A New (k, n)-Threshold Secret Sharing Scheme and Its Extension[⋆]

Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka

KDDI R&D Laboratories, Inc.,
2-1-15 Ohara, Fujimino-Shi, Saitama 356-8502, Japan.
kurihara@kddilabs.jp
http://www.kddilabs.jp/

Abstract. In Shamir's (k, n)-threshold secret sharing scheme (threshold scheme), a heavy computational cost is required to make n shares and recover the secret. As a solution to this problem, several fast threshold schemes have been proposed. This paper proposes a new (k, n)-threshold scheme. For the purpose to realize high performance, the proposed scheme uses just EXCLUSIVE-OR(XOR) operations to make shares and recover the secret. We prove that the proposed scheme is a *perfect* secret sharing scheme, every combination of k or more participants can recover the secret, but every group of less than k participants cannot obtain any information about the secret. Moreover, we show that the proposed scheme is an *ideal* secret sharing scheme similar to Shamir's scheme, which is a *perfect* scheme such that every bit-size of shares equals that of the secret. We also evaluate the efficiency of the scheme, and show that our scheme realizes operations that are much faster than Shamir's. Furthermore, from the aspect of both computational cost and storage usage, we also introduce how to extend the proposed scheme to a new (k, L, n)-threshold *ramp* scheme similar to the existing *ramp* scheme based on Shamir's scheme.

Key words: secret sharing scheme, threshold scheme, threshold ramp scheme, exclusive-or, entropy, random number, ideal secret sharing scheme

1 Introduction

A secret sharing scheme is an important tool for distributed file systems protected against data leakage and destruction, secure key management systems, etc. The basic idea of secret sharing introduced by Shamir and Blakley independently[1, 2] is that a dealer distributes a piece of information (called a share) about the secret to each participant such that qualified subsets of participants can recover the secret but unqualified subsets of participants cannot obtain any information about the secret. Shamir's threshold scheme is based on polynomial interpolation ('Lagrange interpolation') to allow any k out of n participants to recover the secret.

[⋆] An extended abstract of this paper appeared in ISC'08.

However, Shamir's scheme has two problems: large storage is required to retain all the shares, and heavy computational cost is needed to make shares and recover the secret due to processing a $(k-1)$-degree polynomial.

In order to reduce each bit-size of shares in Shamir's scheme, *ramp* secret sharing schemes have been proposed [3–7] that involve a trade-off between security and storage usage. In *ramp* schemes, we can consider *intermediate* sets, which are neither qualified nor forbidden sets to recover the secret, and hence, partially leak information on the secret. For instance, in the (k, L, n)-threshold *ramp* scheme[3, 4], we can recover the secret from arbitrary k or more shares, but no information about the secret can be obtained from any $k - L$ or less shares. Furthermore, we can realize that every bit-size of shares is $1/L$ of the bit-size of the secret. However, an arbitrary set of $k-l$ shares is an *intermediate* set which leaks information about the secret with equivocation $(l/L)H(S)$ for $l = 1, 2, \ldots, L$, where S denotes the random variable induced by the secret s.

On the other hand, as a solution to the heavy computational cost problem associated with Shamir's scheme with no leak of information about the secret from $k-1$ or less shares, Ishizu et al. proposed a fast $(2, 3)$-threshold scheme[8]. By generalizing Ishizu et al.'s scheme for the number of participants, Fujii et al. introduced a fast $(2, n)$-threshold scheme[9, 10]. These schemes enable fast computation to make shares and recover the secret from two or more shares by using just EXCLUSIVE-OR(XOR) operations. In these schemes, no information about the secret can be obtained from one share, but the secret can be recovered from each pair of shares. Furthermore, every bit-size of shares equals the bit-size of the secret as with Shamir's scheme. Especially, in Fujii et al.'s scheme, shares are constructed by concatenating XORed terms of a divided piece of the secret and a random number with the properties of prime numbers. These XORed terms are circulated in a specific pattern and do not overlap with each other. Kurihara et al. proposed a fast $(3, n)$-threshold scheme using XOR operations[11] as an extension of Fujii et al.'s scheme by constructing shares with the secret and two sets of random numbers, which are concatenated XORed terms of a divided piece of the secret and two random numbers. This $(3, n)$-threshold scheme is an *ideal* scheme as with Shamir's and Fujii et al.'s. Since no method has ever been investigated to extend the circulation property of this $(3, n)$-threshold scheme, an extension of this $(3, n)$-threshold scheme has not been proposed before.

Shiina et al. proposed another fast (k, n)-threshold scheme using XOR or additive operations[12]. This scheme can be applied to a cipher or signature which uses a homomorphism, and leaks no information about the secret from less than k shares. However, every bit-size of shares is $({}_nC_k - {}_{n-1}C_k) = O(n^{k-1})$ times as large as the bit-size of the secret. To address this efficiency problem, Kunii et al. introduced an alternative method[13] to construct shares in Shiina et al.'s scheme. However, the bit-size of shares is $\log_2 n$ or more times larger than the bit-size of the secret.

Thus, how to construct a fast (k, n)-threshold scheme using XOR operations such that every bit-size of shares equals the bit-size of the secret, where $k \geq 4$ and arbitrary n, remained an open question.

Our Contributions. In this paper, we present a new (k, n)-threshold scheme which realizes fast computation to make shares and recover the secret by using just XOR operations. Our contribution can be summarized as follows:

- We realize a new (k, n)-threshold scheme by constructing shares with the secret and $k - 1$ sets of random numbers, which are concatenated XORed terms of a divided piece of the secret and $k - 1$ random numbers. These XORed terms are circulated in a specific pattern with k dimensions, and do not overlap with each other because the properties of prime numbers are used.
- We show that the proposed scheme is a *perfect* secret sharing scheme, every combination of k or more participants can recover the secret, but every group of less than k participants cannot obtain any information about the secret. We also show that the proposed scheme is an *ideal* secret sharing scheme similar to Shamir's scheme, which is a *perfect* scheme such that every bit-size of shares equals that of the secret.
- By an implementation on a PC, we show that the proposed scheme is able to make n shares from the secret and recover the secret from k shares more quickly than Shamir's scheme if n is not extremely large. Under our implementation, our scheme performs the operations 900-fold faster than Shamir's for $(k, n) = (3, 11)$.
- We introduce how to extend our (k, n)-threshold scheme to a new (k, L, n)-threshold *ramp* scheme which realizes not only fast computation but also reduction of storage usage to retain n shares.

Organization. The rest of this paper is organized as follows: In Section 2, we give several notations and definitions, and provide a definition of the secret sharing scheme. In Section 3.1 of Section 3, we propose a new (k, n)-threshold scheme using just XOR operations. Moreover, in Section 3.3, we prove that our (k, n)-threshold scheme is an *ideal* secret sharing scheme as with Shamir's, and the efficiency of the proposed scheme is discussed in Section 4. In Section 5, we introduce how to extend our (k, n)-threshold scheme to a new (k, L, n)-threshold *ramp* scheme. Finally, we present our conclusions in Section 6.

2 Preliminaries

2.1 Notations and Definitions

Throughout this paper, we use the following notations and definitions:

- $\oplus$ denotes a bit-wise EXCLUSIVE-OR(XOR) operation.
- $\|$ denotes a concatenation of binary sequences.
- $n \in \mathbb{N}$ denotes the number of participants.
- n_p is a prime number such that $n_p \geq n$.
- Values of indexes of random numbers, divided pieces of the secret, pieces of shares, their XORed terms, and their random variables are elements of $GF(n_p)$. Hence, $X_{c(a \pm b)}$ denotes $X_{c(a \pm b) \bmod n_p}$.

- $H(X)$ denotes Shannon's entropy of a random variable X.
- $|\mathcal{X}|$ denotes the number of elements of a finite set $\mathcal{X}$.
- $2^{\mathcal{X}}$ denotes the family of all subsets of $\mathcal{X}$.

2.2 Secret Sharing Scheme

Let $\mathcal{P} = \{P_i \mid 0 \leq i \leq n-1,\ i \in \mathbb{N}_0\}$ be a set of n participants. Let $\mathcal{D}(\notin \mathcal{P})$ denote a dealer who selects a secret $s \in \mathcal{S}$ and gives a share $w_i \in \mathcal{W}_i$ to every participant $P_i \in \mathcal{P}$, where $\mathcal{S}$ denotes the set of secrets, and $\mathcal{W}_i$ denotes the set of possible shares that P_i might receive.

The access structure $\mathit{\Gamma}(\subset 2^{\mathcal{P}})$ is a family of subsets of $\mathcal{P}$ which contains the sets of participants qualified to recover the secret. Especially, $\mathit{\Gamma}$ of a (k,n)-threshold scheme is defined by $\mathit{\Gamma} = \{A \in 2^{\mathcal{P}} \mid |A| \geq k\}$.

Let S and W_i be the random variables induced by s and w_i, respectively. A secret sharing scheme is *perfect* if

$$H(S|\mathcal{V}_A) = \begin{cases} 0 & (A \in \mathit{\Gamma}) \\ H(S) & (A \notin \mathit{\Gamma}) \end{cases}, \tag{1}$$

where $A \subset \mathcal{P}$ denotes a subset, and $\mathcal{V}_A = \{W_i \mid P_i \in A\}$ denotes the set of random variables of shares that are given to every participant $P_i \in A$. For any *perfect* secret sharing scheme, the inequation $H(S) \leq H(W_i)$ is satisfied[14, 15].

Let $p(s)$ and $p(w_i)$ be the probability mass functions of S and W_i defined as $p(s) = \Pr\{S = s\}$ and $p(w_i) = \Pr\{W_i = w_i\}$, respectively. In general, the efficiency of a secret sharing scheme is measured by the information rate ρ [16] defined by

$$\rho = \frac{H(S)}{\max_{P_i \in \mathcal{P}} H(W_i)}.$$

The maximum possible value of ρ equals one for *perfect* secret sharing schemes. When the probability distributions on $\mathcal{S}$ and $\mathcal{W}_i$ are uniform, i.e. $p(s) = 1/|\mathcal{S}|$ and $p(w_i) = 1/|\mathcal{W}_i|$, the information rate is

$$\rho = \frac{\log_2 |\mathcal{S}|}{\max_{P_i \in \mathcal{P}} \log_2 |\mathcal{W}_i|},$$

that is, the ratio between the length (bit-size) of the secret and the maximum length of the shares given to participants. A secret sharing scheme is said to be *ideal* if it is *perfect* and $\rho = 1$ [16–18]. Shamir's scheme[1] is recognized as being a typical *ideal* secret sharing scheme.

3 A (k, n)-Threshold Scheme

In this section, we describe the proposed (k,n)-threshold scheme. This scheme enables to make n shares (distribution) and recover the secret from k or more

Table 1. Distribution Algorithm of Proposed (k, n)-Threshold Scheme

```
INPUT : $s \in \{0,1\}^{d(n_p-1)}$
OUTPUT : $(w_0, \ldots, w_{n-1})$

1: $s_0 \leftarrow 0^d$, $s_1 \| \cdots \| s_{n_p-1} \leftarrow s$
2: for $i \leftarrow 0$ to $k-2$ do
3:     for $j \leftarrow 0$ to $n_p - 1$ do
4:         $r^i_j \leftarrow GEN(\{0,1\}^d)$
5:     end for
6: end for (discard $r^0_{n_p-1}$)
7: for $i \leftarrow 0$ to $n-1$ do
8:     for $j \leftarrow 0$ to $n_p - 2$ do
9:         $w_{(i,j)} \leftarrow \left(\bigoplus_{h=0}^{k-2} r^h_{h \cdot i+j}\right) \oplus s_{j-i}$
10:    end for
11:    $w_i \leftarrow w_{(i,0)} \| \cdots \| w_{(i,n_p-2)}$
12: end for
13: return $(w_0, \ldots, w_{n-1})$
```

Table 2. Recovery Algorithm of Proposed (k, n)-Threshold Scheme

```
INPUT : $(w_{t_0}, w_{t_1}, \ldots, w_{t_{k-1}})$
OUTPUT : $s$

for $i \leftarrow 0$ to $k-1$ do
    $w_{(t_i,0)} \| \cdots \| w_{(t_i,n_p-2)} \leftarrow w_{t_i}$
end for
$\mathbf{w} \leftarrow (w_{(t_0,0)}, \ldots, w_{(t_0,n_p-2)}, \ldots,$
        $w_{(t_{k-1},0)}, \ldots, w_{(t_{k-1},n_p-2)})^{\mathrm{T}}$
$\mathbf{M} \leftarrow MAT(t_0, \ldots, t_{k-1})$
$(s_1, \ldots, s_{n_p-1})^{\mathrm{T}} \leftarrow \mathbf{M} \cdot \mathbf{w}$
$s \leftarrow s_1 \| \cdots \| s_{n_p-1}$
return $s$
```

shares (recovery) using just XOR operations, for arbitrary threshold k and the number of participants n. We realize this scheme by extending the circulation property of Kurihara et al.'s $(3, n)$-threshold scheme[11]. Moreover, we show that our scheme is an *ideal* scheme as with Shamir's.

3.1 Our Scheme

In this scheme, the secret $s \in \{0,1\}^{d(n_p-1)}$ needs to be divided equally into $n_p - 1$ blocks $s_1, s_2, \ldots s_{n_p-1} \in \{0,1\}^d$, where n_p is a prime number such that $n_p \geq n$, and $d > 0$ denotes the bit-size of every divided piece of the secret. Also, $\mathcal{D}$ uses n shares, $w_0, \cdots, w_{n-1}$, of a (k, n_p)-threshold scheme to construct a (k, n)-threshold scheme if the desired number of participants n is a composite number.

Table 1 and Table 4 denote the distribution algorithm and the structure of shares in our (k, n)-threshold scheme, respectively. To make shares, our (k, n)-threshold scheme requires 3 steps, where line 1, lines 2-6 and lines 6-13 in Table 1 denote the first, second and third step, respectively: First, $\mathcal{D}$ divides the secret $s \in \{0,1\}^{d(n_p-1)}$ into $n_p - 1$ pieces of d-bit sequence $s_1, \ldots, s_{n_p-1} \in \{0,1\}^d$ equally at line 1, where s_0 denotes a d-bit zero sequence, i.e. $s_0 = 0^d$ and $s_0 \oplus a = a$. We call this d-bit zero sequence a 'singular point' of divided pieces

Table 3. Algorithm of the Function $MAT()$

```
INPUT : $t_0, t_1, \ldots, t_{k-1}$
OUTPUT : $\mathbf{M}$

1: for $i \leftarrow 0$ to $k-1$ do
2:     for $j \leftarrow 0$ to $n_p - 2$ do
3:         $\mathbf{v}_{(t_i,j)} \leftarrow VEC(t_i, j) = \left[\mathbf{i}_j^{n_p-1} \ \mathbf{i}_{t_i+j}^{n_p} \ \mathbf{i}_{2t_i+j}^{n_p} \ \cdots \ \mathbf{i}_{(k-2)t_i+j}^{n_p} \ \mathbf{i}_{j-t_i-1}^{n_p-1}\right]$
4:     end for
5: end for
6: $\mathbf{G} \leftarrow (\mathbf{v}_{(t_0,0)}, \ldots, \mathbf{v}_{(t_{k-1},n_p-2)})^{\mathrm{T}}$
7: $\begin{bmatrix} \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{J}_1 \\ \emptyset & \mathbf{G}_0 & \mathbf{J}_0 \end{bmatrix} \leftarrow FG\left(\left[\mathbf{G} \quad \mathbf{I}_{k(n_p-1)}\right]\right) = \left[\bar{\mathbf{G}} \quad \mathbf{J}\right]$
8: $[\mathbf{I}_{n_p-1} \quad \mathbf{M}] \leftarrow BG([\mathbf{G}_0 \quad \mathbf{J}_0])$
9: return $\mathbf{M}$
```

Table 4. Structure of Shares of Proposed (k,n)-Threshold Scheme

	$j=0$	$j=1$	$\cdots$	$j=n_p-2$
$w_{(0,j)}$	$\left\{\bigoplus_{h=0}^{k-2} r_0^h\right\} \oplus s_0$	$\left\{\bigoplus_{h=0}^{k-2} r_1^h\right\} \oplus s_1$	$\cdots$	$\left\{\bigoplus_{h=0}^{k-2} r_{-2}^h\right\} \oplus s_{-2}$
$w_{(1,j)}$	$\left\{\bigoplus_{h=0}^{k-2} r_h^h\right\} \oplus s_{-1}$	$\left\{\bigoplus_{h=0}^{k-2} r_{h+1}^h\right\} \oplus s_0$	$\cdots$	$\left\{\bigoplus_{h=0}^{k-2} r_{h-2}^h\right\} \oplus s_{-3}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$w_{(n-1,j)}$	$\left\{\bigoplus_{h=0}^{k-2} r_{h\cdot(n-1)}^h\right\} \oplus s_{-n+1}$	$\left\{\bigoplus_{h=0}^{k-2} r_{h\cdot(n-1)+1}^h\right\} \oplus s_{-n+2}$	$\cdots$	$\left\{\bigoplus_{h=0}^{k-2} r_{h\cdot(n-1)-2}^h\right\} \oplus s_{-n-1}$

of the secret.[1] Next, at lines 2–6, $(k-1)n_p - 1$ pieces of d-bit random number $r_0^0, \ldots, r_{n_p-2}^0, r_0^1, \ldots, r_{n_p-1}^1, \ldots, r_0^{k-2}, \ldots, r_{n_p-1}^{k-2}$ are chosen from $\{0,1\}^d$ independently from each other with uniform probability $1/2^d$, where $GEN(\mathcal{X})$ denotes a function to generate an $(\log_2 |\mathcal{X}|)$-bit random number from a finite set $\mathcal{X}$. At lines 7–12, $\mathcal{D}$ makes pieces of shares by means of the following equation:

$$w_{(i,j)} = \left\{\bigoplus_{h=0}^{k-2} r_{h\cdot i+j}^h\right\} \oplus s_{j-i}, \tag{2}$$

[1] It is not necessary for the singular point to be s_0, i.e. we can set an arbitrary singular point s_m $(0 \leq m \leq n_p - 1)$ and the others are $n_p - 1$ divided pieces of the secret. For the sake of simplicity, we suppose that the singular point is s_0 in this paper.

where $0 \leq i \leq n-1$, $0 \leq j \leq n_p - 2$. Finally, $\mathcal{D}$ concatenates these pieces and constructs shares $w_i = w_{(i,0)} \parallel \cdots \parallel w_{(i,n_p-2)}$, and sends shares to each participant through a secure channel. If $n < n_p$, lines 7–12 does not work for $0 \leq i \leq n_p - 1$ but it does for $0 \leq i \leq n-1$, and hence $\mathcal{D}$ does not generate $n_p - n$ shares $w_n, \cdots, w_{n_p-1}$. Thus, it is possible to add new participants $P_n, \cdots, P_{n_p-1}$ after distribution by generating $w_n, \cdots, w_{n_p-1}$ anew as necessary. However, to generate new shares, k existing shares should be gathered, and all random numbers and the secret should be stored.

Eq.(2) shows that these pieces of shares are circulated in a specific pattern with k dimensions by the indexes of a divided piece of the secret k random numbers, and do not overlap with each other because the properties of prime numbers are used.

Table 2 denotes the recovery algorithm in the scheme. First, each share is divided into d-bit pieces at lines 1–3. Next, at line 4, $k(n_p - 1)$-dimensional vector $\mathbf{w}$ is generated, which is a vector of divided pieces of shares. At line 5, $k(n_p-1) \times k(n_p-1)$ binary matrix $\mathbf{M}$ is obtained by the function $MAT()$. All divided pieces of the secret, $s_1, \ldots, s_{n_p-1}$, are recovered by calculating $\mathbf{M} \cdot \mathbf{w}$ at line 6. Finally, the secret s is recovered by concatenating $s_1, \ldots, s_{n_p-1}$ at line 7.

Table 3 denotes the algorithm of the function $MAT()$ which makes the matrix $\mathbf{M}$. First, $(kn_p - 2)$-dimensional binary vector $\mathbf{v}_{(t_i,j)}$ is obtained from indexes t_i and j at lines 1–5. $VEC()$ denotes the function to make $\mathbf{v}_{(t_i,j)}$, where $\mathbf{i}^x_y$ denotes a x-dimentional binary row vector such that the only y-th element equals one $(0 \leq y \leq x-1)$ and the others are zero. $\mathbf{v}_{(t_i,j)}$ is defined as the generator vector of $w_{(t_i,j)}$, i.e. $w_{(t_i,j)} = \mathbf{v}_{(t_i,j)} \cdot \mathbf{e}$, where $\mathbf{e}$ is defined by

$$\mathbf{e} = (r^0_0, \ldots, r^0_{n_p-2}, r^1_0, \ldots, r^1_{n_p-1}, \ldots, r^{k-2}_0, \ldots, r^{k-2}_{n_p-1}, s_1, \ldots, s_{n_p-1})^{\mathrm{T}},$$

where s_0 is omitted for the simple reason that $s_0 = 0^d$. For instance, $\mathbf{v}_{(0,1)} =$ (0100 01000 01000 1000) if $k = 4$ and $n_p = 5$. At line 6, the $k(n_p-1) \times (kn_p - 2)$ binary matrix $\mathbf{G}$ is generated by $\mathbf{v}_{(t_0,0)}, \ldots, \mathbf{v}_{(t_{k-1},n_p-2)}$ as follows:

$$\mathbf{G} = \left(\mathbf{v}_{(t_0,0)}, \ldots, \mathbf{v}_{(t_0,n_p-2)}, \ldots, \mathbf{v}_{(t_{k-1},0)}, \ldots, \mathbf{v}_{(t_{k-1},n_p-2)}\right)^{\mathrm{T}},$$

which is the generator matrix such that $\mathbf{w} = \mathbf{G} \cdot \mathbf{r}$. At line 7, the matrix $[\mathbf{G} \quad \mathbf{I}_{k(n_p-1)}]$ is generated by column-wise concatenation, and transformed into a row echelon form $\left[\bar{\mathbf{G}} \quad \mathbf{J}\right] = FG\left(\left[\mathbf{G} \quad \mathbf{I}_{k(n_p-1)}\right]\right)$ by performing the forward elimination step of Gaussian elimination with the elementary row operations on GF(2), where $FG()$ and $\mathbf{I}_{k(n_p-1)}$ denote a forward elimination function and the $k(n_p-1) \times k(n_p-1)$ identity matrix, respectively. Furthermore, $\bar{\mathbf{G}}$ and $\mathbf{J}$ correspond to the transformed matrices from $\mathbf{G}$ and $\mathbf{I}_{k(n_p-1)}$, respectively. And, $\left[\bar{\mathbf{G}} \quad \mathbf{J}\right]$ is divided into block matrices denoted as follows:

$$\left[\bar{\mathbf{G}} \quad \mathbf{J}\right] = \begin{bmatrix} \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{J}_1 \\ \text{Ø} & \mathbf{G}_0 & \mathbf{J}_0 \end{bmatrix},$$

where $\mathbf{G}_0$, $\mathbf{G}_1$ and $\mathbf{G}_2$ are an $(n_p-1) \times (n_p-1)$ block matrix, $(k-1)(n_p-1) \times (n_p-1)$ block matrix and $(k-1)(n_p-1) \times (kn_p - n_p - 1)$ block matrix, respectively. $\mathbf{J}_0$ and $\mathbf{J}_1$ are an $(n_p-1)\times k(n_p-1)$ block matrix and a $(k-1)(n_p-1)\times k(n_p-1)$ block matrix, respectively. $\mathbf{\emptyset}$ denotes a null matrix. Then, the backward substitution part of Gaussian elimination is executed on $[\mathbf{G}_0 \quad \mathbf{J}_0]$, and we obtain the matrix $\left[\mathbf{I}_{n_p-1} \quad \mathbf{M}\right] = BG([\mathbf{G}_0 \quad \mathbf{J}_0])$, where $BG()$ and $\mathbf{M}$ denote the function of backward substitution and a transformed matrix from $\mathbf{J}_0$, respectively. Finally, $MAT()$ outputs $\mathbf{M}$ as a matrix to recover $s_1,\ldots,s_{n_p-1}$ from divided pieces of shares.

Our (k,n)-threshold scheme proposed in this paper is a direct extension of Kurihara et al.'s $(3,n)$-threshold scheme[11] and Fujii et al.'s $(2,n)$-threshold scheme[9] in terms of the structure of shares. Accordingly, the distribution and recovery algorithms of our (k,n)-threshold scheme for $k=3$ and $k=2$ can be utilized as Kurihara et al.'s $(3,n)$-threshold scheme and Fujii et al.'s $(2,n)$-threshold scheme, respectively.

3.2 Example

We present the recovery procedure of our (k,n)-threshold scheme for $k=4$ and $n=n_p=5$ as an example. Table 5 shows the structure of shares of the $(4,5)$-threshold scheme.

Suppose that the participants P_0,P_1,P_2 and P_4 agree to recover the secret s with their shares w_0,w_1,w_2 and w_4. At lines 1-3 of Table 2, w_0,w_1,w_2 and w_4 are equally divided into d-bit pieces, $w_{(0,i)},w_{(1,i)},w_{(2,i)}$ and $w_{(4,i)}$ $(0\le i\le 3)$. Next, we obtain a 16-dimensional vector of divided pieces of shares $\mathbf{w}$ at line 4 of Table 2, which is denoted as follows:

$$\begin{aligned}\mathbf{w} = (&w_{(0,0)},w_{(0,1)},w_{(0,2)},w_{(0,3)},w_{(1,0)},w_{(1,1)},w_{(1,2)},w_{(1,3)},\\ &w_{(2,0)},w_{(2,1)},w_{(2,2)},w_{(2,3)},w_{(4,0)},w_{(4,1)},w_{(4,2)},w_{(4,3)})^{\mathrm{T}}.\end{aligned}$$

At line 5 of Table 2, we execute the function $MAT(0,1,2,4)$ and obtain 16×16 binary matrix $\mathbf{M}$. In the function $MAT()$, first, we obtain the generator matrix $\mathbf{G}$ from indexes of shares at lines 1-6 of Table 3, which is denoted as follows:

$$\mathbf{G} = \begin{pmatrix} 1000\ 10000\ 10000\ 0000 \\ 0100\ 01000\ 01000\ 1000 \\ 0010\ 00100\ 00100\ 0100 \\ 0001\ 00010\ 00010\ 0010 \\[4pt] 1000\ 01000\ 00100\ 0001 \\ 0100\ 00100\ 00010\ 0000 \\ 0010\ 00010\ 00001\ 1000 \\ 0001\ 00001\ 10000\ 0100 \\[4pt] 1000\ 00100\ 00001\ 0010 \\ 0100\ 00010\ 10000\ 0001 \\ 0010\ 00001\ 01000\ 0000 \\ 0001\ 10000\ 00100\ 1000 \\[4pt] 1000\ 00001\ 00010\ 1000 \\ 0100\ 10000\ 00001\ 0100 \\ 0010\ 01000\ 10000\ 0010 \\ 0001\ 00100\ 01000\ 0001 \end{pmatrix}.$$

At line 7 of Table 3, we execute forward elimination step of Gaussian elimination on $[\mathbf{G} \quad \mathbf{I}_{16}]$ and obtain a row echelon matrix $[\bar{\mathbf{G}} \quad \mathbf{J}]$ denoted as follows:

$$
\begin{aligned}
[\bar{\mathbf{G}} \quad \mathbf{J}] &= FG([\mathbf{G} \quad \mathbf{I}_{16}]) \\
&= \left(\begin{array}{cccc|cccc}
1000 & 10000 & 10000 & 0000 & 1000 & 0000 & 0000 & 0000 \\
0100 & 01000 & 01000 & 1000 & 0100 & 0000 & 0000 & 0000 \\
0010 & 00100 & 00100 & 0100 & 0010 & 0000 & 0000 & 0000 \\
0001 & 00010 & 00010 & 0010 & 0001 & 0000 & 0000 & 0000 \\[4pt]
0000 & 11000 & 10100 & 0001 & 1000 & 1000 & 0000 & 0000 \\
0000 & 01100 & 01010 & 1000 & 0100 & 0100 & 0000 & 0000 \\
0000 & 00110 & 00101 & 1100 & 0010 & 0010 & 0000 & 0000 \\
0000 & 00011 & 10010 & 0110 & 0001 & 0001 & 0000 & 0000 \\[4pt]
0000 & 00000 & 10111 & 1101 & 0010 & 0110 & 0100 & 0000 \\
0000 & 00000 & 01111 & 1011 & 0100 & 1100 & 1000 & 0000 \\
0000 & 00000 & 00101 & 1001 & 1001 & 0100 & 1101 & 0000 \\
0000 & 00000 & 00011 & 1000 & 0111 & 1001 & 1110 & 0000 \\
\hline
0000 & 00000 & 00000 & 1001 & 1110 & 0011 & 1111 & 0010 \\
0000 & 00000 & 00000 & 0101 & 0110 & 1100 & 0010 & 1000 \\
0000 & 00000 & 00000 & 0010 & 0011 & 0110 & 0001 & 0100 \\
0000 & 00000 & 00000 & 0001 & 0001 & 0010 & 1010 & 1001
\end{array}\right).
\end{aligned}
$$

And also, $[\bar{\mathbf{G}} \quad \mathbf{J}]$ is divided into six block matrices, $\boldsymbol{\emptyset}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2, \mathbf{J}_0$ and $\mathbf{J}_1$, at line 7 of Table 3. Then, $[\mathbf{G}_0 \quad \mathbf{J}_0]$ is denoted as follows:

$$
[\mathbf{G}_0 \quad \mathbf{J}_0] = \left(\begin{array}{c|cccc}
1001 & 1110 & 0011 & 1111 & 0010 \\
0101 & 0110 & 1100 & 0010 & 1000 \\
0010 & 0011 & 0110 & 0001 & 0100 \\
0001 & 0001 & 0010 & 1010 & 1001
\end{array}\right).
$$

At line 8 of Table 3, we perform backward substitution on $[\mathbf{G}_0 \quad \mathbf{J}_0]$, and obtain the matrix $[\mathbf{I}_4 \quad \mathbf{M}]$ denoted as follows:

$$
\begin{aligned}
BG([\mathbf{G}_0 \quad \mathbf{J}_0]) &= [\mathbf{I}_4 \quad \mathbf{M}] \\
&= \left(\begin{array}{c|c}
1000 & 1111\ 0001\ 0101\ 1011 \\
0100 & 0111\ 1110\ 1000\ 0001 \\
0010 & 0011\ 0110\ 0001\ 0100 \\
0001 & 0001\ 0010\ 1010\ 1001
\end{array}\right).
\end{aligned}
$$

The function $MAT()$ outputs the block matrix $\mathbf{M}$ as a result. We recover all divided pieces of the secret at line 6 of Table 2 with $\mathbf{M}$ and $\mathbf{w}$ by the operation $(s_1, s_2, s_3, s_4)^{\mathrm{T}} = \mathbf{M} \cdot \mathbf{w}$, where each divided piece of the secret is obtained by the following XOR operations:

$$
\begin{aligned}
s_1 &= w_{(0,0)} \oplus w_{(0,1)} \oplus w_{(0,2)} \oplus w_{(0,3)} \oplus w_{(1,3)} \oplus w_{(2,1)} \oplus w_{(2,3)} \oplus w_{(4,0)} \oplus w_{(4,2)} \oplus w_{(4,3)}, \\
s_2 &= w_{(0,1)} \oplus w_{(0,2)} \oplus w_{(0,3)} \oplus w_{(1,0)} \oplus w_{(1,1)} \oplus w_{(1,2)} \oplus w_{(2,0)} \oplus w_{(4,3)}, \\
s_3 &= w_{(0,2)} \oplus w_{(0,3)} \oplus w_{(1,1)} \oplus w_{(1,2)} \oplus w_{(2,3)} \oplus w_{(4,1)}, \\
s_4 &= w_{(0,3)} \oplus w_{(1,2)} \oplus w_{(2,0)} \oplus w_{(2,2)} \oplus w_{(4,0)} \oplus w_{(4,3)},
\end{aligned}
$$

respectively. Thus, we have recovered the secret $s = s_1 \parallel s_2 \parallel s_3 \parallel s_4$.

Table 5. Structure of shares in $(4, 5)$-threshold scheme for $n_p = 5$

	$j = 0$	$j = 1$	$j = 2$	$j = 3$
$w_{(0,j)}$	$r_0^0 \oplus r_0^1 \oplus r_0^2 \oplus s_0$	$r_1^0 \oplus r_1^1 \oplus r_1^2 \oplus s_1$	$r_2^0 \oplus r_2^1 \oplus r_2^2 \oplus s_2$	$r_3^0 \oplus r_3^1 \oplus r_3^2 \oplus s_3$
$w_{(1,j)}$	$r_0^0 \oplus r_1^1 \oplus r_2^2 \oplus s_4$	$r_1^0 \oplus r_2^1 \oplus r_3^2 \oplus s_0$	$r_2^0 \oplus r_3^1 \oplus r_4^2 \oplus s_1$	$r_3^0 \oplus r_4^1 \oplus r_0^2 \oplus s_2$
$w_{(2,j)}$	$r_0^0 \oplus r_2^1 \oplus r_4^2 \oplus s_3$	$r_1^0 \oplus r_3^1 \oplus r_0^2 \oplus s_4$	$r_2^0 \oplus r_4^1 \oplus r_1^2 \oplus s_0$	$r_3^0 \oplus r_0^1 \oplus r_2^2 \oplus s_1$
$w_{(3,j)}$	$r_0^0 \oplus r_3^1 \oplus r_1^2 \oplus s_2$	$r_1^0 \oplus r_4^1 \oplus r_2^2 \oplus s_3$	$r_2^0 \oplus r_0^1 \oplus r_3^2 \oplus s_4$	$r_3^0 \oplus r_1^1 \oplus r_4^2 \oplus s_0$
$w_{(4,j)}$	$r_0^0 \oplus r_4^1 \oplus r_3^2 \oplus s_1$	$r_1^0 \oplus r_0^1 \oplus r_4^2 \oplus s_2$	$r_2^0 \oplus r_1^1 \oplus r_0^2 \oplus s_3$	$r_3^0 \oplus r_2^1 \oplus r_1^2 \oplus s_4$

3.3 The Proof of the Ideal Secret Sharing Scheme

Here, we introduce the following two theorems.

Theorem 1. *Let A denote an arbitrary set of participants such that $|A| \leq k-1$. Then, since A is not in Γ of our proposed scheme, we have*

$$H(S|\mathcal{V}_A) = H(S), \tag{3}$$

where $\mathcal{V}_A$ denotes a set of random variables of shares that are given to each participant in A.

Proof (proof of Theorem 1). Let $A = \{P_{t_0}, \ldots, P_{t_{k-2}}\}$ denote a set of $k-1$ participants, where $t_0, \ldots, t_{k-2} \in GF(n_p)$ are arbitrary numbers such that $0 \leq t_i, t_j \leq n-1$ and $t_i \neq t_j$ if $i \neq j$. Correspondingly, let $\mathcal{V}_A = \{W_{t_0}, \ldots, W_{t_{k-2}}\}$ denote a set of $k-1$ random variables, where $W_{t_0}, \ldots, W_{t_{k-2}}$ are induced by $w_{t_0}, \ldots, w_{t_{k-2}}$, respectively. And also, $W_{(t_i,0)}, \ldots, W_{(t_i,n_p-2)}$ denotes random variables induced by divided pieces of shares $w_{(t_i,0)}, \ldots, w_{(t_i,n_p-2)}$.

The following condition is supposed: $s_1, \ldots, s_{n_p-1}, r_0^0, \ldots, r_{n_p-2}^0, \ldots, r_0^{k-2}, \ldots, r_{n_p-1}^{k-2}$ are independent of each other. And, $r_0^0, \ldots, r_{n_p-2}^0, \ldots, r_0^{k-2}, \ldots, r_{n_p-1}^{k-2}$ are chosen from the finite set $\{0,1\}^d$ with uniform probability $1/2^d$.

We define generator matrices $\mathbf{U}$ and $\mathbf{V}$ which satisfy the following equation:

$$\begin{aligned} \mathbf{w} &= \mathbf{U} \cdot \mathbf{r} \oplus \mathbf{V} \cdot \mathbf{s}, \\ &= (w_{(t_0,0)}, \ldots, w_{(t_0,n_p-2)}, \ldots, w_{(t_{k-2},0)}, \ldots, w_{(t_{k-2},n_p-2)})^{\mathrm{T}}, \end{aligned} \tag{4}$$

where $\mathbf{r}$ and $\mathbf{s}$ are denoted by

$$\begin{aligned} \mathbf{r} &= (r_0^0, \ldots, r_{n_p-2}^0, r_0^1, \ldots, r_{n_p-1}^1, \ldots, r_0^{k-2}, \ldots, r_{n_p-1}^{k-2})^{\mathrm{T}}, \\ \mathbf{s} &= (s_1, \ldots, s_{n_p-1})^{\mathrm{T}}, \end{aligned}$$

respectively. Then, $\mathbf{U}$ and $\mathbf{V}$ are $(k-1)(n_p-1) \times (kn_p-1)$ and $(k-1)(n_p-1) \times (n_p-1)$ matrices, respectively. From Lemma 1, all rows of $\mathbf{U}$ are linearly independent. Therefore, from Lemma 6, all the elements of the $(k-1)(n_p-1)$

dimensional vector obtained by $\mathbf{U}\cdot\mathbf{r}$ are d-bit random numbers which are pairwise independent and uniformly distributed over $\{0,1\}^d$. Thus, the vector $\mathbf{U}\cdot\mathbf{r}$ is uniformly distributed over $\{0,1\}^{d(n_p-1)(k-1)}$. We suppose that $\mathbf{w}'$ denotes a fixed value of $\mathbf{w}$. Then, Eq.(4) means that $\mathbf{w}$, which equals $\mathbf{w}'$, can be obtained with uniform probability $(1/2)^{d(n_p-1)(k-1)}$ from any arbitrary chosen $\mathbf{s}$ (and hence $\mathbf{V}\cdot\mathbf{s}$). Therefore, since $\mathbf{s}$ is independent from $\mathbf{w}$, we have

$$\begin{aligned}&H(S_1,\ldots,S_{n_p-1}|W_{(t_0,0)},\ldots,W_{(t_0,n_p-2)},\ldots,W_{(t_{k-2},0)},\ldots,W_{(t_{k-2},n_p-2)})\\&=H(S|W_{t_0},\ldots,W_{t_{k-2}})\\&=H(S_1,\ldots,S_{n_p-1})=H(S).\end{aligned}$$

Therefore, $H(S|\mathcal{V}_A)=H(S)$ is satisfied. □

Theorem 2. *Let A denote an arbitrary set of participants such that $|A|\geq k$. Then, the recovery algorithm shown in Table 2 and Table 3 can recover all the divided pieces of the secret from shares given to each participant in A.*

Proof (proof of Theorem 2). Let $A=\{P_{t_0},\ldots,P_{t_{k-1}}\}$ denote a set of k participants, where $t_0,\ldots,t_{k-1}\in GF(n_p)$ are arbitrary numbers such that $0\leq t_i,t_j\leq n-1$ and $t_i\neq t_j$ if $i\neq j$. We define generator matrices $\mathbf{U}$ and $\mathbf{V}$ which satisfy the following equation:

$$\begin{aligned}\mathbf{w}&=\mathbf{U}\cdot\mathbf{r}\oplus\mathbf{V}\cdot\mathbf{s},\\&=(w_{(t_0,0)},\ldots,w_{(t_0,n_p-2)},\ldots,w_{(t_{k-1},0)},\ldots,w_{(t_{k-1},n_p-2)})^{\mathrm{T}},\end{aligned}\tag{5}$$

where $\mathbf{r}$ and $\mathbf{s}$ are denoted by

$$\begin{aligned}\mathbf{r}&=(r_0^0,\ldots,r_{n_p-2}^0,r_0^1,\ldots,r_{n_p-1}^1,\ldots,r_0^{k-2},\ldots,r_{n_p-1}^{k-2})^{\mathrm{T}},\\\mathbf{s}&=(s_1,\ldots,s_{n_p-1})^{\mathrm{T}},\end{aligned}$$

respectively. Then, $\begin{bmatrix}\mathbf{U}\ \mathbf{V}\end{bmatrix}$ equals the generator matrix $\mathbf{G}$ at line 6 of Table 3.

We consider the elementary row operation on $\begin{bmatrix}\mathbf{U}\ \mathbf{V}\end{bmatrix}$. Then, from Remark 1, we can obtain $\begin{bmatrix}\bar{\mathbf{U}}\ \bar{\mathbf{V}}\end{bmatrix}$ from $\begin{bmatrix}\mathbf{U}\ \mathbf{V}\end{bmatrix}$ by the elementary row operation, which satisfies the following equation:

$$\begin{bmatrix}\bar{\mathbf{U}}\ \bar{\mathbf{V}}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{r}\\\mathbf{s}\end{bmatrix}=\left(\begin{array}{c}*\\\vdots\\*\\\hline(s_\alpha\oplus s_\beta)\\(s_{\alpha+1}\oplus s_{\beta+1})\\\vdots\\(s_{\alpha-2}\oplus s_{\beta-2})\end{array}\right),$$

where α and β are denoted by

$$\alpha=-\sum_{i=0}^{k-3}t_i-t_{k-2}+t_{k-1},\qquad\beta=-\sum_{i=0}^{k-3}t_i+t_{k-2}-t_{k-1},$$

respectively. Thus, by the XOR operations with pieces of k shares, we can obtain all the elements of the set $\mathcal{X}$ denoted by

$$\mathcal{X} = \{s_{\alpha+m} \oplus s_{\beta+m} \mid 0 \leq m \leq n_p - 2\}.$$

Then, since indexes are elements of $GF(n_p)$, we can also obtain $s_{\alpha-1} \oplus s_{\beta-1}$ from all the elements of $\mathcal{X}$ as follows:

$$s_{\alpha-1} \oplus s_{\beta-1} = \bigoplus_{m=0}^{n_p-2} (s_{\alpha+m} \oplus s_{\beta+m}).$$

Hence, we can consider the set $\mathcal{X}'$ to recover the secret, which is defined by

$$\mathcal{X}' = \{x_m = s_{\alpha+m} \oplus s_{\beta+m} \mid 0 \leq m \leq n_p - 1\}.$$

Then, the following set

$$\{pC \mid 0 \leq p \leq n_p - 1\} \pmod{n_p} = GF(n_p),$$

is a field with order n_p from Lemma 5, where $C = 2(t_{k-2} - t_{k-1})$. Thus, the following equation is satisfied:

$$\begin{aligned} \{C, 2C, \ldots, (n_p - 1)C\} &= \{1, \ldots, n_p - 1\} \pmod{n_p} \\ &= GF(n_p) \backslash \{0\}. \end{aligned} \tag{6}$$

$s_0 = 0^d$ was inserted as a singular point. Therefore, we can recover all the divided pieces of the secret sequentially as follows:

$$\begin{aligned} m &= -\alpha && : s_C = x_{-\alpha}, \\ m &= C - \alpha && : s_{2C} = x_{C-\alpha} \oplus s_C, \\ m &= 2C - \alpha && : s_{3C} = x_{2C-\alpha} \oplus s_{2C}, \\ &\vdots && \qquad \vdots \\ m &= (n_p - 1)C - \alpha && : s_{(n_p-1)C} = x_{(n_p-1)C-\alpha} \oplus s_{(n_p-2)C}, \end{aligned}$$

and since only XOR operations using the property of $GF(n_p)$ are used, this sequential operation can be represented by the elementary row operation on $\left[\bar{\mathbf{U}}\ \bar{\mathbf{V}}\right]$. From Eq.(6), the following equation is satisfied:

$$\left\{s_C, s_{2C}, \ldots, s_{(n_p-1)C}\right\} = \left\{s_1, s_2, \ldots, s_{n_p-1}\right\}.$$

Thus, all the divided pieces of the secret can be recovered from k shares by using an elementary row operation on $\left[\mathbf{U}\ \mathbf{V}\right]$. The elementary row operations do not change the solution set of the system of linear equations represented by a matrix. If it is possible to obtain the solution set by arbitrary elementary row operations, the corollary solution set is the same as the solution set obtained by Gaussian elimination. Therefore, our recovery algorithm using Gaussian elimination at lines 7-8 of Table 3 can recover all the divided pieces of the secret from k shares. □

Table 6. Simulation Environment and Conditions

CPU / RAM	: Pentium 4 3.0GHz / 2.0GB
Operating system	: Debian GNU/Linux 4.0
Compiler	: GCC 4.1
Source of random numbers	: /dev/urandom
Size of the secret s	: 4.5MB
(k, n)	: $(3, 11)$, $(3, 59)$, $(3, 109)$, $(5, 11)$, $(10, 11)$, $(10, 23)$
Implementation of Shamir's scheme	: SSSS Version 0.5[19]
Operating unit in Shamir's scheme	: 8 byte/operation

Theorem 2 means that the following equation is satisfied if $|A| \geq k$:

$$H(S|\mathcal{V}_A) = 0,$$

where $\mathcal{V}_A$ denotes a set of random variables of shares that are given to each participant in A. Thus, from these two theorems, Eq.(1) is satisfied, and the access structure Γ of our scheme is denoted by $\Gamma = \{A \in 2^{\mathcal{P}} \mid |A| \geq k\}$. Furthermore, since every bit-size of shares equals the bit-size of the secret if we can suppose that $s \in \{0, 1\}^{d(n_p-1)}$ $(d > 0)$, i.e. the size of the secret is $d(n_p - 1)$ bits, [2] the information rate ρ equals one. Thus, as with Shamir's scheme, our scheme is also *ideal*.

4 Evaluation of Efficiency

In this section, we evaluate the efficiency of our scheme by comparing it with Shamir's scheme. First, we show the result of computer simulation by implementing both our scheme and Shamir's. Next, we consider the two schemes from the perspective of computational cost.

Computer Simulation. We compared the proposed scheme with that of Shamir's for $(k, n) = (3, 11)$, $(3, 59)$, $(3, 109)$, $(5, 11)$, $(10, 11)$ and $(10, 23)$ by implementation on a PC, where every scheme is implemented for $n = n_p$. Fig.1 denotes the processing time required to make $n(= n_p)$ shares from 4.5 MB data (secret) and recover the 4.5 MB secret from k shares, $w_0, \cdots, w_{k-1}$ by using our scheme and Shamir's scheme. The simulation environment and conditions are summarized in Table 6. For the implementation of Shamir's scheme, we used SSSS Version 0.5[19], which is a free software licensed under the GNU GPL. An 8-byte block was processed in each cycle in the distribution and recovery operations under Shamir's scheme. In Fig.1, the horizontal axis and vertical axis denote pairs of

[2] If the size of the secret s were not multiple of $(n_p - 1)$, it is required to apply padding operations to the secret bit sequence to make shares and hence the bit-size of each share is larger than that of the secret.

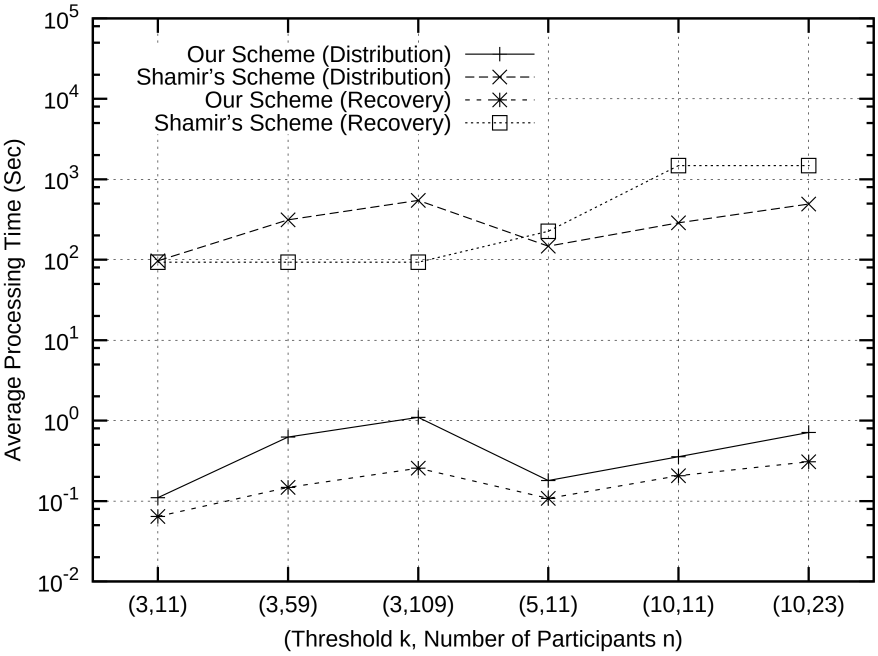

Fig. 1. Distribution and Recovery Processing Time for $n = n_p$

threshold and the number of participants, i.e. (k, n), and the processing time, respectively.

This graph shows that our scheme performed processing much faster than Shamir's. In $(3, 11)$-threshold schemes, our scheme was more than 900-fold faster than Shamir's in terms of both distribution and recovery. Similarly, in $(3, 59)$, $(3, 109)$, $(5, 11)$, $(10, 11)$ and $(10, 23)$-threshold schemes, Fig.1 shows that our scheme achieved far more rapid processing than Shamir's.

Consideration. In our proposed distribution algorithm, step 9 at Table 1 requires $(k-2)d$ bitwise XOR operations to make one divided piece of share $w_{(i,j)}$ which is constructed with s_0, or else, $(k-1)d$ bitwise XOR operations to make $w_{(i,j)}$ constructed without s_0. Thus, $(n_p-2)(k-1)d + (k-2)d$ XOR operations are required to make each share of $w_0, \ldots, w_{n_p-2}$. Furthermore, $(n_p-1)(k-1)d$ XOR operations are required to make w_{n_p-1}. Hence, the average number of XOR operations to make one share is $\left\{(k-1) - \frac{1}{n_p}\right\} \cdot \log_2 |\mathcal{S}|$. Therefore, our distribution algorithm requires an average of

$$\left\{(k-1) - \frac{1}{n_p}\right\} n \cdot \log_2 |\mathcal{S}| = O(kn) \cdot \log_2 |\mathcal{S}|,$$

bitwise XOR operations to make n shares. If $n = n_p$, it equals $\{(k-1)n - 1\} \cdot \log_2 |\mathcal{S}|$. Since the cost of modulo n_p operations on indexes can be regard as being negligible by using the fixed generator matrix in a manner similar to the recovery algorithm, we omit the cost of the operations here for the sake of simplicity.

On the other hand, in the proposed recovery algorithm, we can assume that at the most $\{k(n_p-1)-1\}d$ XOR operations are required to recover one of the divided pieces of the secret with all divided pieces of k shares, and at the most $\{k(n_p-1)-2\}d$ XOR operations are required to recover one of the other divided pieces of the secret with $k(n_p-1)-1$ divided pieces of k shares. Thus, the upper bound of the number of XOR operations required to recover the secret by using a block matrix $\mathbf{M}$ is roughly denoted by

$$\left\{k(n_p-1) - \frac{2n_p-3}{n_p-1}\right\} \cdot \log_2 |\mathcal{S}| = O(kn_p) \cdot \log_2 |\mathcal{S}|.$$

The recovery algorithm also requires $O(k^3{n_p}^3)$ bitwise XOR operations to execute forward elimination (step 7 of Table 3) and partial backward substitution (step 8 of Table 3) of Gaussian elimination as a pre-computation cost to obtain $\mathbf{M}$ at the function $MAT()$.

On the other hand, in Shamir's scheme, $O(kn)$ and $O(k \log^2 k)$ arithmetic operations are required to make shares and recover the secret, respectively[1].

From Fig.1, it is evident that the processing time for distribution in both Shamir's and our scheme is linearly increasing with each of k and n. However, though the processing time for recovery in Shamir's scheme is constant and independent of n if threshold k is fixed, that of our scheme increases as the number of participants $n(= n_p)$ grows in Fig.1. The computational cost of recovery in Shamir's scheme depends only on k, but that in our scheme depends on both k and n_p. Thus, though our scheme is much more efficient than Shamir's for not so large n_p as shown in Fig.1, our scheme will not perform faster processing to recover the secret than Shamir's if n_p is extremely large. We will determine the upper bound of n_p for the value of k as a future work, in which our scheme will be shown to be faster than Shamir's.

5 How to Extend Our Scheme to a Fast *Ramp* Scheme

In terms of improved efficiency for both computational cost and storage usage, a (k, L, n)-threshold *ramp* scheme[4, 3] based on our (k, n)-threshold scheme can be realized. In this section, we briefly show how the new *ramp* scheme can be constructed.

In the distribution phase of our (k, L, n)-threshold *ramp* scheme $(1 \leq L \leq k-1)$, the differences from our (k, n)-threshold scheme can be summarized as follows:

- The secret $s \in \{0,1\}^{dL(n_p-1)}$ is equally divided into $L(n_p-1)$ pieces $s_0^0,\ldots,s_{n_p-2}^0$ $,\ldots,s_0^{L-1},\ldots,s_{n_p-2}^{L-1} \in \{0,1\}^d$. And, the *singular points* in divided pieces of the secret are $s_{n_p-1}^0, \ldots, s_{n_p-1}^{L-1} = 0^d$.

- To make n shares, the dealer $\mathcal{D}$ generates $k-L$ sets of random numbers $\{r_0^0, \dots, r_{n_p-2}^0\}$, $\{r_0^1, \dots, r_{n_p-1}^1\}$, $\dots$, $\{r_0^{k-L-1}, \dots, r_{n_p-1}^{k-L-1}\}$, where the bit-size of each element in every set is d.
- The dealer makes pieces of shares $w_{(i,j)}$ by the following equation:

$$w_{(i,j)} = \left(\bigoplus_{h=0}^{k-L-1} r_{h \cdot i+j}^h \right) \oplus \left(\bigoplus_{h=0}^{L-1} s_{(k-L+h) \cdot i+j}^h \right).$$

The above differences mean that the *ramp* scheme can be realized by replacing $L-1$ sets of random numbers by an $L-1$ set of divided pieces of the secret, where each set of divided pieces of the secret has a singular point. On the other hand, we can recover the secret from k shares by similar recovery algorithm to our (k,n)-threshold scheme. The differences in the recovery phase are only the area of the generator matrix on which the partial backward substitution is performed and hence the size of matrix $\mathbf{M}$.

Then, the bit-size of each share is $1/L$ of the bit-size of the secret, and the efficiency in terms of computational cost for both distribution and recovery is as follows: An average of $O(kn) \cdot \dfrac{\log_2 |\mathcal{S}|}{L}$ bitwise XOR operations is required to make n shares. To generate matrix $\mathbf{M}$, $O(k^3 n_p^3)$ bitwise XOR operations are required. Also, the upper bound of bitwise XOR operations to recover the secret by using $\mathbf{M}$ is $O(kn_p) \cdot \log_2 |\mathcal{S}|$.

In a manner similar to [4], it can be proved that the security property of this *ramp* scheme is same as Yamamoto's *ramp* scheme based on Shamir's scheme.

6 Conclusion

In this paper, we proposed a new (k,n)-threshold secret sharing scheme which uses just XOR operations to make shares and recover the secret, and we proved that the proposed scheme is an *ideal* secret sharing scheme. We estimated the computational cost in our scheme and Shamir's scheme for values of k and n. Also, we implemented our scheme on a PC for specific parameters, and showed that our scheme was more efficient than Shamir's in terms of computational cost provided n is not extremely large. Moreover, we introduced an extension of our scheme to a new (k,L,n)-threshold *ramp* scheme, which can realize both fast computation and reduction of storage usage.

References

1. A. Shamir, "How to share a secret," Commun. ACM, vol.22, no.11, pp.612–613, 1979.
2. G. R. Blakley, "Safeguarding cryptographic keys," Proc. AFIPS, vol.48, pp.313–317, 1979.
3. G. R. Blakley and C. Meadows, "Security of ramp schemes," Proc. CRYPTO '84, LNCS 196, Springer-Verlag, pp.242–269, 1985.

4. H. Yamamoto, "On secret sharing systems using (k, L, n) threshold scheme," IEICE Trans. Fundamentals (Japanese Edition), vol.J68-A, no.9, pp.945–952, 1985.
5. K. Kurosawa, K. Okada, K. Sakano, W. Ogata, and T. Tsujii, "Non perfect secret sharing schemes and matroids," Proc. EUROCRYPT '93, LNCS 765, Springer-Verlag, pp.126–141, 1993.
6. W. Ogata and K. Kurosawa, "Some basic properties of general nonperfect secret sharing schemes," J. Universal Computer Science, vol.4, no.8, pp.690–704, 1998.
7. K. Okada and K. Kurosawa, "Lower bound on the size of shares of nonperfect secret sharing schemes," Proc. ASIACRYPT '94, LNCS 917, Springer-Verlag, pp.34–41, 1994.
8. H. Ishizu and T. Ogihara, "A study on long-term storage of electronic data," Proc. IEICE General Conf., D-9-10, no.1, p.125, 2004. (in Japanese)
9. Y. Fujii, M. Tada, N. Hosaka, K. Tochikubo, and T. Kato, "A fast $(2, n)$-threshold scheme and its application," Proc. CSS2005, pp.631-636, 2005. (in Japanese)
10. N. Hosaka, K. Tochikubo, Y. Fujii, M. Tada, and T. Kato, "$(2, n)$-threshold secret sharing systems based on binary matrices," Proc. SCIS2007, 2D1-4, 2007. (in Japanese)
11. J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A fast $(3, n)$-threshold secret sharing scheme using exclusive-or operations," IEICE Trans. Fundamentals, vol.E91-A, no.1, pp.127-138, Jan. 2008.
12. N. Shiina, T. Okamoto, and E. Okamoto, "How to convert 1-out-of-n proof into k-out-of-n proof," Proc. SCIS2004, pp.1435–1440, 2004. (in Japanese)
13. H. Kunii and M. Tada, "A note on information rate for fast threshold schemes," Proc. CSS2006, pp.101–106, 2006. (in Japanese)
14. E. D. Karnin, J. W. Greene, and M. E. Hellman, "On secret sharing systems," IEEE Trans. Inform. Theory, vol.29, no.1, pp35–41, 1983.
15. R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro, "On the size of shares for secret sharing schemes," J. Cryptology, vol.6, pp.35–41, 1983.
16. C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro, "On the information rate of secret sharing schemes," Proc. CRYPTO '92, LNCS 740, Springer-Verlag, pp.149–169, 1993.
17. D. R. Stinson, "Decomposition constructions for secret sharing schemes," IEEE Trans. Inform. Theory, vol.40, no.1, pp.118–125, 1994.
18. D. R. Stinson, Cryptography: Theory and Practice, CRC Press, Florida, 1995.
19. B. Poettering, "SSSS: Shamir's Secret Sharing Scheme," `http://point-at-infinity.org/ssss/`.

Appendix 1 Lemma 1 - Linearly Independent and Dependent

Lemma 1. *Let $t_0, \dots, t_{L-1} \in GF(n_p)$ denote indexes of L shares, which are arbitrary numbers such that $0 \leq t_i, t_j \leq n-1$ and $t_i \neq t_j$ if $i \neq j$. The matrices $\mathbf{U}$ and $\mathbf{V}$ denote generator matrices of $L(n_p - 1)$ pieces of L shares such that*

$$\begin{aligned}
\mathbf{w} &= \mathbf{U} \cdot \mathbf{r} \oplus \mathbf{V} \cdot \mathbf{s}' \\
&= \left(w_{(t_0,0)}, \dots, w_{(t_0,n_p-2)}, \dots, w_{(t_{L-1},0)}, \dots, w_{(t_{L-1},n_p-2)}\right)^{\mathrm{T}}, \\
\mathbf{r} &= \left(r_0^0, \dots, r_{n_p-2}^0, r_0^1, \dots, r_{n_p-1}^1, \dots, r_0^{k-2}, \dots, r_{n_p-1}^{k-2}\right)^{\mathrm{T}}, \\
\mathbf{s}' &= \left(s_0, s_1, \dots, s_{n_p-1}\right)^{\mathrm{T}},
\end{aligned}$$

where though $s_0 = 0^d$ is a singular point, we include s_0 as a variable in $\mathbf{s}'$ to describe $\mathbf{V}$ briefly.

Then, the following equation is satisfied:

$$rank\left(\begin{bmatrix}\mathbf{U}\ \mathbf{V}\end{bmatrix}\right) = \begin{cases} L(n_p-1) & (1 \leq L \leq k-1) \\ k(n_p-1) & (L \geq k) \end{cases} .$$

Also, we have

$$rank\left(\mathbf{U}\right) = \begin{cases} L(n_p-1) & (1 \leq L \leq k-1) \\ (k-1)(n_p-1) & (L \geq k) \end{cases} .$$

Remark 1. $\begin{bmatrix}\mathbf{U}\ \mathbf{V}\end{bmatrix}$ can be transformed into $\begin{bmatrix}\bar{\mathbf{U}}\ \bar{\mathbf{V}}\end{bmatrix}$ by the elementary row operation if $L = k$, which satisfies the following equation:

$$\begin{bmatrix}\bar{\mathbf{U}}\ \bar{\mathbf{V}}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{r} \\ \mathbf{s}'\end{bmatrix} = \begin{pmatrix} * \\ \vdots \\ * \\ \hline (s_\alpha \oplus s_\beta) \\ (s_{\alpha+1} \oplus s_{\beta+1}) \\ \vdots \\ (s_{\alpha-2} \oplus s_{\beta-2}) \end{pmatrix}, \tag{7}$$

where α and β are denoted by

$$\alpha = -\sum_{i=0}^{k-3} t_i - t_{k-2} + t_{k-1}, \qquad \beta = -\sum_{i=0}^{k-3} t_i + t_{k-2} - t_{k-1},$$

respectively. We also describe the proof of this remark in the proof of Lemma 1.

Proof. $\mathbf{U}$ and $\mathbf{V}$ can be denoted by

$$\mathbf{U} = \begin{pmatrix} \mathbf{I}_{n_p-1} & \mathbf{E}_{(t_0)} & \mathbf{E}_{(2t_0)} & \cdots & \mathbf{E}_{((k-2)t_0)} \\ \mathbf{I}_{n_p-1} & \mathbf{E}_{(t_1)} & \mathbf{E}_{(2t_1)} & \cdots & \mathbf{E}_{((k-2)t_1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{n_p-1} & \mathbf{E}_{(t_{L-1})} & \mathbf{E}_{(2t_{L-1})} & \cdots & \mathbf{E}_{((k-2)t_{L-1})} \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{E}_{((n_p-1)t_0)} \\ \mathbf{E}_{((n_p-1)t_1)} \\ \vdots \\ \mathbf{E}_{((n_p-1)t_{L-1})} \end{pmatrix},$$

respectively. $\mathbf{I}_{n_p-1}$ denotes an $(n_p-1)\times(n_p-1)$ identity matrix and $\mathbf{E}_{(j)}$ $(j \in GF(n_p))$ denotes the following $(n_p-1) \times n_p$ matrix:

$$\mathbf{E}_{(j)} = \left(\begin{array}{c} \mathbf{i}_j \\ \mathbf{i}_{j+1} \\ \vdots \\ \mathbf{i}_{n_p-1} \\ \hline \mathbf{i}_0 \\ \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_{j-2} \end{array}\right) = \left(\begin{array}{ccc|c|ccc} 0 & \cdots & 0 & 0 & & & \\ \vdots & \ddots & \vdots & \vdots & & \mathbf{I}_{n_p-j} & \\ 0 & \cdots & 0 & 0 & & & \\ \hline & & & 0 & 0 & \cdots & 0 \\ & \mathbf{I}_{j-1} & & \vdots & \vdots & \ddots & \vdots \\ & & & 0 & 0 & \cdots & 0 \end{array}\right).$$

First, we consider the elementary row operation on $\begin{bmatrix} \mathbf{U} \; \mathbf{V} \end{bmatrix}$. The matrix $\mathbf{U}$ can be transformed into the following matrix $\mathbf{U}'$ by the elementary row operation:

$$\mathbf{U}' = \left(\begin{array}{c|cccc} \mathbf{I}_{n_p-1} & \mathbf{E}_{(t_0)} & \mathbf{E}_{(2t_0)} & \cdots & \mathbf{E}_{((k-2)t_0)} \\ \hline \emptyset & \mathbf{E}^{(2)}_{(t_0,t_1)} & \mathbf{E}^{(2)}_{(2t_0,2t_1)} & \cdots & \mathbf{E}^{(2)}_{((k-2)t_0,(k-2)t_1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \emptyset & \mathbf{E}^{(2)}_{(t_0,t_{L-1})} & \mathbf{E}^{(2)}_{(2t_0,2t_{L-1})} & \cdots & \mathbf{E}^{(2)}_{((k-2)t_0,(k-2)t_{L-1})} \end{array}\right)$$

$$= \left(\begin{array}{c|c} \mathbf{I}_{n_p-1} & * \\ \hline \emptyset & \\ \vdots & \mathbf{U}'' \\ \emptyset & \end{array}\right),$$

where $\mathbf{E}^{(2)}_{(i,j)} = \mathbf{E}_{(i)} \oplus \mathbf{E}_{(j)}$. Correspondingly, $\mathbf{V}$ is transformed into $\mathbf{V}'$ as follows:

$$\mathbf{V}' = \left(\begin{array}{c} \mathbf{E}_{((n_p-1)t_0)} \\ \hline \mathbf{E}^{(2)}_{((n_p-1)t_0,(n_p-1)t_1)} \\ \vdots \\ \mathbf{E}^{(2)}_{((n_p-1)t_0,(n_p-1)t_{L-1})} \end{array}\right) = \left(\begin{array}{c} \mathbf{E}_{((n_p-1)t_0)} \\ \hline \mathbf{V}'' \end{array}\right).$$

Next, we consider the concatenated block matrix $\mathbf{D}$ which is defined by

$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} \mathbf{U}'' \; \mathbf{V}'' \end{bmatrix} \\ &= \left(\begin{array}{ccc|c} \mathbf{E}^{(2)}_{(t_0,t_1)} & \cdots & \mathbf{E}^{(2)}_{((k-2)t_0,(k-2)t_1)} & \mathbf{E}^{(2)}_{((n_p-1)t_0,(n_p-1)t_1)} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{E}^{(2)}_{(t_0,t_{L-1})} & \cdots & \mathbf{E}^{(2)}_{((k-2)t_0,(k-2)t_{L-1})} & \mathbf{E}^{(2)}_{((n_p-1)t_0,(n_p-1)t_{L-1})} \end{array}\right). \end{aligned}$$

Then, from Lemma 4, the rank of $\mathbf{E}^{(2)}_{i,j}$ $(i \neq j)$ is $n_p - 1$. Here, for the sake of simple description of the elementary row operation on $\mathbf{D}$, we define the $n_p \times n_p$ matrix $\mathbf{H}_{(i,j)}$ by adding one row to $\mathbf{E}^{(2)}_{(i,j)}$ with all rows of $\mathbf{E}^{(2)}_{(i,j)}$ as follows:

$$\mathbf{H}_{(i,j)} = \left(\begin{array}{c} \mathbf{E}^{(2)}_{(i,j)} \\ \bigoplus_{l=0}^{n_p-2} (\mathbf{i}_{i+l} \oplus \mathbf{i}_{j+l}) \end{array}\right).$$

Since indexes are elements of $GF(n_p)$, $\mathbf{H}_{(i,j)}$ can be denoted as follows:

$$\mathbf{H}_{(i,j)} = \begin{pmatrix} \mathbf{E}^{(2)}_{(i,j)} \\ \mathbf{i}_{i-1} \oplus \mathbf{i}_{j-1} \end{pmatrix} = \mathbf{L}_i \oplus \mathbf{L}_j,$$

where $\mathbf{L}_i$ is the rotated identity matrix which is defined by

$$\mathbf{L}_i = \begin{pmatrix} \mathbf{E}_{(i)} \\ \mathbf{i}_{i-1} \end{pmatrix} = \begin{pmatrix} \text{\O} & \mathbf{I}_{n_p-i} \\ \mathbf{I}_i & \text{\O} \end{pmatrix}.$$

Thus, $\mathbf{L}_i \cdot \mathbf{L}_j = \mathbf{L}_j \cdot \mathbf{L}_i = \mathbf{L}_{i+j}$ is satisfied. Then, we consider the following matrix $\mathbf{M}$ to describe briefly the elementary row operation on $\mathbf{D}$, which is defined as follows:

$$\begin{aligned}
\mathbf{M} &= \left[\mathbf{P}\ \mathbf{Q}\right] \\
&= \begin{pmatrix} \mathbf{M}_{(1,1)} \\ \vdots \\ \mathbf{M}_{(1,L-1)} \end{pmatrix} = \left(\begin{array}{c|c} \mathbf{P}_1 & \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_1)} \\ \vdots & \vdots \\ \mathbf{P}_{L-1} & \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_{L-1})} \end{array}\right), \\
\mathbf{P} &= \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{L-1} \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{(t_0,t_1)} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_1)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{(t_0,t_{L-1})} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_{L-1})} \end{pmatrix}, \\
\mathbf{Q} &= \begin{pmatrix} \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_1)} \\ \vdots \\ \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_{L-1})} \end{pmatrix}.
\end{aligned}$$

Since the n_p-th row of $\mathbf{H}_{(i,j)}$ is generated from all rows of $\mathbf{E}^{(2)}_{(i,j)}$, $\mathbf{M}$ is equivalent to $\mathbf{D}$. And hence, $\mathbf{P}$ and $\mathbf{Q}$ are equivalent to $\mathbf{U}''$ and $\mathbf{V}''$, respectively. Thus, the following equations are satisfied:

$$\begin{aligned}
rank\left(\mathbf{D}\right) &= rank\left(\mathbf{M}\right), \\
rank\left(\mathbf{U}''\right) &= rank\left(\mathbf{P}\right), \\
rank\left(\mathbf{V}''\right) &= rank\left(\mathbf{Q}\right).
\end{aligned}$$

The rank of block matrix $\mathbf{M}_{(1,l)}$ $(1 \le l \le L-1)$ equals $n_p - 1$ from Lemma 4. From Lemma 2, $\mathbf{M}$ can be transformed into the following matrix $\bar{\mathbf{M}}$ by the elementary row operation if $1 \le L \le k-1$:

$$\begin{aligned}
\bar{\mathbf{M}} &= \left[\bar{\mathbf{P}}\ \bar{\mathbf{Q}}\right] \\
&= \begin{pmatrix} \bar{\mathbf{M}}_1 \\ \bar{\mathbf{M}}_2 \\ \vdots \\ \bar{\mathbf{M}}_{L-1} \end{pmatrix} = \left(\begin{array}{c|c} \bar{\mathbf{P}}_1 & \mathbf{H}_{(-t_0,-t_1)} \\ \bar{\mathbf{P}}_2 & \mathbf{H}_{(f_2(t_2),g_2(t_2))} \\ \vdots & \vdots \\ \bar{\mathbf{P}}_{L-1} & \mathbf{H}_{(f_{L-1}(t_{L-1}),g_{L-1}(t_{L-1}))} \end{array}\right),
\end{aligned}$$

where $\bar{\mathbf{P}}$ and $\bar{\mathbf{Q}}$ are denoted by

$$\bar{\mathbf{P}} = \begin{pmatrix} \bar{\mathbf{P}}_1 \\ \bar{\mathbf{P}}_2 \\ \vdots \\ \bar{\mathbf{P}}_{L-1} \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{(t_0,t_1)} & * & \cdots & * & * \cdots * \\ \text{Ø} & \mathbf{H}_{(2t_1,2t_2)} & \cdots & * & * \cdots * \\ \vdots & \vdots & \ddots & \vdots & \vdots \ddots \vdots \\ \text{Ø} & \text{Ø} & \cdots & \mathbf{H}_{(2t_{L-2},2t_{L-1})} & * \cdots * \end{pmatrix},$$

$$\bar{\mathbf{Q}} = \begin{pmatrix} \mathbf{H}_{(-t_0,-t_1)} \\ \mathbf{H}_{(f_2(t_2),g_2(t_2))} \\ \vdots \\ \mathbf{H}_{(f_{L-1}(t_{L-1}),g_{L-1}(t_{L-1}))} \end{pmatrix},$$

respectively. $f_m(t_i)$ and $g_m(t_i)$ are denoted by

$$f_m(t_i) = -\sum_{j=0}^{m-2} t_j - t_{m-1} + t_i, \quad g_m(t_i) = -\sum_{j=0}^{m-2} t_j + t_{m-1} - t_i,$$

respectively. From Lemma 2 and Lemma 4, the XORed vectors from the first row to the (n_p-1)-th row of $\bar{\mathbf{M}}_i$ and $\bar{\mathbf{P}}$ $(1 \leq i \leq L-1)$ equal the n_p-th row as follows:

$$\mathbf{m}_{(i,n_p-1)} = \bigoplus_{j=0}^{n_p-2} \mathbf{m}_{(i,j)}, \quad \mathbf{p}_{(i,n_p-1)} = \bigoplus_{j=0}^{n_p-2} \mathbf{p}_{(i,j)},$$

where $\mathbf{m}_{(i,j)}$ and $\mathbf{p}_{(i,j)}$ denote the j-th rows of $\bar{\mathbf{M}}_i$ and $\bar{\mathbf{P}}_i$, denoted as follows, respectively:

$$\bar{\mathbf{M}}_i = \begin{pmatrix} \mathbf{m}_{(i,0)} \\ \vdots \\ \mathbf{m}_{(i,n_p-1)} \end{pmatrix}, \quad \bar{\mathbf{P}}_i = \begin{pmatrix} \mathbf{p}_{(i,0)} \\ \vdots \\ \mathbf{p}_{(i,n_p-1)} \end{pmatrix}.$$

Since the rank of $\mathbf{H}_{(i,j)}$ equals $n_p - 1$, $rank(\bar{\mathbf{M}}_i) = rank(\bar{\mathbf{P}}_i) = n_p - 1$ is satisfied. Hence, from the structure of $\bar{\mathbf{M}}$ and $\bar{\mathbf{P}}$, the following equation is satisfied:

$$rank(\bar{\mathbf{M}}) = rank(\bar{\mathbf{P}}) = (L-1)(n_p-1).$$

Thus, the following equation is satisfied:

$$\begin{aligned} &rank(\mathbf{M}) = rank(\mathbf{D}) = rank(\mathbf{P}) = rank(\mathbf{U}'') \\ &= (L-1)(n_p-1). \end{aligned}$$

Therefore, the rank of $\begin{bmatrix} \mathbf{U}\ \mathbf{V} \end{bmatrix}$ equals $L(n_p-1)$. and all rows of $\mathbf{U}$ are linearly independent, i.e. $rank(\mathbf{U}) = L(n_p-1)$, if $1 \leq L \leq k-1$.

In contrast, from Lemma 2, $\mathbf{M}$ can be transformed into the following matrix $\bar{\mathbf{M}}$ if $L \geq k$:

$$\bar{\mathbf{M}} = \begin{bmatrix} \bar{\mathbf{P}} \ \bar{\mathbf{Q}} \end{bmatrix}$$

$$= \left(\begin{array}{cccc|c} \mathbf{H}_{(t_0,t_1)} & * & \cdots & * & \mathbf{H}_{(-t_0,-t_1)} \\ \emptyset & \mathbf{H}_{(2t_1,2t_2)} & \cdots & * & \mathbf{H}_{(f_2(t_2),g_2(t_2))} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \emptyset & \emptyset & \cdots & \mathbf{H}_{(2t_{k-3},2t_{k-2})} & \mathbf{H}_{(f_{k-2}(t_{k-2}),g_{k-2}(t_{k-2}))} \\ \emptyset & \emptyset & \cdots & \emptyset & \mathbf{H}_{(f_{k-1}(t_{k-1}),g_{k-1}(t_{k-1}))} \\ \emptyset & \emptyset & \cdots & \emptyset & \emptyset \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \emptyset & \emptyset & \cdots & \emptyset & \emptyset \end{array} \right).$$

Thus, from the structure of $\bar{\mathbf{M}}$, the following equations are satisfied:

$$rank(\mathbf{M}) = rank(\mathbf{D}) = (k-1)(n_p-1),$$
$$rank(\mathbf{P}) = rank(\mathbf{U}'') = (k-2)(n_p-1).$$

Therefore, the rank of $\begin{bmatrix} \mathbf{U} \ \mathbf{V} \end{bmatrix}$ equals $k(n_p-1)$ and all rows of $\mathbf{U}$ are linearly dependent, i.e. $rank(\mathbf{U}) = (k-1)(n_p-1)$, if $L \geq k$. Moreover, we can obtain the following vector with $\bar{\mathbf{M}}$:

$$\mathbf{H}_{(f_{k-1}(t_{k-1}),g_{k-1}(t_{k-1}))} \cdot \mathbf{s}' = \begin{pmatrix} (s_{f_{k-2}(t_{k-2})} \oplus s_{g_{k-2}(t_{k-2})}) \\ (s_{f_{k-2}(t_{k-2})+1} \oplus s_{g_{k-2}(t_{k-2})+1}) \\ \vdots \\ (s_{f_{k-2}(t_{k-2})-1} \oplus s_{g_{k-2}(t_{k-2})-1}) \end{pmatrix}.$$

Therefore, by the elementary row operation on $\begin{bmatrix} \mathbf{U} \ \mathbf{V} \end{bmatrix}$, we can obtain the vector denoted at Eq.(7) of Remark 1 if $L = k$. □

Appendix 2 Lemma 2 - The Elementary Row Operations

In this appendix, all definitions, notations and suppositions are same as in Lemma 1.

Lemma 2. *Let* $\mathbf{X}_{(i,j)}^{(h-1)}$ *be a* $n_p \times n_p$ *matrix whose* n_p*-th row equals XORed vector of* 1*st row to* (n_p-1)*-th row of* $\mathbf{X}_{(i,j)}^{(h-1)}$*. And let* $f_m(t_i)$ *and* $g_m(t_i)$ *be*

$$f_m(t_i) = -\sum_{l=0}^{m-2} t_l - t_{m-1} + t_i, \quad g_m(t_i) = -\sum_{l=0}^{m-2} t_l + t_{m-1} - t_i,$$

respectively.

Then, the matrix $\mathbf{M}^{(1)}$

$$\mathbf{M}^{(1)} = \begin{pmatrix} \mathbf{H}_{(t_0,t_1)} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_1)} & \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_1)} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{H}_{(t_0,t_{L-1})} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_{L-1})} & \mathbf{H}_{((n_p-1)t_0,(n_p-1)t_{L-1})} \end{pmatrix},$$

can be transformed into the following matrix by the elementary row operation $(2 \leq m \leq L-1)$*:*

$$\mathbf{M}^{(m)} = \begin{pmatrix} \mathbf{M}^{(m)}_{(1,1)} & \cdots & \mathbf{M}^{(m)}_{(L-1,k-1)} \\ \vdots & \ddots & \vdots \\ \mathbf{M}^{(m)}_{(L-1,1)} & \cdots & \mathbf{M}^{(m)}_{(L-1,k-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^{(m)}_{1} \\ \vdots \\ \mathbf{M}^{(m)}_{L-1} \end{pmatrix},$$

where $\mathbf{M}^{(m)}_{(i,j)}$ *can be denoted by*

$$\mathbf{M}^{(m)}_{(i,j)} = \begin{cases} \mathbf{H}_{(jt_0, jt_1)} & (i=1,\ 1 \leq j \leq k-2), \\ \mathbf{H}_{(-t_0, -t_1)} & (i=1,\ j=k-1), \\ \mathbf{H}_{(2t_{i-1}, 2t_i)} & (2 \leq i \leq m,\ j=i), \\ \mathbf{H}_{(2t_{m-1}, 2t_i)} & (m+1 \leq i \leq L-1,\ j=m), \\ \mathbf{H}_{(f_i(t_i), g_i(t_i))} & (2 \leq i \leq m,\ j=k-1), \\ \mathbf{H}_{(f_m(t_i), g_m(t_i))} & (m+1 \leq i \leq L-1,\ j=k-1), \\ \mathbf{X}^{(i-1)}_{(i,j)} & (2 \leq i \leq m,\ i+1 \leq j \leq k-2), \\ \mathbf{X}^{(m-1)}_{(i,j)} & (m+1 \leq i \leq L-1,\ m+1 \leq j \leq k-2), \\ \boldsymbol{\emptyset} & (2 \leq i \leq m,\ 1 \leq j \leq i-1), \\ \boldsymbol{\emptyset} & (m+1 \leq i \leq L-1,\ 1 \leq j \leq m-1). \end{cases}$$

Remark 2. $\mathbf{X}^{(h-1)}_{(i,j)}$ is denoted by

$$\mathbf{X}^{(h-1)}_{(i,j)} = \bigoplus_{v=h}^{j} \bigoplus_{\lambda_0=h}^{v} \bigoplus_{\lambda_1=h-1}^{\lambda_0-1} \bigoplus_{\lambda_2=h-2}^{\lambda_1-1} \cdots \bigoplus_{\lambda_{h-3}=3}^{\lambda_{h-4}-1} \begin{pmatrix} \mathbf{L}_{\delta_h} \\ \oplus \mathbf{L}_{\delta_h-(t_{h-1}-t_i)} \\ \oplus \mathbf{L}_{\delta_h-(\lambda_{h-3}-1)(t_{h-1}-t_i)} \\ \oplus \mathbf{L}_{\delta_h-(\lambda_{h-3}-2)(t_{h-1}-t_i)} \end{pmatrix},$$

where δ_h denotes the following term:

$$\delta_h = (j-v)t_0 + (v-\lambda_0)t_1 + \sum_{l=0}^{h-4} (\lambda_l - \lambda_{l+1} - 1)t_{l+2} + (\lambda_{h-3}-1)t_{h-1}.$$

From Lemma 4, the n_p-th row of $\mathbf{X}^{(h-1)}_{(i,j)}$ equals the XORed vector of the first row to the (n_p-1)-th row of $\mathbf{X}^{(h-1)}_{(i,j)}$.

Remark 3. If $L = k-1$ and $m = L-1$, $\mathbf{M}^{(L-1)}$ can be denoted as follows:

$$\begin{aligned} \mathbf{M}^{(L-1)} &= \mathbf{M}^{(k-2)} \\ &= \begin{pmatrix} \mathbf{H}_{(t_0,t_1)} & \mathbf{H}_{(2t_0,2t_1)} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_1)} & \mathbf{H}_{(-t_0,-t_1)} \\ \boldsymbol{\emptyset} & \mathbf{H}_{(2t_1,2t_2)} & \cdots & \mathbf{X}^{(1)}_{(2,k-2)} & \mathbf{H}_{(f_2(t_2),g_2(t_2))} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\emptyset} & \boldsymbol{\emptyset} & \cdots & \mathbf{H}_{(2t_{k-3},2t_{k-2})} & \mathbf{H}_{(f_{k-2}(t_{k-2}),g_{k-2}(t_{k-2}))} \end{pmatrix}. \end{aligned}$$

And also, if $L = k$ and $m = L - 1$, $\mathbf{M}^{(L-1)}$ can be denoted as follows:

$$\mathbf{M}^{(L-1)} = \mathbf{M}^{(k-1)}$$

$$= \begin{pmatrix} \mathbf{H}_{(t_0,t_1)} & \mathbf{H}_{(2t_0,2t_1)} & \cdots & \mathbf{H}_{((k-2)t_0,(k-2)t_1)} & \mathbf{H}_{(-t_0,-t_1)} \\ \textbf{Ø} & \mathbf{H}_{(2t_1,2t_2)} & \cdots & \mathbf{X}^{(1)}_{(2,k-2)} & \mathbf{H}_{(f_2(t_2),g_2(t_2))} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \textbf{Ø} & \textbf{Ø} & \cdots & \mathbf{H}_{(2t_{k-3},2t_{k-2})} & \mathbf{H}_{(f_{k-2}(t_{k-2}),g_{k-2}(t_{k-2}))} \\ \textbf{Ø} & \textbf{Ø} & \cdots & \textbf{Ø} & \mathbf{H}_{(f_{k-1}(t_{k-1}),g_{k-1}(t_{k-1}))} \end{pmatrix}.$$

Proof (proof sketch). The proof of this lemma can be derived by the elementary row operation on $\mathbf{M}^{(1)}$ with mathematical induction.

We can obtain $\mathbf{M}^{(2)}$ by the following elementary row operation on $\mathbf{M}^{(1)}$:

$$\mathbf{M}^{(2)} = \begin{pmatrix} & \mathbf{M}^{(1)}_1 & \\ \mathbf{F}^{(1)}_2 \cdot \mathbf{M}^{(1)}_1 & \oplus & \mathbf{G}^{(1)}_2 \cdot \mathbf{M}^{(1)}_2 \\ & \vdots & \\ \mathbf{F}^{(1)}_{L-1} \cdot \mathbf{M}^{(1)}_1 & \oplus & \mathbf{G}^{(1)}_{L-1} \cdot \mathbf{M}^{(1)}_{(L-1)} \end{pmatrix}.$$

where $\mathbf{F}^{(1)}_i$ and $\mathbf{G}^{(1)}_i$ $(2 \leq i \leq L-1)$ are defined as matrices to perform the elementary row operation, and which are defined by

$$\mathbf{F}^{(1)}_i = \bigoplus_{j=1}^{P^{(1)}_i} \mathbf{L}_{j(t_1-t_0)}, \quad \mathbf{G}^{(1)}_i = \bigoplus_{j=1}^{Q^{(1)}_i} \mathbf{L}_{j(t_i-t_0)},$$

respectively. $P^{(1)}_i$ and $Q^{(1)}_i$ are the minimum conting numbers such that

$$P_i(t_1 - t_0) + t_1 \equiv t_i \pmod{n_p},$$
$$Q_i(t_i - t_0) + t_i \equiv t_1 \pmod{n_p},$$

respectively.

Also, by using mathematical induction for $m = 3, \ldots, \alpha, \alpha + 1$, it is shown that $\mathbf{M}^{(m)}$ can be derived by the following elementary row operation on $\mathbf{M}^{(m-1)}$ for $m = 3, \ldots, L-1$:

$$\mathbf{M}^{(m)} = \begin{pmatrix} \mathbf{M}^{(m)}_1 \\ \vdots \\ \mathbf{M}^{(m)}_{m-1} \\ \hline \mathbf{M}^{(m)}_m \\ \vdots \\ \mathbf{M}^{(m)}_{L-1} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^{(1)}_1 \\ \vdots \\ \mathbf{M}^{(m-1)}_{m-1} \\ \hline \mathbf{F}^{(m-1)}_m \cdot \mathbf{M}^{(m-1)}_{m-1} \oplus \mathbf{G}^{(m-1)}_m \cdot \mathbf{M}^{(m-1)}_m \\ \vdots \\ \mathbf{F}^{(m-1)}_{L-1} \cdot \mathbf{M}^{(m-1)}_{m-1} \oplus \mathbf{G}^{(m-1)}_{L-1} \cdot \mathbf{M}^{(m-1)}_{L-1} \end{pmatrix},$$

where $\mathbf{F}^{(m-1)}_i$ and $\mathbf{G}^{(m-1)}_i$ $(m \leq i \leq L-1)$ are defined as matrices to perform the elementary row operation, and which are defined by

$$\mathbf{F}^{(m-1)}_i = \bigoplus_{j=1}^{P^{(m-1)}_i} \mathbf{L}_{2j(t_{m-1}-t_{m-2})-t_{m-1}}, \quad \mathbf{G}^{(m-1)}_i = \bigoplus_{j=1}^{Q^{(m-1)}_i} \mathbf{L}_{2j(t_i-t_{m-2})-t_i},$$

where $P_i^{(m-1)}$ and $Q_i^{(m-1)}$ are the minimum counting numbers such that

$$2P_i^{(m-1)}(t_{m-1} - t_{m-2}) + t_{m-1} \equiv t_i \pmod{n_p},$$
$$2Q_i^{(m-1)}(t_i - t_{m-2}) + t_i \equiv t_{m-1} \pmod{n_p},$$

respectively. Therefore, Lemma 2 is proved.

Also, in the case where $m = k-1$ if $L \geq k$, $\mathbf{M}^{(k-1)}$ (if $L = k$) can be denoted by

$$\mathbf{M}^{(k-1)} = \left(\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{M}_1^{(1)}} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathbf{M}_{k-2}^{(k-2)}} \\ \hline & \mathbf{H}_{(f_{k-1}(t_{k-1}), g_{k-1}(t_{k-1}))} \\ \mathbf{0} & \vdots \\ & \mathbf{H}_{(f_{k-1}(t_{L-1}), g_{k-1}(t_{L-1}))} \end{array}\right),$$

where $\mathbf{0}$ denotes $(L-k+1)n_p \times (k-2)n_p$ null matrix. Then, from Lemma 3, $\mathbf{M}^{(k-1)}$ can be transformed by the elementary row operation as follows:

$$\mathbf{M}^{(k-1)} = \left(\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{M}_1^{(1)}} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathbf{M}_{k-2}^{(k-2)}} \\ \hline & \mathbf{H}_{(f_{k-1}(t_{k-1}), g_{k-1}(t_{k-1}))} \\ & \emptyset \\ \mathbf{0} & \vdots \\ & \emptyset \end{array}\right).$$

□

If we present a detailed description of this proof, it is not difficult to understand but it is too long to be described in full in this paper. Moreover, the detailed proof can be easily derived from the definitions of the elementary row operation denoted in the above proof sketch. Thus, we have omitted a detailed proof here.

Appendix 3 Lemma 3

Lemma 3. *Let $\mathbf{H}_{(i,j)}$ denote the following $n_p \times n_p$ matrix:*

$$\mathbf{H}_{(i,j)} = \mathbf{L}_i \oplus \mathbf{L}_j,$$

where $i, j \in GF(n_p)$, $i \neq j$, and $\mathbf{L}_i$ denotes the following rotated identity matrix:

$$\mathbf{L}_i = \begin{pmatrix} \emptyset & \mathbf{I}_{n_p-i} \\ \mathbf{I}_i & \emptyset \end{pmatrix}.$$

Then, it is possible to make an arbitrary vector from the XOR combination of rows of $\mathbf{H}_{(i,j)}$, whose hamming weight is an even number.

Proof. We suppose that $\mathbf{h}_{(i,j)}^{(l)}$ denotes the l-th row of $\mathbf{H}_{(i,j)}$, which can be denoted as follows:

$$\mathbf{h}_{(i,j)}^{(l)} = \mathbf{i}_{i+l} \oplus \mathbf{i}_{j+l},$$

where $l \in GF(n_p)$.Let $\mathbf{v}$ denote the arbitrary vector whose hamming weight is two. Then, $\mathbf{v}$ can be denoted as follows:

$$\mathbf{v} = \mathbf{i}_{\alpha} \oplus \mathbf{i}_{\beta},$$

where $\alpha, \beta \in GF(n_p)$ and $\alpha \neq \beta$. Then, we define p' by the following equation on $GF(n_p)$:

$$p' = (\beta - \alpha)/(j-i) - 1 \pmod{n_p}.$$

Since the indexes are elements of $GF(n_p)$, the following equation is satisfied:

$$\begin{aligned}\bigoplus_{p=0}^{p'} \mathbf{h}_{(i,j)}^{(\alpha-i+p(j-i))} &= \bigoplus_{p=0}^{p'} \left(\mathbf{i}_{\alpha+p(j-i)} \oplus \mathbf{i}_{\alpha+(p+1)(j-i)}\right)\\ &= \mathbf{i}_{\alpha} \oplus \mathbf{i}_{\alpha+(p'+1)(j-i)}\\ &= \mathbf{i}_{\alpha} \oplus \mathbf{i}_{\beta} = \mathbf{v}.\end{aligned}$$

Thus, it is possible that the arbitrary vector whose hamming weight is two can be generated by the XOR combination of rows of $\mathbf{H}_{(i,j)}$.

On the other hand, we suppose that $\mathbf{u}$ denotes the vector whose hamming weight is an even number greater than two and $||\mathbf{u}||$ denotes the hamming weight of $\mathbf{u}$. Then, $||\mathbf{u}||$ can be denoted by $||\mathbf{u}|| = 2c$, where c is a counting number more than two. Thus, $\mathbf{u}$ can be denoted by the linear combination of vectors whose hamming weight is two. Therefore, an arbitrary vector whose hamming weight is an even number can be generated from the XOR combination of rows of $\mathbf{H}_{(i,j)}$. □

Appendix 4 Lemma 4

Lemma 4. *Suppose n_p is a prime number. Let $\mathcal{X}$ be a set of n_p-dimensional binary vectors defined by*

$$\mathcal{X} = \left\{\mathbf{i}_{i+m} \oplus \mathbf{i}_{j+m} \mid 0 \leq m \leq n_p - 2\right\},$$

where $i, j \in GF(n_p)$, $i \neq j$ and $\mathbf{i}_l$ denotes an n_p-dimensional vector such that

$$\begin{aligned}\mathbf{i}_0 &= (1\ 0\ 0\ \dots\ 0\ 0),\\ \mathbf{i}_1 &= (0\ 1\ 0\ \dots\ 0\ 0),\\ &\vdots\\ \mathbf{i}_{n_p-1} &= (0\ 0\ 0\ \dots\ 0\ 1).\end{aligned}$$

Then, all vectors in $\mathcal{X}$ are linearly independent.

Proof. Let $\mathcal{X}'$ be the set defined by

$$\mathcal{X}' = \{\mathbf{i}_{i+l} \oplus \mathbf{i}_{j+l} \mid 0 \leq l \leq n_p - 1\}.$$

We define α by $\alpha = j - i \pmod{n_p}$. Since n_p is a prime number, from Lemma 5, $\mathcal{X}'$ can be also denoted as follows:

$$\mathcal{X}' = \{\mathbf{i}_{i+l\alpha} \oplus \mathbf{i}_{i+(l+1)\alpha} \mid 0 \leq l \leq n_p - 1\}.$$

We suppose that $\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha}$ $(q \in GF(n_p))$ is an arbitrary element of $\mathcal{X}'$. Then, since indexes are elements of $GF(n_p)$, we can make the XORed vector identical to $\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha}$ if and only if we can use all the elements of $\mathcal{X}' \backslash \{\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha}\}$. That is, we can make $\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha}$ only by the following operation:

$$\begin{aligned}
\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha} &= \bigoplus_{\substack{l=0,\\ l\neq q}}^{n_p-1} \left\{\mathbf{i}_{i+l\alpha} \oplus \mathbf{i}_{i+(l+1)\alpha}\right\} \\
&= (\mathbf{i}_{i+q\alpha} \oplus \mathbf{i}_{i+(q+1)\alpha}) \oplus \bigoplus_{l=0}^{n_p-1} \left\{\mathbf{i}_{i+l\alpha} \oplus \mathbf{i}_{i+(l+1)\alpha}\right\}.
\end{aligned}$$

Therefore, since $\mathcal{X}$ can be denoted by

$$\mathcal{X} = \mathcal{X}' \backslash \{\mathbf{i}_{i-1} \oplus \mathbf{i}_{j-1}\},$$

it is impossible to make the XORed vector identical to $\mathbf{i}_{i+m} \oplus \mathbf{i}_{j+m}$ from the elements of $\mathcal{X} \backslash \{\mathbf{i}_{i+m} \oplus \mathbf{i}_{j+m}\}$. Thus, all vectors in $\mathcal{X}$ are linearly independent. □

Appendix 5 Lemma 5

This lemma indicates just a fact about arithmetic operations on $GF(n_p)$. Thus, we omit the poof of this lemma.

Lemma 5. *Suppose x and y are arbitrary elements such that $x \in GF(n_p) \backslash \{0\}$, $y \in GF(n_p)$. Then, the following equation is satisfied:*

$$\begin{aligned}
\{0, 1, 2, \ldots, n_p - 1\} &= \{y,\ y + x,\ y + 2x, \ldots,\ y + (n_p - 1)x\} \pmod{n_p} \\
&= GF(n_p).
\end{aligned}$$

Appendix 6 Lemma 6

Same lemma and its proof are provided in [11]. Thus, we omit the poof of this lemma.

Lemma 6. *Suppose that* $x_0, \ldots, x_{L-1}$ *(*$L \in \mathbb{N}$*,* $L \geq 2$*) are random numbers which are chosen from the finite set* $\{0,1\}^h$ *(*$h > 0$*) independently from each other with uniform probability* $1/2^h$*. Let* $\mathcal{X}$ *be the set defined by* $\mathcal{X} = \{x_0, \ldots, x_{L-1}\}$*.*

Then, $x_0, \ldots, x_{L-1}$ *and all the XORed combinations of the elements in* $\mathcal{X}$ *are random numbers which are pairwise independent and uniformly distributed over* $\{0,1\}^h$*.*