XORを用いた高速な秘密分散法のデータ容量削減に関する一手法

永井良英† 高荒亮† 岩村惠市†

†東京理科大学
102-0073 東京都千代田区九段北 1-14-6
{nagai,takaara}@sec.ee.kagu.tus.ac.jp,iwamura@ee.kagu.tus.ac.jp

XORを用いた高速な秘密分散法が提案されている．この手法は、Shamirの秘密分散法が秘密情報の分散・復元時に$k-1$次の多項式を処理するため計算負荷が大きいという問題を解決しているが、データ容量の小型化は実現できない．それに対して、データ容量を削減可能なXORを用いたランプ型の秘密分散法も提案されているが、情報が部分的に漏洩するという問題が生じる．そこで，本論文ではこれらの問題を解決するためにXORを用いる秘密分散法に対して，ランプ型秘密分散法と異なるアプローチによってデータ容量を削減する方法を提案する．この方法は高速化と小型化に加え，計算量的な安全性も実現する．

A method for reduction of high-speed data capacity of the secret sharing scheme using XOR

Yoshihide Nagai† Ryo Takaara† Keiichi Iwamura†

†Tokyo University of science
1-14-6 Kudankita, Chiyoda, Tokyo 102-0073, JAPAN
{nagai,takaara}@sec.ee.kagu.tus.ac.jp,iwamura@ee.kagu.tus.ac.jp

Abstract In recent years, the increasing importance of information security, anti-leakage measures and loss of information has become an important issue. (k,n) threshold secret sharing scheme proposed by Shamir is a lot of attention as a method of concealment of information and simultaneously, to avoid risk of loss. However, the Shamirs (k,n) secret sharing scheme the threshold there is a need to handle a polynomial of degree $k-1$ when distributed and recovered information, a problem when the size of the computational load that is applied to the actual application had become. In this paper, the method consists of the XOR operation to solve the problem, such as Uematsu, we propose a distributed and can be restored faster. Also done to prove safety.

1 はじめに

近年，情報の漏洩対策と紛失対策が重要な課題となっている．1979年にShamirによって提案された(k,n)閾値秘密分散法[1]は，情報の秘匿と，紛失によるリスクの回避を同時に実現する手法として注目を浴びている．しかしながら，Shamirの(k,n)閾値秘密分散法は秘密情報の分散・復元時にk-1次の多項式を処理する必要があり，その計算負荷の大きさが実際のアプリケーションに適用する際に問題となっていた．

そこで，栗原らによって，排他的論理和(XOR)を用いて高速に分散・復元可能な(k,n)閾値秘

密分散法 [3] が提案された．これによって，計算負荷が大幅に軽減され，高速処理が実現できる．また，この手法は一つの分散情報からは一切の秘密情報が漏洩せず，分散情報のデータ長と秘密情報のデータ長のサイズが同等となる理想的な秘密分散法である．

一方，秘密情報に関する情報が部分的に漏れる集合を許して，秘密保護に対する安全性の条件を緩めるかわりに，分散情報のデータ長を秘密情報のデータ長より小さくすることを実現できる方式として (k, L, n) ランプ型秘密分散法 [2] が知られている．XOR を用いる秘密分散に対しても栗原等によって，同様の手法 [4] が提案されている．これはある秘密情報を n 個に分散し，そのうち任意の k 個以上の分散情報が集まると秘密情報を復元でき，$k-L+1$ 個以上 $k-1$ 個以下の分散情報が集まると部分的に秘密情報を得ることができるが，$k-L$ 個以下の分散情報が集まっても秘密情報に関して全く情報が得られない方式である．しかしながら，この方式には情報が部分的に漏洩するため，安全性を重視する場面においては適切でない．

そこで，本論文では以下の特徴を持つ XOR を用いた新しい秘密分散方式を提案する．この方式は複数の秘密情報を扱う際，擬似乱数を用いて複数の分散情報を一つにまとめることで，ランプ型秘密分散と異なるアプローチによって XOR を用いる秘密分散法における高速処理に加えて，ストレージに保存するデータの小型化を実現する方式となっている．

1. 複数の秘密情報に対して栗原等の (k, n) 閾値秘密分散法を繰り返し独立に用いた場合よりもシステム全体で必要となる記憶容量が削減できる．
2. 秘密情報の分散と復元は各秘密情報に対して独立に行うことができる．
3. 秘密情報の復元過程において，他の秘密情報の復元に関する情報を復元者は得ることができない．少なくとも，閾値以下の分散値から秘密情報を得ることに対して計算量的安全性を実現する．

以上の特徴によって，提案号式はセンサノードなどのような計算能力が小さく記憶容量も小さい応用に対しても秘密分散を適用していくことが容易になる．

2 提案方式

本章では，2.1 において提案方式で扱う記号の定義を行い，2.2,2.3 で提案する XOR 演算を適応した分散・復元アルゴリズムについて説明する．

2.1 提案方式の記号

$\oplus$:ビット単位の XOR 演算

$\|$:ビット列の結合

$a \cup b$:a または b を用いる

n :分散数（ユーザの人数）

k :閾値

i :ユーザ番号

j :部分分散情報の番号 $(0 \leq j \leq n-2)$

n_p ：$n_p \geq n$ を満たす素数

N :自然数の集合

M :秘密情報の数 $(M \in N)$

m :秘密情報の番号 $(0 \leq m \leq M-1)$

t :情報を削減する人数 $(1 \leq t \leq k-1)$

d :各処理におけるデータのビット長 $(d \in N)$

P :n 人のユーザの集合 $(P = \{P_0, P_1, \cdots, P_{n-1}\})$

D :分散情報の計算・配布を行うディーラ

$\{0,1\}^d$:0 と 1 から構成される d ビットのデータ

S_m :秘密情報 $(S_m \in \{0,1\}^{(n-1)d}, S_m \leq n_p)$

$S_{(m,X)}$:部分秘密情報

$$(1 \leq X \leq n_p - 1), S_{(m,X)} \in \{0,1\}^d,$$
$$S_{(m,0)} \in \{0\}^d$$

$r^{\alpha}_{(m,\beta)}$:独立乱数

$$(r^{\alpha}_{(m,\beta)} \in \{0,1\}^d, 0 \leq \alpha \leq k-2, 0 \leq \beta \leq n_p - 1)$$

$a^{\alpha}_{(m,\beta)}$:係数

$$(a^{\alpha}_{(m,\beta)} \in \{0,1\}^d, 0 \leq \alpha \leq k-2, 0 \leq \beta \leq n_p - 1)$$

$W_{(i,j)}$ ：ユーザ P_i に配布される分散情報

$W_{(m,i,j)}$ ：ユーザ P_i に配布される部分分散情報

$$(W_{(m,i,0)} \| W_{(m,i,1)} \| \cdots \| W_{(m,i,n_p-1)} = W_{(m,i)}) \quad (0 \leq j \leq n_p - 2)$$

$GF(n_p)$ ：$GF(n_p) = \{0, 1, \cdots, n_p - 1\}$

key_i :擬似乱数発生器の鍵

$q_{(m,i)}$:擬似乱数 $q_{(m,i)} \in \{0,1\}^{(n-1)d}$

$q_{(m,i,j)}$:部分擬似乱数 $q_{(m,i,j)} \in \{0,1\}^d$

本章文中の四則演算は明示しない限り n_p を法としたものとする．例えば，演算 $c(a \pm b)$ は $c(a \pm b)\ mod\ n_p$ を意味する．また，希望する分散数 n が合成数である場合，まず (k, n_p) 閾値秘密分散法を構築し，その中の n 個を用いることで，目的を達成する．以降での提案方式の説明では $n = n_p$ とする．

2.2 分散アルゴリズム

提案方式は複数の分散情報を１つにすることによって、データ容量の削減を行うが、全ての分散情報の削減を行うユーザ，一部の分散情報の削減を行うユーザ、分散情報の削減を行わないユーザの 3 種類が存在する．一部の分散情報の削減を行うユーザはすべての分散情報の削減を行うユーザの変形と考えられるので、ここでは説明の簡略化のため，全ての分散情報の削減を行うユーザと，全ての分散情報を持つユーザの，2 通りのみを許可する場合の構成を示す．

D はユーザ番号 $i_0, i_1, \cdots, i_{n-1}$ の n 人から任意の t 人を選び，そのユーザ番号を i_T(分散情報の共通化を行うユーザ) とする．$(1 \leq T \leq t)$

step1) D は $1 \leq T \leq t$ において分散情報を削減するユーザ i_T に独立した乱数 r_{i_T} を初期値，key_{i_T} を鍵として与える．

step2) D はユーザ $i_T (1 \leq T \leq t)$ の乱数 $r_{(i_T)}$ を初期値，key_{i_T} を鍵として擬似乱数発生器を用いて $M(n-1)d$ ビットの擬似乱数 $q_{(m,i_T)}$ を生成する．

step3) D は $1 \leq T \leq t$ においてそれぞれ $n-1$ 個の d ビットの部分擬似乱数 $q_{(m,i_T,j)} (0 \leq j \leq n-2)$ に分割する．また，秘密情報 S_m を $n-1$ 個の d ビットの部分秘密情報に分割し，$S_{(m,0)}$ を生成する．

$$q_{(m,i_T)} = q_{(m,i_T,0)} \| q_{(m,i_T,1)} \| \cdots \| q_{(m,i_T,n-2)}$$

$$S_m = S_{(m,1)} \| S_{(m,2)} \| \cdots \| S_{(m,n-1)}$$

$$S_{(m,0)} \in \{0\}^d$$

step4) D は $Mt(n-1)$ 個の各部分擬似乱数

$$q_{(m,0,0)}, \cdots, q_{(m,0,n-2)}, q_{(m,1,1)}, \cdots, q_{(m,t,n-2)}$$

が以下の式が成り立つように d ビットの係数 a を $Mt(n-1)$ 個求める．

$$q_{(m,i_T,j)} = S_{(m,i_T-j)} \bigoplus \left\{ \bigoplus_{h=0}^{k-2} a^h_{(m,h \cdot i_T + j)} \right\}$$

$$(1 \leq T \leq t, 0 \leq j \leq n-2)$$

step5) D は乱数 r を独立に $M\{n(k-1) - t(n-1) - 1\}$ 個生成し，部分分散情報 $W_{(m,i,j)}$ を $0 \leq i \leq n-1, 0 \leq j \leq n-2$ において $i = i_T$ を除いてそれぞれ生成する．

$$W_{(m,i,j)} = S_{(m,i-j)} \bigoplus \left\{ \bigoplus_{h=0}^{k-2} (r \cup a)^h_{(m,h \cdot i + j)} \right\}$$

$$(0 \leq i \leq n-1, 0 \leq j \leq n-2)$$

step6) D は各部分分散情報

$$W_{(m,i,0)}, \cdots, W_{(m,i,n-2)}$$

を連結して $(n-1)d$ ビットの分散情報 $W_{(m,i)}$ を生成し，分散情報を削減したユーザ P_{i_T} 以外に配布する．

$$W_{(m,i)} = W_{(m,i,0)} \| W_{(m,i,1)} \| \cdots \| W_{(m,i,n-2)}$$

$$(W_{(m,i)} \in \{0,1\}^{(n-1)d})$$

以下に具体例を示す．

$(k,n) = (3,5)$ とし，秘密情報の数を $M = 2$ 分散情報を削減するユーザを $i_T = 0, 1$ とする．

step1) D はユーザ番号 $i_T = 0, 1$ に対して独立乱数 $r_{i_T} = r_0, r_1$ および鍵 $key_{i_T} = key_0, key_1$ を与える．

step2) D は独立乱数 $r_{i_T} = r_0, r_1$ を初期値，および $key_{i_T} = key_0, key_1$ を鍵として擬似乱数発生器を用いて $8d$ ビットの擬似乱数 $q_{(m,i_T)} = q_{(1,0)} \| q_{(2,0)}, q_{(1,1)} \| q_{(2,1)}$ を生成する．

step3) D は擬似乱数

$$q_{(m,i_T)} = q_{(1,0)}, q_{(1,1)}, q_{(2,0)}, q_{(2,1)}$$

をそれぞれ 4 個の d ビットの部分擬似乱数 $q_{(m,i_T,j)} = q_{(1,0,j)}, q_{(1,1,j)}, q_{(2,0,j)}, q_{(2,1,j)} (0 \leq j \leq 3)$ に分割する．また，秘密情報 $S_m = S_1, S_2$ を 4 個の d ビットの部分秘密情報に分割し，$S_{(m,0)} = S_{(1,0)}, S_{(2,0)}$ を生成する．

例)$i_T = 0$ の場合 ($i_T = 1$ の場合も同様にして行う)

$$q_{(1,0)} = q_{(1,0,0)} \| q_{(1,0,1)} \| q_{(1,0,2)} \| q_{(1,0,3)}$$

$$q_{(1,1)} = q_{(1,1,0)} \| q_{(1,1,1)} \| q_{(1,1,2)} \| q_{(1,1,3)}$$

$$S_1 = S_{(1,1)} \| S_{(1,2)} \| S_{(1,3)} \| S_{(1,4)}$$

$$S_{(1,0)} \in \{0\}^d$$

step4) D は 16 個の各部分擬似乱数が以下の式が成り立つように d ビットの係数 a を求める．

例)$i_T = 0$ の場合 ($i_T = 1$ の場合も同様にして行う)

$$q_{(1,0,0)} = S_{(1,0)} \bigoplus a^0_{(1,0)} \bigoplus a^1_{(1,0)}$$

$$q_{(1,0,1)} = S_{(1,4)} \bigoplus a^0_{(1,1)} \bigoplus a^1_{(1,1)}$$

$$q_{(1,0,2)} = S_{(1,3)} \bigoplus a^0_{(1,2)} \bigoplus a^1_{(1,2)}$$

$$q_{(1,0,3)} = S_{(1,2)} \bigoplus a^0_{(1,3)} \bigoplus a^1_{(1,3)}$$

$$q_{(1,1,0)} = S_{(1,1)} \bigoplus a^0_{(1,0)} \bigoplus a^1_{(1,1)}$$

$$q_{(1,1,1)} = S_{(1,0)} \bigoplus a^0_{(1,1)} \bigoplus a^1_{(1,2)}$$

$$q_{(1,1,2)} = S_{(1,4)} \bigoplus a^0_{(1,2)} \bigoplus a^1_{(1,3)}$$

$$q_{(1,1,3)} = S_{(1,3)} \bigoplus a^0_{(1,3)} \bigoplus a^1_{(1,4)}$$

ここで $a^1_{(1,0)}, a^1_{(1,4)}$ のどちらかを独立乱数 $r^1_{(1,0)}, r^1_{(1,4)}$(ここでは $r^1_{(1,0)}$) として与えることで，もう一方の定数 $a^1_{(1,4)}$ が定まる．

step5) D は求めた 16 個の定数および 2 個の独立乱数を用いて表 1 のように部分分散情報を生成する．

step6) D は各部分分散情報を連結して $(n-1)d$ ビットの分散情報 $W_{(m,i)}$ を生成し，ユーザ $P_i = P_2, P_3, P_4$ に配布する．

例)$i_T = 0$ の場合 ($i_T = 1$ の場合も同様にして行う)

$$W_{(1,0)} = W_{(1,0,0)} \| W_{(1,0,1)} \| W_{(1,0,2)} \| W_{(1,0,3)}$$

$$W_{(1,1)} = W_{(1,1,0)} \| W_{(1,1,1)} \| W_{(1,1,2)} \| W_{(1,1,3)}$$

$$W_{(1,2)} = W_{(1,2,0)} \| W_{(1,2,1)} \| W_{(1,2,2)} \| W_{(1,2,3)}$$

$$W_{(1,3)} = W_{(1,3,0)} \| W_{(1,3,1)} \| W_{(1,3,2)} \| W_{(1,3,3)}$$

$$W_{(1,4)} = W_{(1,4,0)} \| W_{(1,4,1)} \| W_{(1,4,2)} \| W_{(1,4,3)}$$

2.3 復元アルゴリズム

復元者は一つの秘密情報を復元したいとする．この場合，分散情報を削減したユーザ

$$P_{i_0}, P_{i_1}, \cdots, P_{i_{t-1}}$$

の t 人と分散情報を削減していないユーザ

$$P_{i_t}, P_{i_{t+1}}, \cdots, P_{i_k}$$

の $k-t$ 人の合わせて k 人から一つずつ分散情報を集め，秘密情報を復元する．

表 1: $k=3, n=5, t=2, m=1$ における部分分散情報の例

ユーザ	$W_{(m,i,j)}$	$j=0$	$j=1$	$j=2$	$j=3$
P_0	$W_{(1,0,j)}$	$S_{1,0} \bigoplus a^0_{1,0} \bigoplus r^1_{1,0}$	$S_{1,1} \bigoplus a^0_{1,1} \bigoplus a^1_{1,1}$	$S_{1,2} \bigoplus a^0_{1,2} \bigoplus a^1_{1,2}$	$S_{1,3} \bigoplus a^0_{1,3} \bigoplus a^1_{1,3}$
P_1	$W_{(1,1,j)}$	$S_{1,1} \bigoplus a^0_{1,0} \bigoplus a^1_{1,1}$	$S_{1,2} \bigoplus a^0_{1,1} \bigoplus a^1_{1,2}$	$S_{1,3} \bigoplus a^0_{1,2} \bigoplus a^1_{1,3}$	$S_{1,4} \bigoplus a^0_{1,3} \bigoplus a^1_{1,4}$
P_2	$W_{(1,2,j)}$	$S_{1,1} \bigoplus a^0_{1,0} \bigoplus a^1_{1,2}$	$S_{1,2} \bigoplus a^0_{1,1} \bigoplus a^1_{1,3}$	$S_{1,3} \bigoplus a^0_{1,2} \bigoplus a^1_{1,4}$	$S_{1,4} \bigoplus a^0_{1,3} \bigoplus r^1_{1,0}$
P_3	$W_{(1,3,j)}$	$S_{1,1} \bigoplus a^0_{1,0} \bigoplus a^1_{1,3}$	$S_{1,2} \bigoplus a^0_{1,1} \bigoplus a^1_{1,4}$	$S_{1,3} \bigoplus a^0_{1,2} \bigoplus r^1_{1,0}$	$S_{1,4} \bigoplus a^0_{1,3} \bigoplus a^1_{1,1}$
P_4	$W_{(1,4,j)}$	$S_{1,1} \bigoplus a^0_{1,0} \bigoplus a^1_{1,4}$	$S_{1,2} \bigoplus a^0_{1,1} \bigoplus r^1_{1,0}$	$S_{1,3} \bigoplus a^0_{1,2} \bigoplus a^1_{1,1}$	$S_{1,4} \bigoplus a^0_{1,3} \bigoplus a^1_{1,2}$

分散情報を削減したユーザ

$$P_{i_0}, P_{i_1}, \cdots, P_{i_{t-1}}$$

は擬似乱数生成器を用いて各自の乱数

$$r_{i_0}, \cdots, r_{i_{t-1}}$$

を初期値として各自に D から配布された鍵

$$key\ i_0, key\ i_1, \cdots, key\ i_{t-1}$$

を使用し，擬似乱数

$$q_{(m,i_0)}, q_{(m,i_1)}, \cdots, q_{(m,i_{t-1})}$$

$$(q_{(m,i)} = W_{(m,i)})$$

を生成し，該当する秘密情報に関する分散情報を提供する．

分散情報を削減していないユーザ

$$P_{i_t}, P_{i_{t+1}}, \cdots, P_{i_k}$$

は各自の分散情報のうち，該当する秘密情報に関するもの

$$W_{(m,i_t)}, W_{(m,i_{t+1})}, \cdots, W_{(m,i_k)}$$

を提供する．

この際，復元者は得たい秘密情報に関する分散情報のみが渡される．

ここで $W_{t_0}, \cdots, W_{t_{k-1}}$ の k 個の分散情報が集まったとする．$(0 \le t_0 \le \cdots \le t_k \le n-1)$

step1) k 個集まった分散情報を全て部分分散情報に分割する

$$W_{t_0} \leftarrow W_{(t_0,0)}, W_{(t_0,1)}, \cdots, W_{(t_0,n-2)}$$

$$\vdots$$

$$W_{t_{k-1}} \leftarrow W_{(t_{k-1},0)}, W_{(t_{k-1},1)}, \cdots, W_{(t_{k-1},n-2)}$$

step2) 集まった全ての各部分分散情報を以下のように表し $(kn-2)$ 元の 2 進数ベクトル $V_{(ti,j)}$ を生成する．

部分分散情報 $W_{(t_i,j)}$ の場合

$$W_{(t_i,j)} = V_{(t_i,j)} \cdot R_{(k,n)}$$

$$R_{(k,n)} = (S_1, \cdots, S_{n-1}, r^0_0, \cdots,$$

$$r^0_{n-2}, r^1_0, \cdots, r^1_{n-1}, \cdots, r^{k-2}_0, \cdots, r^{k-2}_{n-1})^T$$

例えば，$k=3, n=5$ の場合

$$W_{(2,1)} = S_4 \bigoplus r^0_1 \bigoplus r^1_3$$

$$R_{(3,5)} = (S_1, \cdots, S_4, r^0_0, \cdots, r^0_3, r^1_0, \cdots, r^1_4)^T$$

$$V_{(2,1)} = (0001\ 0100\ 00010)$$

と表す．

step3) step2) で集まった $V_{(t_0,0)}, \cdots, V_{(t_{k-1},n-2)}$ の $k(n-1)$ 個のベクトルから以下の 2 進数の $\{k(n-1) \times (kn-2)\}$ の行列 $M^{(k,n)}_{(t_0,\cdots,t_{k-1})}$ を生成する．

$$M^{(k,n)}_{(t_0,\cdots,t_{k-1})} = (V_{(t_0,0)}, \cdots, V_{(t_0,n-1)}, \cdots,$$

$$V_{(t_{k-1},0)}, \cdots, V_{(t_{k-1},n-1)})^T$$

step4) 集まった全ての部分分散情報を表す $k(n-1)$ 元ベクトル $W_{(t_0,\cdots,t_{k-1})}$ のように表す．

$$W_{(t_0,\cdots,t_{k-1})} = (W_{(t_0,0)}, \cdots,$$

$$W_{(t_0,n-2)}, \cdots, W_{(t_{k-1},0)}, \cdots, W_{(t_{k-1},n-2)})^T$$

$$W_{(t_0,\cdots,t_{k-1})} = M^{(k,n)}_{(t_0,\cdots,t_{k-1})} \cdot R_{(k,n)}$$

step5) Gauss-Jordan の消去法 (掃き出し法) を用いることによって行列 $M^{(k,n)}_{(t_0,\cdots,t_{k-1})}$ を 2 進数の行列 $G_{(k,n)} = G(M^{(k,n)}_{(t_0,\cdots,t_{k-1})})$ に変形することにより，全ての部分秘密情報の $k(n-1)$ 元ベクトル $S_{(k,n)}$ を求める．また，Gauss-Jordan の消去法 (掃き出し法) での計算処理は全て $GF(2)$ で行うものとする．まず，$G_{(k,n)}$ は以下のように表す．

$$G_{(k,n)} = \left\{ \begin{array}{cc} I & \Phi \\ \Phi & \Delta k \end{array} \right\}$$

$I : (n-1) \times (n-1)$ の単位行列
Φ:零行列

$$\Delta k = \left(\begin{array}{ccccc} I & \Phi \| c_1 & \Phi \| c_1 & \cdots & \Phi \| c_1 \\ \Phi & I \| c_1 & \Phi & \cdots & \Phi \\ \Phi & \Phi & I \| c_1 & \cdots & \Phi \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi & \Phi & \Phi & \cdots & I \| c_1 \end{array} \right)$$

c_1:$c_1 = (1, \cdots, 1)^T$　$n-1$ 元のベクトル

ここで，step4) の式 (1) に Gauss-Jordan の消去法 (掃き出し法) を用いることで以下のような式が求まる．

$$S_{(k,n)} = G_{(k,n)} \cdot R_{(k,n)}$$

また，$S_{(k,n)} = (S_1, S_2, \cdots, S_{n-1}, *, \cdots, *)^T$ と表せる．
($*$ は部分分散情報以外の XOR 演算した情報)
よって，全ての部分秘密情報を得る．

step6) 全ての部分秘密情報を連結して秘密情報 S を復元する．

$$S = S_1 \| S_2 \| \cdots \| S_{n-1}$$

3　提案方式の評価

提案方式の場合，$r_{21}, r_{22}, \cdots, r_{mk-1}$ の値を r_{i_T} を初期値として key_j を利用して生成した擬似乱数の値と等しくなるように設定することでユーザに配布する分散情報は r_{i_T} と key_j のみとなり，分散情報の数を減らすと共に，安全性も同様に保っている．本節では，提案方式の記憶容量に関する評価を行う．ただし，栗原らによる手法 [3] を従来手法として説明する．

複数の秘密情報 $s_1, s_2, \cdots, s_m$ について，提案方式では，分散情報を擬似乱数生成器を用いて生成するユーザが保管する情報は 1 つの乱数と鍵であり，その他のユーザは分散情報を M 個保管することとなる．そのため鍵のサイズを $|key_i|$ とすると，分散情報を 1 つだけ持つユーザに必要な記憶容量は

$$|key_i| + |W_i| = |key_i| + |s_i|$$

となる．ここで．簡単のため $|key_i| = |s_i|$ とすると，$2|s_i|$ となる．また，分散情報を複数持つユーザに必要な記憶容量は，

$$M \times |W_i| = M \times |s_i|$$

となる．また，分散情報を削減するユーザを t 人として，提案方式においてシステム全体で必要な記憶容量は，

$$(t \times 2|s_i|) + ((n-t) \times M|s_i|)$$

となる．ここで，従来法において，システム全体で必要な分散情報に関する記憶容量は，

$$Mn|W_i| = Mn|s_i|$$

であるので，提案方式においてシステム全体で必要な記憶容量と比較すると以下の様になる．

$$\frac{t \times 2|s_i| + (n-t) \times m|s_i|}{Mn|s_i|} = \frac{2t + M \times (n-t)}{Mn}$$

ここで，秘密情報の個数である M が十分大きい場合上記式は以下の様になる．

$$\frac{2 \times t + M \times (n-t)}{Mn} \simeq 1 - \frac{t}{n}$$

これより提案方式は分散情報を削減する人数が多いほど従来法に比べシステム全体として持つべき記憶容量を削減することが可能であり，分散人数を増加させることでも同様に記憶容量の削減効果があると言える．

ここで，具体的な例として $M = 100$ とした時の分散を行う人数 n および，削減を行う人数

表 2: 従来方式に対する提案方式の記憶容量比較

n	t	容量比
2	1	51%
3	1	67%
	2	35%
4	1	76%
	2	51%
	3	27%
5	1	80%
	2	61%
	3	41%
	4	22%

t における提案方式で必要な記憶容量を従来法と比較した場合の容量比を表 1 に示す (ただし，k は $k \times t + 1$ の条件を満たすとする)．例えば $n = 4$，$t = 3$ の場合，従来法に比べ提案方式は 27 % の容量で済むことが分かる．

これより，提案方式において t の値が大きくなるつまり，分散情報を初期値 r_{i_T} および鍵 key_i から擬似乱数を利用して生成するユーザが増えれば増えるほど従来法に比べ記憶容量の削減効果が大きいということがわかる．

また，安全性については頁制限のため証明は省略するが，用いる疑似乱数生成方式が計算量的に安全であれば，提案方式も安全であることが証明できる．

4 応用例

本章では，応用例としてセンサネットワークのような小さな計算能力および記憶容量しか持たない場面での使用法を示す．

センサネットワークは基地局，クラスタヘッド，エンドノードという構成要素からなるクラスタツリー型の通信トポロジであるとする．構成の様子を図 1 に示す．また，基地局は十分な計算能力および記憶容量を持っており，それ以外のノードは計算能力および記憶容量が小さいとする．特に，クラスタヘッドとなるノードは特殊なノードではなく，基地局以外のノードの中から選択されるとする．

クラスタヘッドがエンドノードとの通信を行う際，ノード毎に異なる鍵で共通鍵暗号などを用いた場合，クラスタヘッドは自身のクラスタに属する全てのノードの鍵を知っている必要がある．大規模なセンサネットワークを想定した場合，すべての鍵をノードがストレージに保管することは効率的ではない．そこで，提案方式をこの場合に当てはめることで各ノードの持つ情報を削減することが出来，効率的な鍵共有が可能である．

具体例を以下に示す．

あるプロトコルによりクラスタヘッドが選択されたとする．このとき，クラスタヘッドは自らのクラスタに属すノードの秘密鍵を知らない．各ノードは秘密通信に用いる秘密鍵を保有し，基地局はすべてのノード ID とその秘密鍵を知っているとする．このとき，クラスタヘッドは自らのクラスタに属すノードの ID を基地局に送る．基地局はディーラとしてクラスタヘッドになったノードの秘密情報を共通の分散値，各ノードの秘密情報を鍵（秘密情報）として本提案 $(n = k = 2)$ により各ノードに配布する分散値を定め，クラスタヘッドに送信する．クラスタヘッドは各ノードにその分散値を配布する．このとき，クラスタヘッドは各ノードへの分散値を記憶しない．各ノードは送られてきた分散値を記憶する．鍵共有時は各ノードは分散値をクラスタヘッドの送り，クラスタヘッドは秘密情報を復元することにより鍵を取り出す．

これにより，クラスタヘッドと各ノードは各ノードが予めもつ秘密情報を共有でき，それを秘密鍵とすることで暗号通信が実現できる．この場合，クラスタヘッドは M 個の秘密鍵を記憶する必要がない．

5 まとめ

本論文では 3 における記憶容量の評価より，提案方式では複数の秘密情報に対して栗原等の (k, n) 閾値秘密分散法を繰り返し独立に用いた場合よりもシステム全体で必要となる記憶容量

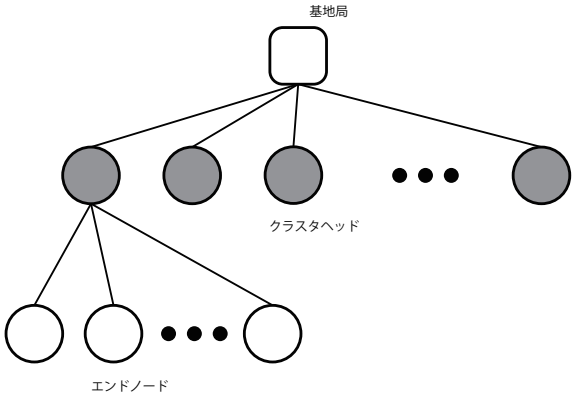

図 1: クラスターツリー型の例

が削減できることがわかった.

また，2.3 における分散・復元アルゴリズムから，提案方式では復元者に対して秘密情報の分散と復元は各秘密情報に対して独立に行うことがわかる．特に秘密情報の復元過程において，他の秘密情報の復元に関する情報を復元者は 得ることができないことがわかった.

さらに，4 において提案方式のセンサネットワークにおけるクラスタツリー型の鍵管理方式への応用例を示した.

以上から提案方式によって秘密分散法を扱う状況下において記憶容量や計算能力に制限のあるモバイル端末などでの情報管理にも適している.

参考文献

[1] A. Shamir,“ How to Share a Secret, ”Commun. ACM,vol.22, no.11, pp.612613, 1979.

[2] G.Blackley and C.Meadows: “ Security of Ramp Schemes, ”CRYPTO ’84,vol.196 pp.242-268,1984.

[3] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, “On a fast (k, n)-threshold secret sharing scheme,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,vol. E91-A, No. 9, pp. 2365-2378, Sep. 2008.

[4] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, “A fast (k, L, n)-threshold ramp secret sharing scheme,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E92-A, No. 8, pp. 1808-1821, Aug. 2009.

[5] Y. Fujii, K. Tochikubo, N. Hosaka, M. Tada, and T. Kato, “(k, n) threshold schemes using xor operations,” in Technical Report of IEICE, vol. 107, no. 44, ISEC2007-05, May 2007, pp. 31–38.

[6] 多田美奈子, 藤井吉弘, 保坂範和, 栃窪孝也, 加藤岳久,“ 閾値 3 の秘密分散法の構成法, ” CSS2005 予稿集, 2005.

[7] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, “A fast $(3, n)$-threshold secret sharing scheme using exclusive-or operations,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E91-A, No. 1, pp. 127–138, Jan. 2008.

[8] 山本博資,“ (k, L, n) しきい値秘密分散システム, ” 電子通信学会論文誌, vol.J68-A,no.9, pp.945-952 (1985)

[9] 馬場雪乃, 岩村惠市.“ メモリやデータベースに適した秘密分散法, ” SCIS2007 予稿集, 3D1-4, Jan.2007

[10] 植松裕基, 岩村惠市.“ メモリやデータベースに適した秘密分散法, ” SCIS2010 予稿集, 1F2-6, jan.2010