Improvement of a Secret Sharing Scheme to Reduce the Total Data Size

Tatsuki Ishii[1], Kuniaki Tsuji[1], Yuya Tarutani[2], Yukinobu Fukushima[3], and Tokumi Yokohira[2]

1: Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University

2: Faculty of Interdisciplinary Sience and Engineering in Health Systems, Okayama University

3: Faculty of Enviroment, Life, Natural Sience and Technology, Okayama University

3-1-1, Tsushima-Naka, Okayama-city, Okayama, 700-8530, Japan

Email: {y-tarutn, fukusima, yokohira}@okayama-u.ac.jp

***Abstract*— In recent years, a variety of information has been converted into electronic data including highly private information. To protect such secret data from loss or theft, secret sharing schemes have been proposed. In the schemes, for given integers n and k, n data blocks called shares are generated from the secret data so that following conditions (i) and (ii) are satisfied. (i) The secret data is constructed from k or more shares. (ii) We can not obtain any information regarding the secret data from $k-1$ or less shares. We have proposed a secret sharing scheme to reduce the total data size of all shares. In this paper, we improve the scheme to further reduce the total data size using such a property of the EXOR operation that arbitrary logic variable a_i among $n-1$ logic variables a_0, a_1, $\cdots$, a_{n-2} is obtained from a_0, a_1, $\cdots$, a_{i-1}, a_{i+1}, $\cdots$, a_{n-2} and the EXOR operation $(a_0 \oplus a_1 \oplus \cdots \oplus a_{n-2})$ of the $n-1$ logic variables. The improved scheme can reduce the total data size by about 33% at most compared to our previous scheme.**

I. INTRODUCTION

With the development and spread of information technologies, various types of information such as electronic medical records and online banking data have been converted into electronic data. The main scheme to prevent their leakage is to encrypt them [1]-[4]. However if the secret key for encryption is stolen or lost due to disaster, the encryption becomes meaningless.

Secret sharing schemes [5]-[8] have been proposed to protect secret keys and other confidential data such as highly sensitive files from loss or theft. In the schemes, given integers n and k, n blocks called shares are generated from the secret data, and each share is saved in a different location. We can construct the secret data by collecting k or more shares, and we can not construct the secret data even if we collect $k-1$ or less shares and there are no information leakage from the $k-1$ or less shares. Since the computational cost of the schemes of papers [5] and [6] is large, they are only applicable to small size data such as secret keys. On the other hand, paper [7] can be applied to larger data sizes because the computational cost is greatly reduced by using EXOR operation. However, since the size of each share is the same as the size of the secret data (say L Bytes) in the three schemes, the total data size of shares is nL. In order to reduce the total data size, we have proposed a secret sharing scheme [8] which can reduce the total data size to $2/n$.

In this paper, we propose a new secret sharing scheme to further reduce the total amount of data. In our previous scheme [8], the secret data is partitioned into n blocks of equal size B_0, B_1, $\cdots$, B_{n-1} and $n-k+1$ blocks B_i, B_{i-1}, $\cdots$, B_{i-n+k} ($+$ and $-$ are mod n operation) consist of share i $(0 \le i \le n-1)$ after the $n-k+1$ blocks are encrypted. In our proposed scheme, we assume $k \le n-1$ (when $k=n$, we use our previous scheme). The secret data is partitioned into $n-1$ blocks of equal size B_0, B_1, $\cdots$, B_{n-2} and B_{n-1} is constructed as their EXOR $(B_0 \oplus B_1 \oplus \cdots \oplus B_{n-2})$, and $n-k$ blocks B_i, B_{i-1}, $\cdots$, $B_{i-n+k+1}$ consist of share i after the $n-k$ blocks are encrypted. Although each block B_i in our new scheme is a little bit larger than each block B_i in our previous scheme, the number of blocks in each share in our proposed scheme is smaller by one, and consequently total number of blocks in our proposed scheme is smaller by n. Therefore, we can reduce the total data size for most values of n and k.

The rest of the paper is organized as follows, Section II describes the scheme of paper [8]. Section III presents our proposed scheme. Section IV discusses the total data size, and finally Section V concludes the paper.

II. PREVIOUS SCHEME TO REDUCE THE TOTAL DATA SIZE

Our previous scheme constructs n shares as follows for given integers n and k.

(1) Partition the secret data into n blocks of equal size B_0, B_1, $\cdots$, B_{n-1} as show in Fig. 1.

B_0	B_1	$\cdots$	B_{n-2}	B_{n-1}

Fig. 1. Block Partitioning in Our Previous Scheme

(2) Create TABLE I, where row 1 is B_0, B_1, $\cdots$, B_{n-1}, and row j $(2 \le j \le n-k+1)$ is obtained by rotating row 1 $j-1$ times.We call the set of B_i in column i *block set* BS_i.

(3) Generate a random value R whose number of bits is equal to each B_i, and generate value X_i $(0 \le i \le n-1)$ which is a share of R (R can be constructed by collecting k or more X_is and cannot be constructed even if we collect $k-1$ or less X_is) using the secret sharing scheme described in paper [7] (note that the number of bits in each X_i is also equal to each B_i).

TABLE I. Block Sets in Our Previous Scheme

	BS_0	BS_1	$\cdots$	BS_{i-1}	BS_i	BS_{i+1}	$\cdots$	BS_{n-2}	BS_{n-1}
row 1	B_0	B_1	$\cdots$	B_{i-1}	B_i	B_{i+1}	$\cdots$	B_{n-2}	B_{n-1}
row 2	B_{n-1}	B_2	$\cdots$	B_{i-2}	B_{i-1}	B_i	$\cdots$	B_{n-2}	B_{n-2}
	B_{n-2}	B_{n-1}	$\cdots$	B_{i-3}	B_{i-2}	B_{i-1}	$\cdots$	B_{n-4}	B_{n-3}
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
row $n-k+1$	B_k	B_{k+1}	$\cdots$	$B_{i-n+k-1}$	B_{i-n+k}	$B_{i-n+k+1}$	$\cdots$	B_{k-2}	B_{k-1}

TABLE II. Encrypted Block Sets in Our Previous Scheme

	EBS_0	EBS_1	$\cdots$	EBS_{i-1}	EBS_i	EBS_{i+1}	$\cdots$	EBS_{n-2}	EBS_{n-1}
row 1	$B_0 \oplus R$	$B_1 \oplus R$	$\cdots$	$B_{i-1} \oplus R$	$B_i \oplus R$	$B_{i+1} \oplus R$	$\cdots$	$B_{n-2} \oplus R$	$B_{n-1} \oplus R$
row 2	$B_{n-1} \oplus R$	$B_0 \oplus R$	$\cdots$	$B_{i-2} \oplus R$	$B_{i-1} \oplus R$	$B_i \oplus R$	$\cdots$	$B_{n-3} \oplus R$	$B_{n-2} \oplus R$
row 3	$B_{n-2} \oplus R$	$B_{n-1} \oplus R$	$\cdots$	$B_{i-3} \oplus R$	$B_{i-2} \oplus R$	$B_{i-1} \oplus R$	$\cdots$	$B_{n-4} \oplus R$	$B_{n-3} \oplus R$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
row $n-k+1$	$B_k \oplus R$	$B_{k+1} \oplus R$	$\cdots$	$B_{i-n+k-1} \oplus R$	$B_{i-n+k} \oplus R$	$B_{i-n+k+1} \oplus R$	$\cdots$	$B_{k-2} \oplus R$	$B_{k-1} \oplus R$
	X_0	X_1	$\cdots$	X_{i-1}	X_i	X_{i+1}	$\cdots$	X_{n-2}	X_{n-1}

(4) Execute the EXOR operation of R and each block B_i $(0 \le i \le n-1)$ as shown in TABLE II (note that because each B_i is a plain text, we encrypt each B_i using the EXOR operation), and include each X_i into BS_i. We call such created BS_i *encrypted* BS_i (EBS_i).

(5) Consider each EBS_i as share i.

Because we can collect all B_is $(0 \le i \le n-1)$ by collecting arbitrary k block sets (see paper [8] for the proof), we can collect all $B_i \oplus R$s $(0 \le i \le n-1)$ by collecting k encrypted block sets. On the other hand, we can obtain R by collecting arbitrary k encrypted block sets because the k encrypted block sets includes k X_is. Thus we can collect all B_is by collecting arbitrary k encrypted block sets.

TABLEs III and IV show block sets and encrypted block sets for $n = 5$ and $k = 3$. We can confirm that arbitrary three encrypted block sets have all $B_i \oplus R$s, that is, $B_1 \oplus R$, $B_2 \oplus R$, $B_3 \oplus R$, $B_4 \oplus R$ and $B_5 \oplus R$.

TABLE III. Block Sets in Our Previous Scheme ($n = 5, k = 3$)

BS_0	BS_1	BS_2	BS_3	BS_4
B_0	B_1	B_2	B_3	B_4
B_4	B_0	B_1	B_2	B_3
B_3	B_4	B_0	B_1	B_2

TABLE IV. Encrypted Block Sets in Our Previous Scheme ($n = 5, k = 3$)

EBS_0	EBS_1	EBS_2	EBS_3	EBS_4
$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$	$B_4 \oplus R$
$B_4 \oplus R$	$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$
$B_3 \oplus R$	$B_4 \oplus R$	$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$
X_0	X_1	X_2	X_3	X_4

III. Proposed Scheme

In our proposed scheme, the secret data is partitioned into $n-1$ blocks of equal size, B_0, B_1, $\cdots$, B_{n-2} and B_{n-1} is constructed as their EXOR $(B_0 \oplus B_1 \oplus \cdots \oplus B_{n-2})$, and $n-k$ blocks B_i, B_{i-1}, $\cdots$, $B_{i-n+k+1}$ consist of share i after the $n-k$ blocks are encrypted. Although each block B_i in our new scheme is a little bit larger than each block B_i in our previous scheme, the number of blocks in each share in our proposed scheme is smaller by one, and consequently total number of blocks in our proposed scheme is smaller by n. Therefore, we can reduce the total data size for most values of n and k. The detailed steps to create shares is as follows.

(1) Partition the secret data into $n-1$ blocks of equal size, $B_0, B_1, \cdots, B_{n-2}$ as shown in Fig. 2.

B_0	B_1	$\cdots$	B_{n-3}	B_{n-2}

Fig. 2. Block Partitioning in Our Proposed Scheme

(2) Create TABLE V, where B_{n-1} is the EXOR of B_0, B_1, $\cdots$, B_{n-2} and row 1 is B_0, B_1, $\cdots$, B_{n-2}, B_{n-1} and row j is obtained by rotating row 1 $j-1$ times.

(3) Generate a random value R whose number of bits is equal to each B_i, and generate value X_i which is a share of R using the scheme described in paper [7].

(4) Perform the EXOR operation of each B_is $(0 \le i \le n-2)$ and R as shown in TABLE VI, and if the number of operands in B_{n-1}, which is $n-1$, is an odd number, then perform the EXOR operation of B_{n-1} and R (note that TABLE VI corresponds to the case of an odd number and we do not perform the EXOR operation of B_{n-1} and R if $n-1$ is an even number), and include X_i $(0 \le i \le n-1)$ into EBS_i.

(5) Consider each EBS_i as share i

The TABLEs VII and VIII show block sets and encrypted block sets respectively created by the proposed scheme for the $n = 5, k = 3$. In the same way, TABLE IX and X show block sets and encrypted block sets respectively for $n = 6, k = 3$. As shown in TABLEs VIII and X, when $n = 5$ $(n = 6)$, because $n - 1$ is an even (odd) number, we do not (do) perform the EXOR operation of B_{n-1} and R. Later we describe the reason why we do not (do) perform the EXOR operation of B_{n-1} and R when $n - 1$ is an even (odd) number.

TABLE VII. Block Sets in Our Proposed Scheme ($n = 5, k = 3$)

BS_0	BS_1	BS_2	BS_3	BS_4
B_0	B_1	B_2	B_3	B_4
B_4	B_0	B_1	B_2	B_3

TABLE VIII. Encrypted Block Sets in Our Proposed Scheme ($n = 5, k = 3$)

EBS_0	EBS_1	EBS_2	EBS_3	EBS_4
$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$	B_4
B_4	$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$
X_0	X_1	X_2	X_3	X_4

TABLE IX. Block Sets in Our Proposed Scheme $(n = 6, k = 3)$

BS_0	BS_1	BS_2	BS_3	BS_4	BS_5
B_0	B_1	B_2	B_3	B_4	B_5
B_5	B_0	B_1	B_2	B_3	B_4
B_4	B_5	B_0	B_1	B_2	B_3

TABLE X. Encrypted Block Sets in Our Proposed Scheme $(n = 6, k = 3)$

EBS_0	EBS_1	EBS_2	EBS_3	EBS_4	EBS_5
$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$	$B_4 \oplus R$	$B_5 \oplus R$
$B_5 \oplus R$	$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$	$B_4 \oplus R$
$B_4 \oplus R$	$B_5 \oplus R$	$B_0 \oplus R$	$B_1 \oplus R$	$B_2 \oplus R$	$B_3 \oplus R$
X_0	X_1	X_2	X_3	X_4	X_5

When we collect all B_is $(0 \le i \le n - 2)$, it is trivial that we can construct the secret data. So, we prove that we can collect all B_is by collecting arbitrary k EBS_is after proving the following lemma.

[Lemma] We can collect $n - 1$ B_is among n B_is by collecting arbitrary k BS_is. ∎

[Proof of the lemma] Assuming that k BS_is, BS_{i_0}, BS_{i_1}, $\cdots$, $BS_{i_{k-1}}$ among n BS_is are collected, we prove by dividing the following two cases.

(Case 1) Case that for every value of m $(0 \le m \le k - 1)$, $i_{m+1} - i_m$ (+ is mod k addition) is less than or equal to $n - k - 1$:

In TABLE XI, the collected BS_is are hatched in black. We first focus on columns between BS_{i_0} and BS_{i_1}. In row 1, blocks from B_{i_0+1} to B_{i_1-1} are not included in the hatched area corresponding to BS_is. However, in row 2, B_{i_1-1} is included in the hatched area because row 2 is obtained by rotating row 1 one time. Generally speaking, in row $i_1 - i_0 - j + 1$ $(1 \le j \le i_1 - i_0 - 1)$, B_{i_0+j} is included in the hatched area. Thus, in row $i_1 - i_0$, B_{i_0+1} is included in the hatched area. On the other hand, because $i_1 - i_0 < n - k$ from the assumption, row 2 to $i_1 - i_0$ are located above row $n - k$. Thus, blocks B_{i_0+1} to B_{i_1-1} are included in the hatch area. The discussion for $m = 0$ holds for the other values of m. Therefore, every block which is not included in the hatched area in row 1 is included in the hatched area in another row above row $n - k$.

(Case 2) Case that for an value m $(0 \le m \le k - 1)$, $i_{m+1} - i_m$ (+ is mod k addition) is equal to $n - k$:

Without loss of generality, we assume that $i_1 - i_0$ is equal to $n - k$. Note that because we collected k BS_is among n BS_is, BS_0 to BS_{i_0} and BS_{i_1} to BS_{n-1} are all collected and BS_{i_0+1} to BS_{i_1-1} are not collected. In TABLE XII, we show this situation. The hatched area in black means the collected BS_is, and the non-hatched area means non-collected BS_is. As described in Case 1, B_{i_0+j} is included in the hatched area in row $i_1 - i_0 - j + 1$. Therefore, each of B_{i_0+1} to B_{i_1-1} which are not included in the hatched area in row 1 is included in the hatched area in another row as shown in TABLE XII. On the other hand, only B_{i_0+1} is not included in the hatched area in any row as shown in TABLE XII. ∎

Using the lemma describe above, we prove that we can collect all $B_i s$ $(0 \le i \le n - 2)$ by collecting arbitrary k EBS_is as follows.

We first prove when $n - 1$ is an odd number. Because TABLE V and VI are the same structure except the last row where X_is are included in TABLE VI, we can collect $n - 1$ encrypted block $B_i \oplus R$s among n encrypted block $B_i \oplus R$s $(0 \le i \le n - 1)$ from the lemma. If the collected $n - 1$ encrypted blocks are $B_0 \oplus R$, $B_1 \oplus R$, $\cdots$, $B_{n-2} \oplus R$ (that is $B_{n-1} \oplus R$ is not included), the we can collect B_0, B_1, $\cdots$, B_{n-2} because R is calculated from X_is which are included in the collected EBS_is. If the collected $n - 1$ encrypted blocks do not included $B_i \oplus R$, that is, $B_0 \oplus R$, $B_1 \oplus R$, $\cdots$, $B_{i-1} \oplus R$, $B_{i+1} \oplus R$, $\cdots$, $B_{n-2} \oplus R$, $B_{n-1} \oplus R$ are included, we can collect B_0, B_1, $\cdots$, B_{i-1}, B_{i+1}, $\cdots$, B_{n-2}, B_{n-1} because R is calculated from X_is which are included in the collected EBS_is. From the definition of B_{n-1}, $B_{n-1} = B_0 \oplus B_1 \oplus \cdots B_{i-1} \oplus B_i \oplus B_{i+1} \oplus \cdots \oplus B_{n-2}$. Thus, we can obtain B_i by performing the EXOR operation of B_0, B_1, $\cdots$, B_{i-1}, B_{i+1}, $\cdots$, B_{n-2} and B_{n-1}. Thus we can collect all $n - 1$ B_is $(0 \le i \le n - 2)$.

Next we prove when $n - 1$ is an even number. When $n - 1$ is an even number, we do not the EXOR operation of B_{n-1} and R as described above. In the proof for an odd number of $n - 1$ described above, if we change $B_{n-1} \oplus R$ to B_{n-1}, the same proof holds for an even number of $n - 1$.

We explain the reason why we do not perform the EXOR operation of B_{n-1} and R when $n - 1$ is an even number. In TABLE VIII, assume that we put $B_4 \oplus R$ instead of B_4. When we collect EBS_1 and EBS_4, we obtain $B_0 \oplus R$, $B_1 \oplus R$, $B_3 \oplus R$ and $B_4 \oplus R$ $(= B_0 \oplus B_1 \oplus B_2 \oplus B_3 \oplus R)$. When we perform the EXOR operation of the four EXOR

expressions, B_2 itself (which is a plain text) is obtained, which means we obtain a part of the secret data by collecting two $(k-1)$ shares, and consequently it does not satisfy a necessary condition of secret sharing schemes. In order to avoid such situation, we make every EXOR expression have an even number of operands because the EXOR operation of EXOR expressions with even numbers of operands makes an EXOR expression with an even number of operands, which leads to avoidance of exposing a part of the secret data. Thus, when $n-1$ (the number of operands in B_{n-1}) is an even number, we do not perform the EXOR operation of B_{n-1} and R.

TABLE V. Block Sets in Our Proposed Scheme

	BS_0	BS_1	$\cdots$	BS_{i-1}	BS_i	BS_{i+1}	$\cdots$	BS_{n-2}	BS_{n-1}
row 1	B_0	B_1	$\cdots$	B_{i-1}	B_i	B_{i+1}	$\cdots$	B_{n-2}	B_{n-1}
row 2	B_{n-1}	B_0	$\cdots$	B_{i-2}	B_{i-1}	B_i	$\cdots$	B_{n-3}	B_{n-2}
row 3	B_{n-2}	B_{n-1}	$\cdots$	B_{i-3}	B_{i-2}	B_{i-1}	$\cdots$	B_{n-4}	B_{n-3}
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
row $n-k$	B_{k+1}	B_{k+2}	$\cdots$	B_{i-n+k}	$B_{i-n+k+1}$	$B_{i-n+k+2}$	$\cdots$	B_{k-1}	B_k

TABLE VI. Encrypted Block Sets in Our Proposed Scheme

	EBS_0	EBS_1	$\cdots$	EBS_{i-1}	EBS_i	EBS_{i+1}	$\cdots$	EBS_{n-2}	EBS_{n-1}
row 1	$B_0 \oplus R$	$B_1 \oplus R$	$\cdots$	$B_{i-1} \oplus R$	$B_i \oplus R$	$B_{i+1} \oplus R$	$\cdots$	$B_{n-2} \oplus R$	$B_{n-1} \oplus R$
row 2	$B_{n-1} \oplus R$	$B_0 \oplus R$	$\cdots$	$B_{i-2} \oplus R$	$B_{i-1} \oplus R$	$B_i \oplus R$	$\cdots$	$B_{n-3} \oplus R$	$B_{n-2} \oplus R$
row 3	$B_{n-2} \oplus R$	$B_{n-1} \oplus R$	$\cdots$	$B_{i-3} \oplus R$	$B_{i-2} \oplus R$	$B_{i-1} \oplus R$	$\cdots$	$B_{n-4} \oplus R$	$B_{n-3} \oplus R$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
row $n-k$	$B_{k+1} \oplus R$	$B_{k+2} \oplus R$	$\cdots$	$B_{i-n+k} \oplus R$	$B_{i-n+k+1} \oplus R$	$B_{i-n+k+2} \oplus R$	$\cdots$	$B_{k-1} \oplus R$	$B_k \oplus R$
	X_0	X_1	$\cdots$	X_{i-1}	X_i	X_{i+1}	$\cdots$	X_{n-2}	X_{n-1}

TABLE XI. Block Sets in Case 1

			$\leq n-k-1$ columns									
	$\cdots$	BS_{i_0}	BS_{i_0+1}	$\cdots$	BS_{i_0+j}	BS_{i_0+j+1}	$\cdots$	BS_{i_1}	$\cdots$	$BS_{i_{k-1}}$	$\cdots$
row 1	$\cdots$	B_{i_0}	B_{i_0+1}	$\cdots$	B_{i_0+j}	B_{i_0+j+1}	$\cdots$	B_{i_1}	$\cdots$	$B_{i_{k-1}}$	$\cdots$
row 2	$\cdots$	B_{i_0-1}	B_{i_0}	$\cdots$	B_{i_0+j-1}	B_{i_0+j}	$\cdots$	B_{i_1-1}	$\cdots$	$B_{i_{k-2}}$	$\cdots$
	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
row i_1-i_0-j+1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+j}	$\cdots$	$\cdots$	$\cdots$
	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
row i_1-i_0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+1}	$\cdots$	$\cdots$	$\cdots$
	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
row $n-k$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

TABLE XII. Block Sets in Case 2

$= n - k$ columns

	$\cdots$	BS_{i_0}	BS_{i_0+1}	BS_{i_0+2}	$\cdots$	BS_{i_0+j}	BS_{i_0+j+1}	$\cdots$	BS_{i_1-1}	BS_{i_1}	$\cdots$
row 1	$\cdots$	B_{i_0}	B_{i_0+1}	B_{i_0+2}	$\cdots$	B_{i_0+j}	B_{i_0+j+1}	$\cdots$	B_{i_1-1}	B_{i_1}	$\cdots$
row 2	$\cdots$	B_{i_0-1}	B_{i_0}	B_{i_0+1}	$\cdots$	B_{i_0+j-1}	B_{i_0+j}	$\cdots$	B_{i_1-2}	B_{i_1-1}	$\cdots$
	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$
row $i_1 - i_0 - j + 1$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+j}	$\cdots$
	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$
row $n-k-2$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+4}	$\cdots$
row $n-k-1$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+3}	$\cdots$
row $n-k$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	B_{i_0+1}	B_{i_0+2}	$\cdots$

IV. Total Data Size in the Proposed Scheme

The total data size in our previous scheme is calculated as follows. The size of each block is L/n, each block set has $n-k+1$ blocks and X_i whose size is L/n, and the number of block sets is n. So, the total data size is calculated as follows.

$$\left\{\frac{L}{n}(n-k+1)+\frac{L}{n}\right\}n=(n-k+2)L$$

On the other hand, the total data size in our proposed scheme is calculated as follows. The size of each block is $L/(n-1)$, each block set has $n-k$ blocks and X_i whose size is $L/(n-1)$, and the number of block sets is n. So, the total data size is calculated as follows.

$$\left\{\frac{L}{n-1}(n-k)+\frac{L}{n-1}\right\}n=\frac{nL(n-k+1)}{n-1}$$

Thus, the ratio of the total data size in our proposed scheme to that in our previous scheme is calculated as follows.

$$\frac{\frac{nL(n-k+1)}{n-1}}{(n-k+2)L}=\frac{n(n-k+1)}{(n-1)(n-k+2)}$$

Fig. 3 shows the ratios for $n=5$, 10 and 20. When $k=1$, the ratio is equal to $n^2/(n^2-1)$ and consequently the total data size in our proposed scheme is a little bit larger than that in our previous scheme. When $k=2$, the ratio is equal to one, which means the total data sizes in the both schemes are the same. When $k\geq 3$, the ratio is smaller than one, which means the total data size in our proposed scheme is smaller than that in our previous scheme. When the value of n is fixed, the ratio becomes smaller for larger value of k and the maximum value (when $k=n-1$) of the ratio is $2n/3(n-1)$, and consequently the ratio is about $2/3$ for large value of n.

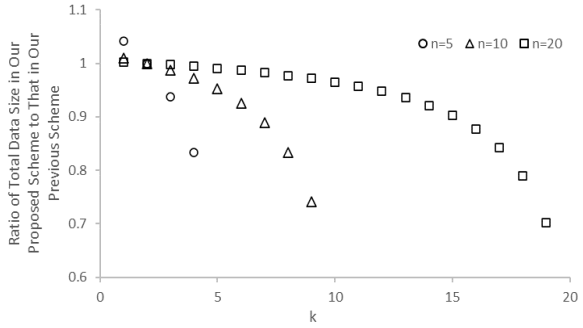

Fig. 3. Ratio of Total Data Size in Our Proposed Scheme to That in Our Previous Scheme

V. Conclusion

In this paper, we have proposed a secret sharing scheme that reduces the total data size in our previous scheme. For given integers n and k (n is the number of shares and k is a necessary number of shares to construct the secret data), we show that (i) when $k=1$, the total data size in our proposed scheme is a little bit larger than that in our previous scheme, (ii) when $k=2$, the total data sizes in the both schemes are the same, and (iii) when $k\geq 3$, the total data size in our proposed scheme is smaller than that in our previous scheme, and when the value of n is fixed, the ratio of the total data size in our proposed scheme to that in our previous scheme becomes smaller for larger value of k and the maximum value of the ratio is $2n/3(n-1)$, which is about $2/3$ for large value of n. From (i) ~ (iii), we can conclude that our propose scheme can reduce the total data size for most values of n and k, and the reduction is a maximum of about 33%.

Our future work is as follows. When some of nodes where shares are located can not be used due to failures or natural disaster, locations of shares have a big impact on the possibility of constructing the secret data. Thus, our future work is to consider how to locate shares in the Internet to increase the possibility.

References

[1] National Institute of Standards and Technology, "Announcing the Advanced Encryption Standard (AES)", FIPS Publication 197, 2001.

[2] R. L. Rivest, A. Shamir, L. Adelman, "A Method for Obtaining Digital Signature and Public-key Cryptosystems", Communications of the ACM, pp 120126, Issue 2, Volume 21, Feb. 1978.

[3] V. S. Miller, "Use of elliptic curves in cryptography", CRYPTO '85 (LNCS 218), pp. 417-426, 1986.

[4] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, Vol. 4X, No. 177, pp. 203-209, 1987.

[5] A Shamir, "How to Share a Secret", Communication of the ACM, vol.22, No11, pp.612-613, 1979.

[6] G. Blakley, "Safeguarding cryptographic keys", Proc. of AFIPS, 48, pp. 313-317, 1979.

[7] Yoshihiro FUJII, Kyouka TOCHIKUBO, Norikazu HOSAKA, Minako TADA, and Takehisa KATO, "(k, n) Threshold Schemes Using XOR Operations", THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGIEERS, IEICE Technical Report ISE2007-5 (in Japanese).

[8] Kuniaki Tsuji, Shiden Kishimoto, Yuya Tarutani, Yukinobu Fukushima, and Tokumi Yokohira "A Secret Sharing Scheme to Reduce the Total Data Size", ICTC 2021