Vol.2014-CSEC-65 No.2
Vol.2014-IOT-25 No.2
2014/5/22

XOR を用いる秘密分散法の多値化とそれを用いた秘匿計算法

髙橋加寿子[†1]　須賀祐治[†2]　岩村惠市[†1]

近年，ビッグデータという言葉が話題になっている．ビッグデータの活用が進展していくためには，プライバシーや秘密情報の取り扱いが大きな課題となる．その為，プライバシーや秘密情報を秘匿しながらデータの分析や解析を行う秘匿演算の一手法として秘密分散法が注目されている．高速手法として栗原等が提案した XOR を用いた(k,n)秘密分散が知られているが、この手法は秘密情報を数値ではなくビット列として扱うため、秘匿演算ができない。今回、この XOR を用いた(*k,n*)閾値秘密分散法を拡張し，秘密情報をビット列ではなく，多値の数値として扱う(*k,n*)閾値秘密分散法を提案する．これにより、Shamir の秘密分散法に比べ処理の高速性を維持しつつ秘密情報を数値として扱い、秘匿演算を実現することができる．

Multivalue-converting of 2 value series in Secret Sharing Scheme using XOR, and Secure Function Evaluation which uses it

KAZUKO TAKAHASHI[†1]　YUJI SUGA[†2]
KEIICHI IWAMURA[†1]

Recently, secret sharing scheme has attracted attention as a technique for analyzing and analyze data while concealing the confidential information and privacy. This time, to extend the Kurihara scheme is a (*k,n*) threshold secret sharing scheme-fast using XOR, rather than a bit string secret information, we propose a (*k,n*) threshold secret sharing scheme to treat as a number of multi-valued. It is thereby possible, is treated as numeric secret information while maintaining the high speed of the process compared to the secret sharing scheme Shamir, realizing the concealment operation.

1. はじめに

近年，ビッグデータ(*bigdata*)という言葉が話題になっている.ビッグデータという厳密な定義はまだ存在しないが，その大きな特徴は，ビッグデータが単純なデータ量(*Volume*) の増加だけでなく，データの多様性(*Variety*)，発生の速さ(*Velocity*)，正確さ(*Veracity*)，を持つことである．さらに，その取得や分析の方法によってこれまでなかった革新的なサービスやビジネスモデルが実現可能だと理解されたため，現在の研究における一つのキーワードとなっている．ビッグデータの活用が進展していくためには，プライバシーや秘密情報の取り扱いが大きな課題となる．ビッグデータの活用を健全に推進していくには，この問題を解消しつつデータ分析の成果を享受できるように対策を講じていくことが必要となる．その対策の一つとして，プライバシー保護データマイニングがある．プライバシー保護データマイニングを実現する一手法として Shamir によって提案された(*k,n*)閾値秘密分散法（以降 Shamir 法）が上げられるが，Shamir 法は多項式を解く必要があるため，計算量が多くなるという問題があり膨大なデータに対する高速な演算が必要なビッグデータの演算には適していない

一方，(*k,n*)閾値秘密分散法における計算量を少なくする方式として栗原らによって、秘密情報をビット列として扱い XOR による演算だけで秘密分散を実現する手法（以降 XOR 法）が提案されている．また、栗原らによって記憶容量を削減できる XOR を用いたランプ型秘密分散法[6]も提案されている。このように，計算量と記憶容量の少ない XOR 法が提案されているが，これらの方式では準同型性を持たないため，秘匿計算に対応できないという問題がある．準同型性がある場合，分散情報をそのまま用いて演算を行うことが出来るため，ビッグデータを用いた統計計算などに有用である．そこで，本論文ではまず XOR 法を拡張するために、秘密情報をビット列ではなく、多値の数値として扱い，XOR でなく加算と減算だけで(*k,n*)閾値秘密分散法を実現する手法を提案する．次に，その多値の分散値を用いて秘匿計算を行うことを検討する。すなわち、XOR 法を多値化することによって，Shamir 法の問題であった *Lagrange* の補間公式による計算量の問題を解決し，さらに従来の XOR 法の問題であった秘匿計算を可能にする手法を提案する．

2. 従来方式

本章では秘密分散法の代表的な手法として Shamir 法と，

†1 東京理科大学
Tokyo University of Science
†2 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc.

栗原らによって提案された XOR 法について示す．後者の手法は秘密情報の分散・復元処理を高速に実現できる．また XOR を用いた閾値秘密分散法では秘密情報のデータ長と分散情報のデータ長が等しくなるという特徴がある．

2.1 Shamir の(*k*,*n*)閾値秘密分散法

本節では Shamir 法について示す．Shamir の提案した多項式補間による方法では，以下の様にして(*k*,*n*)閾値秘密分散法を実現する．

2.1.1 分散アルゴリズム

(1)$S<p$ かつ $n<p$ である任意の素数 p を選び，以下の演算は mod p 上で行うものとする．
(2)GF(p)の元から，異なる n 個の $x_i(i=1,\cdots,n)$を選びだし，ユーザ *ID* とする．
(3)GF(p)の元から，$k-1$ 個の乱数 $a_l(l=1,\cdots,k-1)$を選んで，以下の式を生成する．

$$W_i = S + a_1 x_i + a_2 x_i^2 + \cdots + a_{k-1} x_i^{k-1}$$

(4)上記式の x_i に各ユーザ *ID* を代入して，分散情報 W_i を計算し，各ユーザにこれらのユーザ *ID* x_i と生成した分散情報 W_i を配付する．

2.1.2 復元アルゴリズム

(1)復元に用いる分散情報を $W_{im}(m=1,\cdots,k)$とする．また，その分散情報に対応するユーザ *ID* を x_{im} とする．
(2)分散式に x_{im} と W_{im} を代入し，k 個の連立方程式を解いて，秘密情報 S を得る．S の復元の際には，*Lagrange* の補間公式を用いると便利である．

この方式は，多くの秘密情報を扱った場合でもそれぞれ独立に多項式を定めることにより、秘密情報を独立に復元することができ，それぞれの分散情報から秘密情報を復元してもほかの秘密情報に関する情報を一切得ることはできない．しかし，容量削減効果については，システム全体で必要となる記憶容量は元の秘密情報の n 倍となり多くのデータを取り扱う場合を想定すると，非常に効率が悪い．

2.2 XOR を用いた(k,n)閾値秘密分散法

本節では，XOR を用いた閾値秘密分散法の分散，復元アルゴリズムについて説明する．

2.2.1 記号の定義

本節で使用する記号及び演算子を以下に定義する．

$\oplus$:ビット単位の XOR 演算
$\|$:ビット列の結合
n :分散数
k :閾値
i :ユーザ番号
j :部分分散情報の番号 $(0 \le j \le n-2)$
n_p :$n_p \ge n$ を満たす素数
N :自然数の集合
d :各処理におけるデータのビット長
P :n 人のユーザの集合
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 :秘密情報($S\in\{0,1\}^{(n-1)d}$)
S_x :部分秘密情報($1\le x\le n_p$-1,$S_0\in\{0\}^d$)
r^{α}_{β} :数($r^{\alpha}_{\beta}\in\{0,1\}^d, 0\le\alpha\le k$-2,$0\le\beta\le n_p$-1)
W_i :ユーザ P_i に配布される分散情
$W_{(i,j)}$:ユーザ P_i に配布される部分分散情報
$GF(n_p)$:$GF(n_p)=\{0,1,\cdots,n_p-1\}$

XOR 演算を用いた高速な(*k*,*n*)閾値秘密分散法のアルゴリズムに関する四則演算は，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 $c(a\pm b) \bmod n_p$ を意味する．また，希望する分散数 n が合成数である場合，(k,n_p)閾値秘密分散法を構成し，その中から分散情報を n 個用いることで，(*k*,*n*)閾値秘密分散法を実現している．よって，以降での XOR 演算を用いた高速な(*k*,*n*)閾値秘密分散法の説明では，簡単化のため $n=n_p$ とする．

2.2.2 分散アルゴリズム

(1)D は秘密情報 S を n-1 個の部分秘密情報に分割し、加えて $S_0\in\{0\}^d$ を生成する．

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

(2)D は d ビットの乱数 r^{α}_{β} を全て独立に(k-1)n-1 個生成する．

$$r_0^0, r_1^0, \cdots r_{n-2}^0, r_0^1, \cdots, r_{n-2}^1, r_{n-1}^1, \cdots, r_0^{k-2}, \cdots, r_{n-1}^{k-2} \in \{0,1\}^d$$

(3)D は部分分散情報 $W_{(i,j)}$ を以下の式により $0\leqq i\leqq n$-1,$0\leqq j\leqq n$-2 においてそれぞれ生成する．

$$W_{(i,j)} = S_{j-i} \oplus \left\{ \bigoplus_{h=0}^{k-2} r_{h\cdot i+j}^{h} \right\} \in \{0,1\}^d$$
$$(0 \le i \le n-1, 0 \le j \le n-2)$$

(4)D は $0\le i\le n$-1 において各部分分散情報 $W_{(i,0)}, W_{(i,1)}, \cdots, W_{(i,n-2)}$を連結して($n$-1)$d$ ビットの分散情報 W_i を生成し各ユーザに配布する．

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

2.2.3 復元アルゴリズム

(1)復元に用いる分散情報を $W_{t0},\cdots,W_{tk-1}$ とする$(0\le t_0\le\cdots\le t_k\le n$-1) ．$k$ 個の分散情報を部分分散情報に分割する．

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

(2)分割した部分分散情報を以下のように表し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

$$W_{(t_i,j)} = V_{(t_i,j)} \cdot R_{(k,n)}$$
$$R_{(k,n)} = \left(S_1, \cdots, S_{n-1}, r_0^0, \cdots, r_{n-2}^0, r_0^1, \cdots, r_{n-1}^1, \cdots, r_0^{k-2}, \cdots, r_{n-1}^{k-2}\right)^T$$

(3)ベクトル $V_{(t0,0)}, \cdots, V_{(tk-1,n-2)}$ から以下の行列を生成する.

$$M_{(t_0,\cdots,t_{k-1})}^{(k,n)} = \left(V_{(t_0,0)}, \cdots, V_{(t_0,n-1)}, \cdots, V_{(t_{k-1},0)}, \cdots, V_{(t_{k-1},n-1)}\right)^T$$

(4)部分分散情報を以下のベクトル $W_{(t0,\cdots,tk-1)}$のように表す.

$$W_{(t_0,\cdots,t_{k-1})} = \left(W_{(t_0,0)}, \cdots, W_{(t_0,n-2)}, \cdots, W_{(t_{k-1},0)}, \cdots, W_{(t_{k-1},n-2)}\right)^T$$
$$W_{(t_0,\cdots,t_{k-1})} = M_{(t_0,\cdots,t_{k-1})}^{(k,n)} \cdot R_{(k,n)}$$

(5)Gauss-Jordan の消去法を用いて行列 $M_{(t_0,\cdots,t_{k-1})}^{(k,n)}$ を2進数の行列 $G_{(k,n)} = G\left(M_{(t_0,\cdots,t_{k-1})}^{(k,n)}\right)$ に変形することにより、全ての部分秘密情報のベクトル $S_{(k,n)}$ を求める.

(6)全ての部分秘密情報を連結し，秘密情報 S を得る.

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

3. 提案方式

2章で示した栗原らが提案したXOR法は秘密情報をビット列として扱っているため高速に演算が行える利点があるという一方で，準同型性を持たないため秘匿演算ができないという問題をもつ．そのため，XOR 法でビット列として扱っている秘密情報を数値として扱えるよう多値化を行い，*Lagrange* の補間公式を用いずに加算・減算のみで分散・復元を行う方式を以下に示す.

3.1 提案方式の記号の定義

$\|$:ビット列の結合
n :分散数
k :閾値
i :ユーザ番号
j :部分分散情報の番号
n_p: $n_p \geq n, n_p > S,$を満たす素数
N :自然数の集合
P :n 人のユーザの集合
D :分散情報の計算・配布を行うディーラ
S :秘密情報
Sx :部分秘密情報$(1 \leq x \leq n_p\text{-}1, S_0 \in \{0\}^d)$
$r^{\alpha}{}_{\beta}$:乱数$(0 \leq \alpha \leq k\text{-}2, 0 \leq \beta \leq n_p\text{-}1)$
Wi :ユーザ Pi に配布される分散情
$W(i,j)$:ユーザ Pi に配布される部分分散情報
$GF(n_p)$:$GF(n_p)=\{0,1,\ldots,n_p\text{-}1\}$

本章の(k,n)閾値秘密分散法のアルゴリズムに関する四則演算は，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，(k,n_p)閾値秘密分散法を構成し，その中から分散情報を n 個用いることで，(k,n)閾値秘密分散法を実現している．よって，以降での (k,n)閾値秘密分散法の説明では，簡単化のため $n=n_p$ とする.

3.2 分散アルゴリズム

(1)D は秘密情報 S を n-1 個の部分秘密情報に分割し、加えて $S_0 \in \{0\}^d$ を生成する.

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

(2)D は d ビットの乱数 $r^{\alpha}{}_{\beta}$ を全て独立に$(k\text{-}1)n\text{-}1$ 個生成する.

$$r_0^0, r_1^0, \cdots r_{n-2}^0, r_0^1, \cdots, r_{n-2}^1, r_{n-1}^1, \cdots, r_0^{k-2}, \cdots, r_{n-1}^{k-2}$$

(3)D は部分分散情報 $W_{(i,j)}$ を以下の式により $0 \leqq i \leqq n\text{-}1, 0 \leqq j \leqq n\text{-}2$ においてそれぞれ生成する.

$$W_{(i,j)} = S_{j-i} + \left\{ \overset{k-2}{\underset{h=0}{+}} r_{h \cdot i+j}^h \right\}$$
$$\left(0 \leq i \leq n-1, 0 \leq j \leq n-2\right)$$

また，次のときの $S_{j\text{-}i}$ の符号を反転する.

i=1 かつ j=2,3
$i \geq 2$ かつ j=1

(4)D は $0 \leq i \leq n\text{-}1$ において各部分分散情報 $W_{(i,0)}, W_{(i,1)}, \cdots, W_{(i,n-2)}$を連結して分散情報 W_i を生成し各ユーザに配布する.

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

3.3 復元アルゴリズム

(1)復元に用いる分散情報を $W_{t0}, \cdots, W_{tk\text{-}1}$ とする$(0 \leq t_0 \leq \cdots \leq t_k \leq n\text{-}1)$ ． k 個の分散情報を部分分散情報に分割する.

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

(2)分割した部分分散情報を以下のように表し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

$$W_{(t_i,j)} = V_{(t_i,j)} \cdot R_{(k,n)}$$
$$R_{(k,n)} = \left(S_1, \cdots, S_{n-1}, r_0^0, \cdots, r_{n-2}^0, r_0^1, \cdots, r_{n-1}^1, \cdots, r_0^{k-2}, \cdots, r_{n-1}^{k-2}\right)^T$$

(3)ベクトル $V_{(t0,0)}, \cdots, V_{(tk-1,n-2)}$ から以下の行列を生成する.

$$M_{(t_0,\cdots,t_{k-1})}^{(k,n)} = \left(V_{(t_0,0)}, \cdots, V_{(t_0,n-1)}, \cdots, V_{(t_{k-1},0)}, \cdots, V_{(t_{k-1},n-1)}\right)^T$$

(4)部分分散情報を以下のベクトル $W_{(t0,\cdots,tk-1)}$のように表す.

$$W_{(t_0,\cdots,t_{k-1})}=\left(W_{(t_0,0)},\cdots,W_{(t_0,n-2)},\cdots,W_{(t_{k-1},0)},\cdots,W_{(t_{k-1},n-2)}\right)^T$$
$$W_{(t_0,\cdots,t_{k-1})}=M^{(k,n)}_{(t_0,\cdots,t_{k-1})}\cdot R_{(k,n)}$$

(5)Gauss-Jordan の消去法を用いて行列$M^{(k,n)}_{(t_0,\cdots,t_{k-1})}$を 2 進数の行列$G_{(k,n)}=G\left(M^{(k,n)}_{(t_0,\cdots,t_{k-1})}\right)$ に変形することにより、全ての部分秘密情報のベクトル $S_{(k,n)}$ を求める.

$G_{(k,n)}$は以下のように表す.

$$G_{(k,n)}=\begin{pmatrix} I & \Phi \\ \Phi & \Delta k \end{pmatrix}$$

$I:(n-1)\times(n-1)$の単位行列

$\boldsymbol{\Phi}$**:**零行列

$$\Delta k=\begin{pmatrix} I & 0|c_1 & 0|c_1 & \cdots & 0|c_1 \\ 0 & I|c_1 & \Phi & \cdots & \Phi \\ 0 & \Phi & I|c_1 & \cdots & \Phi \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \Phi & \Phi & \Phi & I|c_1 \end{pmatrix}$$

$0:(n-1)\times(n-1)$の零位行列

$c_1:c_1=(1,\cdots,1)^T$ $n-1$元のベクトル

|:連結

この際，(4)の式に Gauss-Jordan の消去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，$S_{(k,n)}$は以下のように表せる.

$S_{(k,n)}=(S_1,S_2,\cdots,S_{n-1},*,\cdots,*)^T$

よって，全ての部分秘密情報を得る.

(6)全ての部分秘密情報を連結し，秘密情報 S を得る.

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

4. 提案方式の評価

4.1 計算量

本章では秘密分散法の代表的な手法とされる Shamir 法と提案方式との計算量の比較を行う．今回の計算量の比較は両方式の分散・復元処理に生じる際の加算・乗算の回数による比較である.

なお, 本章の演算は Shamir 法においても提案方式と同様に, 演算は mod n_p($S<n_p$ かつ $n<n_p$ である任意の素数 n_p)上 で行っているとする.

Shamir 法の分散における計算量が生じる手順は **2.1.1 分散アルゴリズム**において以下の手順(3)である.

(3)GF(p)の元から，$k-1$ 個の乱数 a_l(l=1,⋯,k-1)を選んで，以下の式を生成する.

$$W_i=S+a_1x_i+a_2x_i^2+\cdots+a_{k-1}x_i^{k-1}$$

この時，1 つの分散情報 W_i には k-1 個の加算と $k-1+\frac{(k-1)(k-2)}{2}$個の乗算が行われている．分散情報は n 個となっているため加算と乗算の個数はそれぞれ$n(k-1)$個と$n\left(k-1+\frac{(k-1)(k-2)}{2}\right)$個となる.

Shamir 法の復元における計算量が生じる手順は **2.1.2 復元アルゴリズム**において以下の手順(2)である.

(2)分散式に x_{im} と W_{im} を代入し, k 個の連立方程式を解いて, 秘密情報 S を得る.

この時，集められた k 個の分散情報を $f(ID_{j1}),f(ID_{j2}),\cdots,f(ID_{jk})$としたとき以下の式を解くことで S が求められると考えられる.

$$S=\sum_{i=1}^{k} f\left(ID_{ji}\right)\prod_{\substack{1\le r\le k\\ r\neq 1}}\frac{ID_{jr}}{ID_{jr}-ID_{ji}}$$

上記の式より加算と乗算の個数は $2k$ 個と $2k$-1 個となる.

提案方式の分散における計算量が生じる手順は **3.2.分散アルゴリズム**において以下の手順(3)である.

(3)D は部分分散情報 $W_{(i,j)}$を以下の式により $0\leqq i\leqq n$-1,$0\leqq j\leqq n$-2 においてそれぞれ生成する.

$$W_{(i,j)}=S_{j-i}+\left\{\sum_{h=0}^{k-2} r_{h\cdot i+j}^{h}\right\}$$
$$(0\le i\le n-1,0\le j\le n-2)$$

また，次のときの S_{j-i} の符号を反転する.

i=1 かつ j=2,3

$i\geq 2$ かつ j=1

この時，1 つの部分分散情報 $W_{(i,j)}$には k-1 個の加算が行われる．また部分分散情報 $W_{(i,j)}$は $0\leqq i\leqq n$-1,$0\leqq j\leqq n$-2 において生成される為，個数は $n\times(n$-1)個となる．よって加算の個数は $n(n$-1)(k-1)個となる.

提案方式の復元における計算量が生じる手順は **3.3.分散アルゴリズム**において以下の手順(5)である.

(5)Gauss-Jordan の消去法を用いて行列$M^{(k,n)}_{(t_0,\cdots,t_{k-1})}$を 2 進数の行列$G_{(k,n)}=G\left(M^{(k,n)}_{(t_0,\cdots,t_{k-1})}\right)$ に変形することにより、全ての部分秘密情報のベクトル $S_{(k,n)}$ を求める.

この時，1 つの部分秘密情報を求めるために k 個の分散

情報からそれぞれ約 k 個の部分分散情報を集め加減算する必要があると考えられる．また，部分秘密情報は n-1 個存在する．よって加算の個数は$k^2(\mathrm{n}-1)$個となる．

Shamir 法と提案方式の分散処理・復元処理における加算・乗算の計算量の比較を表 1 に示す．

表 1　Shamir 法と提案方式の
分散・復元処理における計算量の比較

	Shamir 法		提案方式	
	分散処理	復元処理	分散処理	復元処理
加算	$\mathrm{n}(\mathrm{k}-1)$	$2\mathrm{k}$	$\mathrm{n}(\mathrm{n}-1)\times(\mathrm{k}-1)$	$k^2(\mathrm{n}-1)$
乗算	$n\left(k-1+\frac{(k-1)(k-2)}{2}\right)$	$2\mathrm{k}+1$	0	0

提案方式では分散・復元において全ての処理を加算のみで行っている．一般的に加算は乗算に比べて十分に計算量が小さいため，分散・復元に乗算が必要な Shamir 法と比較し計算量が少なくなっていると考えられる．

また，今回は加算・乗算の回数のみの計算量の比較を行った．しかし提案方式では Shamir 法と違い秘密情報の分割を行うが，法となる n_p は同じとしているため提案方式は Shamir 法より大きなサイズの秘密情報を扱っていることになる。よって、秘密情報のサイズを同じにすると提案方式は法となる n_p が Shamir 法よりも小さくなると考えられる．なお，秘密情報のサイズを同じにした場合の厳密な計算量の比較に関しては今後の課題とする．

5. 提案手法を用いた秘匿演算

5.1 分散情報同士の加算

秘密情報 S の分散情報$W_{t_0},\cdots,W_{t_{k-1}}$と秘密情報 S'の分散情報$W'_{t_0},\cdots,W'_{t_{k-1}}$のそれぞれ$k$個の分散情報の加算について以下に示す．$(0 \le t_0 \le \cdots \le t_k \le n-1)$

(1)ユーザは保持している分散情報を分割し部分分散情報とする．この時，それぞれの部分分散情報の最上位に 0 を置く．例:123→0123

(2)ユーザは保持している部分分散情報同士を加算する．

$$W_{t_0}+W'_{t_0} \to W_{(t_0,0)}+W'_{(t_0,0)}, W_{(t_0,1)}+W'_{(t_0,1)}, \cdots, W_{(t_0,n-2)}+W'_{(t_0,n-2)}$$
$$\vdots$$
$$W_{t_{k-1}}+W'_{t_{k-1}} \to W_{(t_{k-1},0)}+W'_{(t_{k-1},0)}, W_{(t_{k-1},1)}+W'_{(t_{k-1},1)}, \cdots, W_{(t_{k-1},n-2)}+W'_{(t_{k-1},n-2)}$$

(3)加算した部分分散情報を復元する．部分分散情報を以下のように表し 2 進数ベクトル$V_{(t_i,j)}$を生成する．

部分分散情報$W_{(t_i,j)}$の場合

$$W_{(t_i,j)} = V_{(t_i,j)} \cdot R_{(k,n)}$$
$$R_{(k,n)} = \begin{pmatrix} S_1+S'_1,\cdots,S_{n-1}+S'_{n-1}, r_0^0+r_0'^0,\cdots,r_{n-2}^0+r_{n-2}'^0, r_0^1+r_0'^1 \\ ,\cdots,r_{n-1}^1+r_{n-1}'^1,\cdots,r_0^{k-2}+r_0'^{k-2},\cdots,r_{n-1}^{k-2}+r_{n-1}'^{k-2} \end{pmatrix}^T$$

(4)ベクトル$V_{(t_0,0)},\cdots,V_{(t_{k-1},n-2)}$から以下の行列を生成する．

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

(5)部分分散情報を以下のベクトル$W_{(t_0,\cdots,t_{k-1})}+W'_{(t_0,\cdots,t_{k-1})}$のように表す．

$$W_{(t_0,\cdots,t_{k-1})}+W'_{(t_0,\cdots t_{k-1})} = \begin{pmatrix} W_{(t_0,0)}+W'_{(t_0,0)},\cdots,W_{(t_0,n-2)}+W'_{(t_0,n-2)},\cdots, \\ W_{(t_{k-1},0)}+W'_{(t_{k-1},0)},\cdots,W_{(t_{k-1},n-2)}+W'_{(t_{k-1},n-2)} \end{pmatrix}^T$$
$$W_{(t_0,\cdots,t_{k-1})}+W'_{(t_0,\cdots t_{k-1})} = M^{(k,n)}_{(t_0,\cdots,t_{k-1})} \cdot R_{(k,n)}$$

(6)Gauss-Jordan の消去法を用いて行列$M^{(k,n)}_{(t_0,\cdots,t_{k-1})}$を 2 進数の行列$G_{(k,n)} = G\left(M^{(k,n)}_{(t_0,\cdots,t_{k-1})}\right)$に変形することにより、全ての部分秘密情報の $k(n\text{-}1)$元のベクトル $S_{(k,n)}$を求める．

まず，$G_{(k,n)}$は以下のように表す．

$$G_{(k,n)} = \begin{pmatrix} I & \Phi \\ \Phi & \Delta k \end{pmatrix}$$

I: $(n-1)\times(n-1)$の単位行列

Φ : 零行列

$$\Delta k = \begin{pmatrix} I & 0|c_1 & 0|c_1 & \cdots & 0|c_1 \\ 0 & I|c_1 & \Phi & \cdots & \Phi \\ 0 & \Phi & I|c_1 & \cdots & \Phi \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \Phi & \Phi & \Phi & I|c_1 \end{pmatrix}$$

0: $(n-1)\times(n-1)$の零位行列

C_1: $C_1 = (1,\cdots,1)^T$　$n-1$元のベクトル

｜:連結

ここで，(5)の式に Gauss-Jordan の消去法を用いることで以下のような式が求まる．

$$S_{(k,n)} + S'_{(k,n)} = G_{(k,n)} \cdot R_{(k,n)}$$

また，$S_{(k,n)} + S'_{(k,n)}$は$(S_1+S'_1, S_2+S'_2, \cdots, S_{n-1}+S'_{n-1}, *, \cdots, *)^T$と表せる．よって，全ての部分秘密情報を得る．

(7)全ての部分秘密情報を 1 桁ずつずらし連結することで，秘密情報 S+S'を得ることが出来る．

以上の手順を用いることで分散情報同士の加算を行うことができるが，分散情報同士の減算に関しても加算と同様の手順を行うことで実現可能である．

6. まとめ

本論文では XOR を用いた高速な(k,n)閾値秘密分散法である栗原方式を応用し，多値化された値での適応が可能であり，加法準同型性を持つ(k,n)閾値秘密分散法を示した．この方式は *Lagrange* の補間公式を用いないことで少ない計

算量を実現している．本論文の提案方式の安全性の検証と秘密情報の大きさを考慮した厳密な計算量の比較については今後の課題とする．

参考文献

[1] 武田浩一，井手剛，“ビッグデータ処理の展望”PROVISION No.72 IBM(2012)
[2]佐久間淳，小林重信，“プライバシ保護データマイニング”人工知能学会誌 vol.24,no.2(2009)
[3] A. Shamir, “How to Share a Secret”, Commun. ACM, vol.22,no.11, pp.612-613, 1979.
[4]栗原淳， 清元晋作， 福島和英， 田中俊昭，“排他的論理和を用いた高速な閾値法の機能拡張(1)” ISEC2007,107 巻,209 号, pp.1-8(2007)
[5]栗原淳，清本晋作，福島和英，田中俊昭，“排他的論理輪を用いた高速な(4,*n*)閾値秘密分散法と(*k*,*n*)閾値法への拡張”ISEC2007-4, pp.23-30 (2007)
[6]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.
[7]高荒亮，岩村惠市，“XOR を用いた高速な(*k*,*L*,*n*)ランプ型秘密分散法に関する研究” 情報処理学会シンポジウム論文集，2009 巻，11 号，pp.949-954
[8]永井良英，高荒亮，岩村惠市，“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”CSS2012
[9]高橋慧，岩村惠市，“クラウドコンピューティングに適した計算量的安全性を持つ秘密分散法” CSS2012
[10]岡本栄司 “暗号理論入門[第 2 版]”共立出版株式会社, 2002 年発行