1)}{p^m-t} = \prod_{t=1}^{n}\left(1-\frac{t-1}{p^m-t}\right).$$

If $p^m \gg 2n-1$ holds, the ratio is close to 1 and it shows that the majority of generator matrices are generator matrices for securing $S_1 \cdot S_2$. In the case $p=2$, all generator matrices are generator matrices for securing $S_1 \cdot S_2$.

3.3. Almost Optimal Cheating-Detectable Strong $(2,2,n)$ Ramp SSS using $S_1 \cdot S_2$

We show our scheme. Our proposed scheme can be applied to $S^2 \in GF(p^m)^2$. Here, m is a positive integer and p is a prime number. We suppose that each S_j is uniformly distributed over $\mathcal{S} = GF(p^m)$. In our scheme, there is no restriction on p (e.g., the scheme of [8] has the restriction $p \geq L+2$) because of using $S_1 \cdot S_2$. Let $l \in \{1,...,m\}$. We assume that the following holds.

$$(k=2<p^m,\ 2n-1<p^m) \text{ and } (k=2<p^l,\ n \leq p^l)$$

When $p=2$,

$$(k=2<2^m,\ n<2^m) \text{ and } (k=2<2^l,\ n \leq 2^l).$$

Let f be a surjective linear mapping ($f : GF(p^m) \to GF(p^l)$). It satisfies (2) and (3). For example, such a mapping is given by the mapping that extracts the last l digits from $x \in GF(p^m)$ in vector representation.

The encoding procedure is as follows. Shares are defined as $V_i := (W_i, U_i)$ $(1 \leq i \leq n)$. In particular, $W_i \in GF(p^m)$ and $U_i \in GF(p^l)$ are given by

$$\begin{bmatrix} W_1 & \cdots & W_n \end{bmatrix} = \begin{bmatrix} S_1 & S_2 \end{bmatrix} A,$$
$$\begin{bmatrix} U_1 & \cdots & U_n \end{bmatrix} = \begin{bmatrix} f(S_1 \cdot S_2) & R_1' \end{bmatrix} B.$$

Here, $A \in GF(p^m)^{2\times n}$ is a generator matrix of a linear strong $(2,2,n)$ ramp SSS for securing $S_1 \cdot S_2$, $B \in GF(p^l)^{2\times n}$ is a generator matrix of a linear $(2,n)$ SSS, and R_1' is a uniform random number over $GF(p^l)$.

The decoding procedure is as follows. Let $\hat{V}_{i_1} = (\hat{W}_{i_1}, \hat{U}_{i_1})$ and $\hat{V}_{i_2} = (\hat{W}_{i_2}, \hat{U}_{i_2})$ be the input of the decoder, let $\boldsymbol{a}_1, ..., \boldsymbol{a}_n$ be columns of A, and let $\boldsymbol{b}_1, ..., \boldsymbol{b}_n$ be columns of B. Then, define $C \in GF(p^m)^{2\times 2}$ and $D \in GF(p^l)^{2\times 2}$ as

$$C = (c_{ij}) := \begin{bmatrix} \boldsymbol{a}_{i_1} & \boldsymbol{a}_{i_2} \end{bmatrix}^{-1},$$
$$D = (d_{ij}) := \begin{bmatrix} \boldsymbol{b}_{i_1} & \boldsymbol{b}_{i_2} \end{bmatrix}^{-1}.$$

From the encoding procedure, the followings hold for correct shares (W_{i_1}, U_{i_1}) and (W_{i_2}, U_{i_2}).

$$\begin{bmatrix} S_1 & S_2 \end{bmatrix} = \begin{bmatrix} W_{i_1} & W_{i_2} \end{bmatrix} C$$
$$\begin{bmatrix} f(S_1 \cdot S_2) & R_1' \end{bmatrix} = \begin{bmatrix} U_{i_1} & U_{i_2} \end{bmatrix} D$$

The decoder checks whether

$$f((c_{11}\hat{W}_{i_1}+c_{21}\hat{W}_{i_2})\cdot(c_{12}\hat{W}_{i_1}+c_{22}\hat{W}_{i_2})) = \sum_{\ell=1}^{2} d_{\ell 1}\hat{U}_{i_\ell} \quad (7)$$

holds or not. If it holds, the decoder outputs $\hat{S}^2$ where

$$\hat{S}_j = c_{1j}\hat{W}_{i_1} + c_{2j}\hat{W}_{i_2},\ j=1,2.$$

If it is not satisfied, the decoder outputs $\perp$.

Theorem 2. Our proposed scheme is a strong $(2,2,n)$ ramp SSS with correlation level $(l)_p$, and the size of shares, the size of random number used in encoding, and successful cheating probabilities are as follows.

$$\log_p |\mathcal{V}_i| = m+l,\ i=1,...,n, \quad (8)$$
$$\log_p |\mathcal{R}| = l, \quad (9)$$
$$P_{imp^*(a)} = p^{-l},\ a=1, \quad (10)$$
$$P_{imp(a)} = p^{-l}(1-p^{-m}),\ a=1, \quad (11)$$
$$P_{sub(a)} \leq p^{-l},\ a=1. \quad (12)$$

Furthermore, for any correlation level $(l)_p$, (8) to (11) achieve the lower bounds of (k,L,n) ramp SSSs or the lower bound of strong (k,L,n) ramp SSSs, and (12) almost achieves the lower bound of strong (k,L,n) ramp SSSs.

Before proving Theorem 2, we show the following lemma.

Lemma 1. For any $\{i_1,i_2\} \subseteq \{1,\ldots,n\}$, 3-tuple $(W_{i_1}, W_{i_2}, U_{i_1})$ is uniformly distributed over $GF(p^m)^2 \times GF(p^l)$. In particular, these 3 random variables are mutually independent.

Lemma 1 can be proven in the same way as [8, Lemma 2] (see the full version).

Proof of Theorem 2. We briefly show it as follows.

Proof that our scheme is a strong $(2,2,n)$ ramp SSS with correlation level $(l)_p$. It can be proven from the fact that we use a strong $(2,2,n)$ ramp SSS to divide a secret into shares and a $(2,n)$ SSS to divide $f(S_1 \cdot S_2)$ into shares, and Lemma 1. □

Proofs of (8) and (9). These are clear from $V_i \in GF(p^m) \times GF(p^l)$ and $R_1' \in GF(p^l)$. Moreover, these achieve the lower bounds from Proposition 1. □

For showing the proofs of (10), (11), and (12), we assume that, in decoding, one share $V_{i_1} = (W_{i_1}, U_{i_1})$ is forged into $\bar{V}_{i_1} = (\bar{W}_{i_1}, \bar{U}_{i_1})$, and $V_{i_2} = (W_{i_2}, U_{i_2})$ is correct. Let

$$\Delta_j(W_{i_1}, \bar{W}_{i_1}) = \bar{S}_j - S_j\ (\bar{S}_j = c_{1j}\bar{W}_{i_1} + c_{2j}W_{i_2}),\ j=1,2.$$

Define

$$g(\bar{W}_{i_1}, W_{i_1}, W_{i_2}) := (c_{11}\bar{W}_{i_1} + c_{21}W_{i_2})\cdot(c_{12}\bar{W}_{i_1} + c_{22}W_{i_2}) - (c_{11}W_{i_1} + c_{21}W_{i_2})\cdot(c_{12}W_{i_1} + c_{22}W_{i_2}).$$

From (7),

$$f((c_{11}\bar{W}_{i_1}+c_{21}W_{i_2})\cdot(c_{12}\bar{W}_{i_1}+c_{22}W_{i_2})) = d_{11}\bar{U}_{i_1}+d_{21}U_{i_2} \quad (13)$$

needs to be satisfied, for attack to succeed. Then, we have

$$f(g(\bar{W}_{i_1}, W_{i_1}, W_{i_2})) = d_{11}(\bar{U}_{i_1} - U_{i_1}) \quad (14)$$

from (13), the property of f, and the fact that correct shares satisfy

$$f((c_{11}W_{i_1}+c_{21}W_{i_2})\cdot(c_{12}W_{i_1}+c_{22}W_{i_2})) = \sum_{\ell=1}^{2} d_{\ell 1}U_{i_\ell}.$$

Thus, the definition of the success of attack $\psi(\bar{V}_{i_1}, V_{i_2}) \neq \perp$ is given by (14), and $\psi(\bar{V}_{i_1}, V_{i_2}) \notin \{S^2, \perp\}$ is given by (14) and

$$\Delta_1(W_{i_1}, \bar{W}_{i_1}) \neq 0 \text{ and } \Delta_2(W_{i_1}, \bar{W}_{i_1}) \neq 0. \quad (15)$$

Proofs of (10) and (11). These are proven from the fact that $(\bar{W}_{i_1}, \bar{U}_{i_1})$, W_{i_1}, U_{i_1}, and W_{i_2} are mutually independent, and others (see the full version). In addition, these achieve the lower bounds from Proposition 1. □

Proof of (12). We consider the case that the value of a share to be forged is $V_{i_1} = v_{i_1} (= (w_{i_1}, u_{i_1}))$. Define

$$\begin{aligned}\bar{\mathcal{V}}'_{i_1} := \{(\bar{w}_{i_1}, \bar{u}_{i_1}) \in (GF(p^m) \times GF(p^l)) : \\ \Pr\{(\bar{W}_{i_1}, \bar{U}_{i_1}) = (\bar{w}_{i_1}, \bar{u}_{i_1}) | V_{i_1} = v_{i_1}\} > 0, \\ \Delta_1(w_{i_1}, \bar{w}_{i_1}) \neq 0,\ \Delta_2(w_{i_1}, \bar{w}_{i_1}) \neq 0\}.\end{aligned}$$

Now, to prove (12), we show the following lemma.

Lemma 2. Fix $(\bar{w}_{i_1}, \bar{u}_{i_1}) \in \bar{\mathcal{V}}'_{i_1}$ arbitrarily. Then, there are p^{m-l} values of $w_{i_2} \in GF(p^m)$ which satisfy $f(g(\bar{w}_{i_1}, w_{i_1}, w_{i_2})) = d_{11}(\bar{u}_{i_1} - u_{i_1})$.

Proof of Lemma 2. From the property of f, $f(\alpha) = d_{11}(\bar{u}_{i_1} - u_{i_1})$ is satisfied for just p^{m-l} values of $\alpha \in GF(p^m)$. In addition, when $(\bar{w}_{i_1}, \bar{u}_{i_1}) \in \bar{\mathcal{V}}'_{i_1}$,

$$\begin{aligned}g(\bar{w}_{i_1}, w_{i_1}, w_{i_2}) = (c_{21}\Delta_2(w_{i_1}, \bar{w}_{i_1}) + c_{22}\Delta_1(w_{i_1}, \bar{w}_{i_1}))w_{i_2} \\ + (c_{11}\Delta_2(w_{i_1}, \bar{w}_{i_1}) + c_{12}\Delta_1(w_{i_1}, \bar{w}_{i_1}))w_{i_1} \\ + \Delta_1(w_{i_1}, \bar{w}_{i_1}) \cdot \Delta_2(w_{i_1}, \bar{w}_{i_1})\end{aligned}$$

is obtained. The coefficient of w_{i_2} is represented as $(c_{11}c_{22} + c_{12}c_{21})\delta_1$ (here, δ_1 is defined as $\delta_1 = \bar{w}_{i_1} - w_{i_1}$). From $(\bar{w}_{i_1}, \bar{u}_{i_1}) \in \bar{\mathcal{V}}'_{i_1}$ and using a generator matrix for securing $S_1 \cdot S_2$, $(c_{11}c_{22} + c_{12}c_{21})\delta_1 \neq 0$ holds. Since the coefficient of w_{i_2} is non-zero, $g(\bar{w}_{i_1}, w_{i_1}, w_{i_2})$ is a linear polynomial in w_{i_2}. Hence, there is one value of w_{i_2} which satisfies $g(\bar{w}_{i_1}, w_{i_1}, w_{i_2}) = \alpha$ for each of α. Thus, there are p^{m-l} values of w_{i_2} satisfying $f(g(\bar{w}_{i_1}, w_{i_1}, w_{i_2})) = d_{11}(\bar{u}_{i_1} - u_{i_1})$. □

From the above, the success probability of substitution attack is as follows.

$$\begin{aligned}&\Pr\{\psi(\bar{V}_{i_1}, V_{i_2}) \notin \{S^L, \perp\} | V_{i_1} = v_{i_1}\} \\ &= \Pr\{(14), (15) | V_{i_1} = v_{i_1}\} \\ &= \sum_{\bar{v}_{i_1} \in \bar{\mathcal{V}}'_{i_1}} \Pr\{(14), \bar{V}_{i_1} = \bar{v}_{i_1} | V_{i_1} = v_{i_1}\} \\ &= \sum_{\bar{v}_{i_1} \in \bar{\mathcal{V}}'_{i_1}} \Pr\{\bar{V}_{i_1} = \bar{v}_{i_1} | V_{i_1} = v_{i_1}\} \\ &\quad \cdot \Pr\{(14) | V_{i_1} = v_{i_1}, \bar{V}_{i_1} = \bar{v}_{i_1}\} \\ &\overset{(a)}{=} p^{-l} \sum_{\bar{v}_{i_1} \in \bar{\mathcal{V}}'_{i_1}} \Pr\{\bar{V}_{i_1} = \bar{v}_{i_1} | V_{i_1} = v_{i_1}\} \\ &\leq p^{-l}\end{aligned} \tag{16}$$

Here, (a) holds because

$$\begin{aligned}&\Pr\{(14) | V_{i_1} = v_{i_1}, \bar{V}_{i_1} = \bar{v}_{i_1}\} \\ &= \Pr\{f(g(\bar{w}_{i_1}, w_{i_1}, W_{i_2})) = d_{11}(\bar{u}_{i_1} - u_{i_1}) | V_{i_1} = v_{i_1}\} \\ &= p^{-m} \cdot p^{m-l} = p^{-l}.\end{aligned} \tag{17}$$

(17) holds from the Markov chain $\bar{V}_{i_1} \to V_{i_1} \to W_{i_2}$, $\Pr\{W_{i_2} = w_{i_2} | V_{i_1} = v_{i_1}\} = p^{-m}$ for any $w_{i_2} \in GF(p^m)$ (from Lemma 1), and Lemma 2.

(16) holds for any $v_{i_1} = (w_{i_1}, v_{i_1})$. Compared to (1), the lower bound has not been achieved, to be exact ($P_{sub(a=1)}$ is within $\frac{1}{1-p^{-m}}$ times the lower bound with $|\mathcal{V}_i| = p^{m+l}$). However, (1) is not tight. Thus, it (almost) achieves the lower bound. □

From the above, Theorem 2 is proven. □

4. Difficulty of Applying Our Strategy to More Generalized Parameters

Our verification function cannot apply to more generalized parameters, in particular, $(k \geq 3, 2, n)$. In the parameters, an attacker can succeed in substitution attacks with probability 1 and no condition of a generator matrix prevents it. In addition, applying our strategy to more generalized parameters (k, L, n) is probably difficult. This is because, conditions of a generator matrix with which a verification function detects substitution attacks are probably complicated in $k, L \geq 3$. We discuss them in detail in the full version of this paper.

5. Conclusion

In this research, we have proposed an (almost) optimal cheating-detectable strong $(2, 2, n)$ ramp SSS using $S_1 \cdot S_2$. The proposed scheme achieves the lower bounds on the size of shares, the size of random number used in encoding, and the success probability of impersonation attack, and almost achieves the lower bound on the success probability of substitution attack for any correlation level $(l)_p$. Our scheme differs from existing research in that it uses a limited type, and is the first to almost achieve the lower bound on the success probability of substitution attack.

The future task is to propose a cheating-detectable strong (k, L, n) ramp SSS (not only $(2, 2, n)$) which achieves the lower bounds on them.

REFERENCES

[1] G. R. Blakley, "Safeguarding cryptographic keys," Proc. National Computer Conference, vol.48, pp.313-317, 1979.

[2] Adi Shamir, "How to share a secret," Commun. ACM, vol.22, no.11, pp.612-613, 1979.

[3] Hirosuke Yamamoto, "On Secret Sharing System Using (k, L, n) Threshold Scheme," IEICE Transactions, vol.J68-A no.9, pp.945-952, 1985 (in Japanese).

[4] Sergio Cabello, Carles Padró, and Germán Sáez, "Secret Sharing Schemes with Detection of Cheaters for a General Access Structure," Designs, Codes and Cryptography, vol.25, pp.175-188, 2002.

[5] Wakaha Ogata, Kaoru Kurosawa, and Douglas R. Stinson, "Optimum secret sharing scheme secure against cheating," SIAM J. Discrete Math., vol.20, no.1, pp.79-95, 2006.

[6] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs, "Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors", EUROCRYPT 2008, pp.471-488, 2008.

[7] Wataru Nakamura, and Hirosuke Yamamoto, "A Ramp Threshold Secret Sharing Scheme against Substitution Attacks," 2016 Symposium on Cryptography and Information Security, 3A1-1, 2016 (in Japanese).

[8] Wataru Nakamura, Hirosuke Yamamoto, and Terence Chan, "A Cheating-Detectable (k, L, n) Ramp Secret Sharing Scheme," IEICE Transactions, vol.E100-A, no.12, pp.2709-2719, 2017.

[9] Tomoki Agematsu and Satoshi Obana, "Almost optimal cheating-detectable $(2, 2, n)$ ramp secret sharing scheme," 2019 Seventh International Symposium on Computing and Networking (CANDAR), pp. 1-9, 2019.