Vol.2017-CSEC-77 No.16
Vol.2017-IOT-37 No.16
2017/5/26

多値化法による秘匿四則演算及びその秘匿部分一致検索への応用

鴇田恭平[†1]　岩村惠市[†1]

概要：栗原らは秘密分散を計算量の少ない XOR 演算のみで実現する手法（XOR 法）を提案したが，それは四則演算に対する準同型性を持たないため，秘匿演算に対応できない．そこで， XOR 法をもとに，多値のデータを扱う手法（多値化法）に拡張する．さらに，その多値化法によって秘匿四則演算を実現できる手法を提案する．秘密分散法を用いた秘匿四則演算は，加算準同型性と定数倍準同型性があれば神宮らが提案した次数を変化させずに四則演算を行える手法（TUS 法）によって実現できる．ただし，この手法には秘密情報に 0 を含む場合，秘密情報が漏洩するため，秘密情報に 0 を含まないという条件がある．多値化法は XOR 法と同様に秘密情報を分割して秘密分散を行うが，それに TUS 法をそのまま適用すると，分割された部分秘密情報に 0 を含むとその部分秘密情報が漏洩する．そこで，本論文では多値化法にコンパクト法を適用しても部分秘密情報が漏洩しない手法を提案し，その応用として秘匿部分一致検索を実現する．

キーワード：秘密分散法，秘匿演算，準同型性，秘匿検索，部分一致

The safety and application of the secret sharing scheme using multi-value method to concealment partial match retrieval

KYOHEI TOKITA[†1]　KEIICHI IWAMURA[†2]

Abstract: Kurihara et al. suggested the XOR method, which can lessen a computational cost. But we can't do concealment computations with it because it doesn't have a homomorphism. In order to solve the problem, we suggest the multi-value method based on it, which handles multi-value data. And we suggest the method to do concealment computations with it. We can do concealment computations in secret sharing scheme which have additive homomorphism and constant factor homomorphism by using TUS method proposed by Shingu et al, which enables them do concealment computations without changing the dimension. But because the leak of secret data will happen if one or more parts of secret data are zero, secret data can't include zero. In this paper, we propose the method which can use TUS method without the leak of parts of secret data, and we realize concealment partial match retrieval using the method.

Keywords: Secret Sharing Scheme, Concealment computation, Homomorphism, Concealment retrieval, Partial match retrieval

1. はじめに

近年，ビッグデータという言葉が注目されている．その大きな特徴は，単純に膨大なデータ量を持つことだけでなく，データの多様性，発生の速さ，正確さも持つことである．ウェブサービス分野では現在既に活用が進んでおり，さらに各データを連携させることでさらなる付加価値の創出も期待されるため，現在の研究における一つのキーワードとなっている[1]．一方でプライバシーや秘密情報の取り扱いが大きな課題である[2]．ビッグデータの活用を推進するには，この問題を解消しつつデータ分析結果を享受できる対策を講じる必要がある．その対策の一つに，プライバシー保護データマイニングがある[3]．中でも秘匿演算手法として Shamir の(k,n)閾値秘密分散法(以降 Shamir 法)[4]が上げられるが， k が大きい場合の秘密情報の分散・復元時の計算負荷が問題となっていた．

栗原らが計算量の少ない XOR 法[5]を提案したが，準同型性を持たないため，秘匿演算に対応できない．それを解決するため， XOR 法をもとに，多値のデータを扱う多値化法に拡張する．更に秘匿乗算にも対応させるため，神宮らが提案した TUS 法[6]の適用を考えるが，そのまま適用すると，分割された部分秘密情報に 0 を含むとその部分秘密情報が漏洩する問題があった．

そこで，本論文では多値化法にコンパクト法を適用しても部分秘密情報が漏洩しない手法を提案し，その応用として秘匿部分一致検索を実現する．

2. 従来方式

本章では栗原方式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・復元を行う多値化法を以下に示す．

2.1 多値化法

2.1.1 多値化法の記号の定義

|| :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

†1 東京理科大学
Tokyo University of Science

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
S_x :部分秘密情報($1 \leq x \leq n_p$-1,$S_0 \in \{0\}^d$)
r^{α}_{β} :乱数($0 \leq \alpha \leq k$-2,$0 \leq \beta \leq n_p$-1)
W_i :ユーザ P_i に配布される分散情報
$W_{(i,j)}$: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,…,n_p-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 $c(a \pm b) \bmod n_p$ 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，以降の説明では，簡単化のため $n=n_p$ とする．

2.1.2 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．

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

(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．

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

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

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

(4)D は $0 \leq i \leq n$-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．

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

2.1.3 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$(0 \leq t_0 \leq \cdots \leq t_k \leq n-1)$ ．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．

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

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

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．

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

2.2 TUS 法

Ben-Or らによる従来の秘匿乗算[7]は Shamir 法による分散値をそのまま用いて乗算するため，復元に必要な分散値の数が 2k-1 個に増加する．しかし，TUS 法は，秘密情報に乱数を乗じて秘匿化秘密情報を生成し，それを秘密分散する．秘匿乗算を行う際には，秘匿化秘密情報を一時的に復元してスカラー量として扱い，他の分散値と乗算を行う．これにより, 乗算した際に多項式の次数は増加しないので, 閾値を変化させないコンパクトな秘匿乗算を行うことができる．

以下，TUS 法で用いる記号の定義，分散と復元，秘匿乗算，秘匿加減算について示す．また，コンパクト法での値の選択および演算はGF(p)上で行われ，用いられる乱数は 0 を除くGF(p)上の一様乱数である．

2.2.1 記号定義

a,b,c:秘密情報
$\alpha_j, \beta_j, \gamma_j$:乱数

$\overline{[x]_i}$:値 x に対するサーバS_iの保持する分散値
$[x]_i$:値 x に関連してサーバS_iが保持する分散値集合

2.2.2 分散

(1)ディーラは k 個の乱数α_j(j=0,…,k-1)を生成し, $\alpha=\prod_{j=0}^{k-1}\alpha_j$を計算する.
(2)ディーラはαaを計算し，$\alpha a,\alpha_0,\ldots,\alpha_{k-1}$を Shamir 法で n 台のサーバに分散する.
(3)サーバS_i(i=0,…,n-1)は秘密情報 a に対する分散値集合として$[a]_i:=(\overline{[\alpha a]_i},\overline{[\alpha_0]_i},\ldots,\overline{[\alpha_{k-1}]_i})$を保持する.

2.2.3 復元

(1)ユーザは k 台のサーバS_jから$[a]_j$を受け取る.
(2)ユーザはサーバS_jから受け取った$[a]_j:=(\overline{[\alpha a]_j},\overline{[\alpha_0]_j},\ldots,\overline{[\alpha_{k-1}]_j})$を用いてαa,$\alpha_0,\ldots,\alpha_{k-1}$を復元し，以下より a を復元する.

$$\alpha=\prod_{j=0}^{k-1}\alpha_j$$

$$\alpha a\times\alpha^{-1}=a$$

2.2.4 秘匿乗除算

c=abを計算するとき，各サーバS_iは 2.2.2 で計算された秘密情報 a,b に関する分散値集合$[a]_i$と$[b]_i$を持ち, 以下の手順により分散値集合$[c]_i$を計算する．また，(1)におけるS_0は任意に定められる．ただし，a≠0とする.
(1)サーバS_0は任意の k 台のサーバS_j(j=0,…,k-1)から$\overline{[\alpha a]_j}$を受け取る.
(2)サーバS_0はαaを復元し，すべてのサーバへαaを送信する.
(3)全てのサーバS_i(i=0,…,n-1)はそれぞれ$\overline{[\alpha\beta ab]_i}=\alpha a\times\overline{[\beta b]_i}$を計算する.
(4)k 台のサーバS_j(j=0,…,k-1)は任意の k 台のサーバから各々$\left(\overline{[\alpha_0]_j},\ldots,\overline{[\alpha_{k-1}]_j}\right),\left(\overline{[\beta_0]_j},\ldots,\overline{[\beta_{k-1}]_j}\right)$を集め，$\alpha_j,\beta_j$を復元する.
(5)k 台のサーバS_jは$\alpha_j\beta_j$を計算して，n 台のサーバに分散する.
(6)全サーバS_iはそれぞれ秘密情報 c の分散値集合として$[c]_i:=(\overline{[\alpha\beta ab]_i},\overline{[\alpha_0\beta_0]_i},\ldots,\overline{[\alpha_{k-1}\beta_{k-1}]_i})$を保持する.

秘匿除算を行う場合は，(3)における計算を$\overline{[\beta b/\alpha a]_i}=\overline{[\beta b]_i}/\alpha a$に，(5)における$\alpha_j\beta_j$を$\beta_j/\alpha_j$とすることによって$[c]_i:=(\overline{[\beta b/\alpha a]_i},\overline{[\beta_0/\alpha_0]_i},\ldots,\overline{[\beta_{k-1}/\alpha_{k-1}]_i})$を得ることができる.

2.2.5 秘匿加減算

c=a±bを計算するときに，各サーバS_iは 2.2.2 で計算された秘密情報 a，b に関する分散値集合$[a]_i$と$[b]_i$を持ち，以下の手順により分散値集合$[c]_i$を計算する.
(1)k 台のサーバS_j(j=0,…,k-1)は任意の k 台のサーバからそれぞれ$\left(\overline{[\alpha_0]_j},\ldots,\overline{[\alpha_{k-1}]_j}\right),\left(\overline{[\beta_0]_j},\ldots,\overline{[\beta_{k-1}]_j}\right)$を集め，$\alpha_j,\beta_j$を復元し，乱数$\gamma_j$を生成して，サーバ$S_0$に$\gamma_j/\alpha_j,\gamma_j/\beta_j$を送信する.
(2)サーバS_0は以下よりγ/α,γ/βを復元し，全てのサーバに送信する.

$$\frac{\gamma}{\alpha}=\prod_{j=0}^{k-1}\frac{\gamma_j}{\alpha_j}\qquad\frac{\gamma}{\beta}=\prod_{j=0}^{k-1}\frac{\gamma_j}{\beta_j}$$

(3)全てのサーバS_i(i=0,…,n-1)はそれぞれ以下の計算を行う.

$$\overline{[\gamma(a\pm b)]_i}=\frac{\gamma}{\alpha}\overline{[\alpha a]_i}\pm\frac{\gamma}{\beta}\overline{[\beta b]_i}$$

(4)k 台のサーバS_j(j=0,…,k-1)はγ_jを n 台のサーバに分散する.
(5)全てのサーバS_i(i=0,…,n-1)は分散値集合$[c]_i=(\overline{[\gamma(a\pm b)]_i},\overline{[\gamma_0]_i},\ldots,\overline{[\gamma_{k-1}]_i})$を保持する.

3. 提案方式

2 章で示された XOR 法は四則演算に関する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点を持つ. また, XOR 法の秘密情報を単に数値として扱っても, 秘密情報を復元する際にガウスの消去法が必要となり，かえって Shamir 法より計算量が大きくなってしまう．また，XOR 法の復元における計算を単純に加減算のみに置き換えると, 部分秘密情報を求める際に係数が発生し，乗除算が必要となる問題があった.

そこで， 多値化法に TUS 法を適用しても部分秘密情報が漏洩しない手法でかつ，従来よりも計算量を削減できる手法を提案する.

3.1 記号の定義

n :分散数
k :閾値
i :ユーザ番号$(i\in GF(n_p))$
j :部分分散情報の番号$(j\in GF(n_p-1))$
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
S_x :部分秘密情報$(1\leq x\leq n_p-1, S_0\in\{0\}^d)$
r^{α}_{β} :乱数$(0\leq\alpha\leq k-2, 0\leq\beta\leq n_p-1)$
W_i :ユーザ P_i に配布される分散情報
$W_{(i,j)}$:ユーザ P_i に配布される部分分散情報
$GF(n_p)$:$GF(n_p)=\{0,1,\ldots,n_p-1\}$

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

Vol.2017-CSEC-77 No.16
Vol.2017-IOT-37 No.16
2017/5/26

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，以降での提案方式の説明では，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 を素数 p 以下の n-1 個の e 進数の部分秘密情報に分割し，$S_0,S_1,\ldots,S_{n-1}$ を生成する．この時，$S_0 = 0$ とする．

$$S=S_1\cdot e^{n-2}+S_2\cdot e^{n-3}\cdots+S_{n-1}\cdot e^0=\sum_{i=1}^{n-1} S_i\cdot e^{n-1-i}$$

(2)D は乱数 r_β^α を全て独立に(k-1)n-1 個生成する．

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

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

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

また，以下の場合 S_{j-i} の符号を反転し，$-S_{j-i}$ に相当する有限体上の元を用いる．

i=1 かつ j=2,3 または i≥2 かつ j=1

(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．

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

3.3 復元アルゴリズム

(1)k 個の集まった分散情報から全ての部分分散情報を取り出す．

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

(2)ここで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 元のベクトル $V_{(t_i,j)}$ を生成し，部分分散情報に含まれている $R_{(k,n)}$ の成分に対応した部分は 1(S_{j-i} の符号を反転した場合は-1)，その他は 0 とする．

部分分散情報 $W_{(t_i,j)}$ の場合，

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

(3)(2)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 $\{k(n-1)\times(kn-1)\}$ の行列 $M_{(t_0,\cdots,t_{k-1})}^{(k,n)}$ を生成する．

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

(4)集まったすべての部分分散情報を k(n-1)元のベクトル $W_{(t_0,\cdots,t_{k-1})}$ と表す．

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

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

ここで，行列 $M_{(t_0,\cdots,t_{k-1})}^{(k,n)}$ を Gauss-Jordan の消去法(掃き出し法)を用いて対角化処理を行う．これによって，全ての部分秘密情報に該当する部分を求めて $S_1,\ldots,S_{n-1}$ を得る．

(5)全ての部分秘密情報を以下の式に代入して秘密情報を復元する．

$$S=S_1\cdot e^{n-2}+S_2\cdot e^{n-3}\cdots+S_{n-1}\cdot e^0=\sum_{i=1}^{n-1} S_i\cdot e^{n-1-i}$$

3.4 提案方式の準同型性

3.4.1 加法準同型性

秘密情報 S の分散情報 $W_{t_0},\cdots,W_{t_{k-1}}$，秘密情報 S'の分散情報 $W'_{t_0},\cdots,W'_{t_{k-1}}$ が，2.1.2 の手順によってそれぞれ生成されているものとし，任意の k 人のユーザ $(t_0,\ldots,t_{k-1}\in GF(n),t_0\le\cdots\le t_{k-1})$ から k 個ずつの分散情報 $W_{t_0},\cdots,W_{t_{k-1}},W'_{t_0},\cdots,W'_{t_{k-1}}$ が集まった場合の秘密情報の秘匿加算を示す．

(1)ユーザは保持している対応する部分分散情報同士すべてを加算する．

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

ここで，$W''_{(i,j)}$ は次式のようになる．

$$W''_{(i,j)}=S_{j-i}+S'_{j-i}+\left(\sum_{h=0}^{k-2} r_{h\cdot i+j}^h\right)+\left(\sum_{h=0}^{k-2} r'^h_{h\cdot i+j}\right)$$

$S''_{j-i}=S_{j-i}+S'_{j-i}$, $\left(\sum_{h=0}^{k-2} r''^h_{h\cdot i+j}\right)=\left(\sum_{h=0}^{k-2} r_{h\cdot i+j}^h\right)+\left(\sum_{h=0}^{k-2} r'^h_{h\cdot i+j}\right)$ とすると，

$$W''_{(i,j)}=W_{(i,j)}+W'_{(i,j)}=S''_{j-i}+\left(\sum_{h=0}^{k-2} r''^h_{h\cdot i+j}\right)$$

と書くことができ，3.3 の手順に従い，復元することができる．

(2)2.1.3 の手順に従い，全ての部分秘密情報を復元する．このとき復元される部分秘密情報は以下のとおりである．

$$(S''_1,S''_2,\ldots,S''_{n-1})=(S_1+S'_1,S_2+S'_2,\ldots,S_{n-1}+S'_{n-1})$$

(3)復元した全ての部分秘密情報を以下の式に代入して秘密情報を復元する．

$$S''_1\cdot e^{n-2}+S''_2\cdot e^{n-3}+\cdots+S''_{n-1}\cdot e^0=\sum_{i=1}^{n-1}(S_i+S'_i)\cdot e^{n-1-i}=S+S'$$

これは加算の例だが，減算も同様に可能である．よって，提案方式は加法準同型性を持つことが分かる．

3.4.2 定数倍準同型性

秘密情報 S の分散情報$W_{t_0},\cdots,W_{t_{k-1}}$が，3.2 の手順によって生成されているものとし，任意の k 人のユーザ$(t_0,\ldots,t_{k-1}\in GF(n),t_0\leq\cdots\leq t_{k-1})$から，k 個の分散情報$W_{t_0},\cdots,W_{t_{k-1}}$が集まった場合の秘密情報の定数倍乗算を示す．

(1)ユーザは保持している部分分散情報すべてに任意の定数 a を乗じる．

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

ここで，$aW_{(i,j)}$は次式のようになる．

$$aW_{(i,j)}=aS_{j-i}+a\left(\sum_{h=0}^{k-2}r^{h}_{h\cdot i+j}\right)$$

$aS_{j-i}=S'_{j-i}$, $a\left(\sum_{h=0}^{k-2}r^{h}_{h\cdot i+j}\right)=\left(\sum_{h=0}^{k-2}r'^{h}_{h\cdot i+j}\right)$とすると，

$$W'_{(i,j)}=aW_{(i,j)}=S'_{j-i}+\left(\sum_{h=0}^{k-2}r'^{h}_{h\cdot i+j}\right)$$

と書くことができ，3.3 の手順に従い，復元することができる，

(2)3.3 の手順に従い，全ての部分秘密情報を復元する．このとき，復元される部分秘密情報は以下のとおりである．

$$(S'_1,S'_2,\ldots,S'_{n-1})=(aS_1,aS_2,\ldots,aS_{n-1})$$

(3)復元した全ての部分秘密情報を以下の式に代入して秘密情報を復元する．

$$S'_1\cdot e^{n-2}+S'_2\cdot e^{n-3}+\cdots+S'_{n-1}\cdot e^{0}=a\sum_{i=1}^{n-1}S_i\cdot e^{n-1-i}=aS$$

よって，提案方式は定数倍乗算準同型性を持つことが分かる．

3.5 提案方式への TUS 法の適用

秘密分散法を用いて秘匿演算をするためには，加法準同型性だけではなく，乗除算にも対応する必要がある．TUS 法は，加法準同型性と定数倍乗算準同型性を持つ秘密分散法に適用可能であるので，提案方式にも適用可能である．しかし，TUS 法における秘匿乗算は，一方の秘密情報に乱数を乗じた秘匿化秘密情報の形で復元し，それを他方の秘密情報の分散値にスカラーとして乗じる手法である．したがって秘匿乗算において秘匿化秘密情報が 0 となると，秘匿乗算結果が漏洩してしまうため，扱う秘密情報は 0 以外でなくてはならない．また一方で提案方式において，TUS 法をそのまま適用すると，秘密情報を部分秘密情報に分割した際に部分秘密情報が 0 となる恐れがあり，秘匿化部分秘密情報を復元した時点で，その部分秘密情報が漏洩してしまうため，それに対応するための手法を提案する．ただし，ここではn_p及び S より大きな素数 p を法とし，分散する秘密情報は 0 でないとする．

3.5.1 記号の定義

$\overline{[x]}_j$:値xに対するサーバP_jの保持する分散値

$[x]_j$:値xに関連してサーバP_jの保持する分散値集合

α_i,β_i:サーバ P_i が生成する乱数$\left(\prod_{i=0}^{k-1}\alpha_i=\alpha,\prod_{i=0}^{k-1}\beta_i=\beta\right)$

$[\alpha]_j=\left(\overline{[\alpha_0]}_j,\ldots,\overline{[\alpha_{k-1}]}_j\right)$:乱数αに対する分散値集合

$[\beta]_j=\left(\overline{[\beta_0]}_j,\ldots,\overline{[\beta_{k-1}]}_j\right)$:乱数βに対する分散値集合

W_j:ユーザ $P_j\left(j\in GF(n_p)\right)$に配布される秘密情報 S に関する分散情報

W'_j:ユーザ P_j に配布される秘密情報 S'に関する分散情報

なお，全ての分散値はすべて同じ n_p,k,e により分散されているものとする．$\left(i\in GF(k),j\in GF(n_p)\right)$

3.5.2 分散アルゴリズム

(1)D は秘密情報 S を素数 p 以下の n-1 個の e 進数の部分秘密情報に分割し，$S_1,\cdots,S_{n-1}$ を生成する．この時，$S_0=0$ とする．

$$S=S_1\cdot e^{n-2}+S_2\cdot e^{n-3}\cdots+S_{n-1}\cdot e^{0}=\sum_{i=1}^{n-1}S_i\cdot e^{n-1-i}$$

(2)D は k 個の乱数α_iを生成し，3.2 の手順で分散し，その積αを計算する．ただし，$\alpha\in GF(e)\backslash 0$とする．

(3)(1)で生成した部分秘密情報に(3)で得たαを乗じ，$\alpha S_1,\ldots,\alpha S_{n-1}$を得る．

(4)3.2(2)以降と同様の手順で分散を行う．その際，$S_1,\ldots,S_{n-1}$を用いる代わりに$\alpha S_1,\ldots,\alpha S_{n-1}$を用いる．

ここで，提案方式において部分分散情報に用いられている部分秘密情報は以下のようになる，

$$S_0,\alpha S_1,\ldots,\alpha S_{n-1}$$

3.5.3 復元アルゴリズム

(1)3.3 の手順に従い，k 個の分散情報$W_{t_0},\cdots,W_{t_{k-1}}$から全ての部分分散情報を復元し，$\alpha S_1,\ldots,\alpha S_{n-1}$を得る．

(2)復元した全ての部分秘密情報を以下の式に代入して秘匿化秘密情報を復元する．

$$\alpha S_1\cdot e^{n-2}+\alpha S_2\cdot e^{n-3}+\cdots+\alpha S_{n-1}\cdot e^{0}=\sum_{i=1}^{n-1}\alpha S_i\cdot e^{n-1-i}=\alpha S$$

(3)分散されたα_iをすべて復元し，以下を計算する．

$$\alpha S\prod_{i=0}^{k-1}\alpha_i^{-1}=S$$

3.5.4 秘匿加減算

それぞれ k 個ずつの分散情報を集めた時の TUS 法による

Vol.2017-CSEC-77 No.16
Vol.2017-IOT-37 No.16
2017/5/26

秘匿加減算を示す.

(1)サーバP_i(i=0,1,…,k-1)は指定された i に対する$\overline{[\alpha_i]_j}$,$\overline{[\beta_i]_j}$を k 個集め，α_i,β_iを復元し，乱数γ_iを生成して分散しておき，γ_i/α_i,γ_i/β_iを計算する.
(2)復元者はそれぞれ k 個のγ_i/α_i,γ_i/β_iを集め，以下の式よりγ/α,γ/βを復元し，演算に参加するほかの k-1 台のサーバに送る.

$$\frac{\gamma}{\alpha}=\prod_{i=0}^{k-1}\frac{\gamma_i}{\alpha_i} \qquad \frac{\gamma}{\beta}=\prod_{i=0}^{k-1}\frac{\gamma_i}{\beta_i}$$

(3)サーバP_j(j=0,1,…,n-1)は以下の演算をして，$M_{ADDj}(M_{SUBj})$を生成する.

$$M_{ADDj}\left(M_{SUBj}\right)=\frac{\gamma}{\alpha}W_j\pm\frac{\gamma}{\beta}W'_j$$

(4)全てのサーバP_j(j=0,…,n-1)は分散値集合$[S\pm S']_j=(M_{ADD(SUB)j},\overline{[\gamma_0]_j},\ldots,\overline{[\gamma_{k-1}]_j})$を保持する.

ここで k 個の$M_{ADDj}(M_{SUBj})$から復元される秘匿化秘密情報は，以下に示すとおりである.

$$\gamma(S\pm S')$$

3.5.5 秘匿乗除算

それぞれ k 個ずつの分散情報を集めた時のコンパクト法による秘匿乗除算を示す．あらかじめ 0 の分散値として，以下の部分秘密情報によって構成されるW_j^0が用意されているものとする.

$$\left(S_0^0,S_1^0,S_2^0,\ldots,S_{n-2}^0,S_{n-1}^0\right)=\left(0,-\gamma e,\gamma e(e-1),\ldots,\gamma e(e-1),\gamma e^2\right)$$

(1)3.5.4 の手順に従い，W_j^0とW_jの秘匿加算を行い，全てのサーバP_j(j=0,…,n-1)は分散値集合$[S+0]_j=(M_{ADDj},\overline{[\delta_0]_j},\ldots,\overline{[\delta_{k-1}]_j})$を保持する.

(2)復元者は分散値集合$[S+0]_j$を k 個集めて，3.5.3(1)(2)の手順に従いδSを復元し，演算に参加するほかの k-1 台のサーバに送る，ただし，δS=0のとき，除算は実行できない.
(2)サーバP_i(i=0,1,…,k-1)はW'_iと$\delta S\left(\delta S^{-1}\right)$を掛け合わせ$M_{MULi}(M_{DIVi})$を生成する.

$$M_{MULi}(M_{DIVi})=(\delta S)^{\pm1}(W'_{(i,0)},W'_{(i,1)},\ldots,W'_{(i,n-2)})$$

(3)サーバP_i(i=0,1,…,k-1)は指定された i に対する$\overline{[\delta_i]_j}$,$\overline{[\beta_i]_j}$を k 個集め，δ_i,β_iを復元し，$\delta_i\beta_i\left(\beta_i/\delta_i\right)$を計算する.
(5)k 台のサーバP_jは，$\delta_j\beta_j$を n 台のサーバに分散する.
(6)全サーバP_iはそれぞれ秘密情報 SS'(S'/S)の分散値集合として$[SS']_i:=(\overline{[\beta(\delta S)^{\pm1}S']_i},\overline{[\delta_0^{\pm1}\beta_0]_i},\ldots,\overline{[\delta_{k-1}^{\pm1}\beta_{k-1}]_i})$を保持する.

ここで k 個の$M_{MULi}(M_{DIVi})$から復元される秘匿化秘密情報は．以下に示すとおりである.

$$\beta(\delta S)^{\pm1}S'$$

4. 安全性

XOR 法の安全性証明[9]を応用することで提案方式の安全性を証明する．まず，コンパクト法の安全性として，以下が[7]に示されている.

・秘匿乗除算においてサーバで 1 度αSを復元するが，αSから秘密情報 S が漏洩することはない.

1 つの秘匿化秘密情報αSの値に対して，秘密情報 S は GF(p)\0上で均一に分散されている.その為, αSと S は独立している．よって，以下のように示すことができる.

$$H(S|\alpha S)=H(S)$$

同様に以下も示すことができる.

$$H(\alpha|\alpha S)=H(\alpha)$$

次に以下の 2 つの定理により，提案方式の安全性を証明する.

定理 1:k-1 個以下の分散情報からは秘密情報に関する情報は一切漏洩しない.
定理 2:k 個以上の分散情報から秘密情報を復元することが可能である.

定理 1 および 2 は，以下の式で表現でき，それぞれ定理 1 の証明，定理 2 の証明により導かれる.

$$H(S|V_A)=\begin{cases}H(S)\ (A\notin\Gamma)\\ 0\ (A\in\Gamma)\end{cases}$$

ただし，(k,n)閾値秘密分散法のアクセス構造Γは$\Gamma=\left\{A\in2^P\middle||A|\geq k\right\}$と定義されている.

4.1.1 定理 1

ユーザ数が|A|≤k-1を満たすような任意の集合 A を用意する．A はアクセス構造Γには含まれていないため，以下を満たす.

$$H(S|V_A)=H(S)$$

ここでV_Aは A 内の各ユーザに与えられている分散時の確率変数を表す.

[定理 1 の証明]

$A=\left\{P_{t_0},\cdots,P_{t_{k-2}}\right\}$は k-1 人のユーザを示す．ここで$t_0,\cdots,t_{k-2}\in GF\left(n_p\right)$は$i\neq j$において$0\leq t_i,t_j\leq n-1,t_i\neq t_j$を満たす任意の数である.

これに対して，$V_A=\left\{w_{t_0},\cdots,w_{t_{k-2}}\right\}$は k-1 個の確率変数を示し，分散情報$W_{t_0},\cdots,W_{t_{k-2}}$によってそれぞれ誘導される．同様にして$w_{(t_i,0)},\cdots,w_{(t_i,n_p-2)}$は，部分分散情報$W_{(t_i,0)},\cdots,W_{(t_i,n_p-2)}$により導かれる確率変数である.

ここで，次の条件を想定する.

・部分秘密情報$S_1,\cdots,S_{n_p-1}$と擬似乱数$r_0^0,\cdots,r_{n_p-2}^0,\cdots,r_0^{k-2},\cdots,r_{n_p-1}^{k-2}$は互いに独立している.

・$r_0^0,\cdots,r_{n_p-2}^0,\cdots,r_0^{k-2},\cdots,r_{n_p-1}^{k-2}$は一様な確率 1/p である有限集合

Vol.2017-CSEC-77 No.16
Vol.2017-IOT-37 No.16
2017/5/26

$\{0,1,\cdots,p\text{-}1\}$から選択される．

次に，k-1 個の分散情報を構成する以下の式を満たすよう U と V 行列を定義する．

$$W=U\cdot r+V\cdot S=\left(W_{(t_0,0)},\cdots,W_{(t_0,n_p-2)},\cdots,W_{(t_{k-2},0)},\cdots,W_{(t_{k-2},n_p-2)}\right)^T$$

ここでr,Sはそれぞれ以下のように表される．

$$S=\left(S_0,S_1,\cdots,S_{n_p-1}\right)^T,\ r=\left(r_0^0,\cdots,r_{n_p-2}^0,r_0^1,\cdots,r_{n_p-1}^1,\cdots,r_0^{k-2},\cdots,r_{n_p-1}^{k-2}\right)^T$$

S について，$S_0=0$ であるが，説明の簡単のため変数として扱う．分散情報内におけるそれぞれの擬似乱数 r の有無を定義する U 行列については XOR 法と同様な式で表すことが出来，含む部分秘密情報を定義する V 行列は，符号反転部分があるため，XOR 法と一部異なっている．

U と V はそれぞれ $(k\text{-}1)(n_p\text{-}1)\times\{(k\text{-}1)n_p\text{-}1\}$ と $(k\text{-}1)(n_p\text{-}1)\times n_p$行列である．ここで式変形をして次式を得る．

$$U\cdot r=W\text{-}V\cdot S$$

U は線形独立であるから，U·rによって得られた $(k\text{-}1)(n_p\text{-}1)$次元のベクトルのすべての要素はペアごとの独立かつ均一に分散された$\{0,1,\cdots,p\text{-}1\}$上の乱数である．

したがって S から得られた W は，それぞれ GF(p)上で均一に分散されている．その為，S と W は独立している．よって，以下のように示すことができる．

$$H\left(S_1,\cdots,S_{n_p-1}\left|w_{(t_0,0)},\cdots,w_{(t_0,n_p-2)},\cdots,w_{(t_{k-2},n_p-2)}\right.\right)$$

$$=H\left(S\left|w_{t_0},\cdots,w_{t_{k-2}}\right.\right)=H\left(S_1,\cdots,S_{n_p-1}\right)=H(S)$$

以上より，$H(S|V_A)=H(S)$を満たす．

提案方式にコンパクト法を適用した場合，分散情報から復元される必要がある秘密情報はαS, αの 2 つであり，次式を得る．

$$H(\alpha S|V_A)=H(\alpha S)\ (A\notin\Gamma)$$
$$H(\alpha|V_A)=H(\alpha)\ (A\notin\Gamma)$$

よって提案方式では k 個未満の分散情報から秘密情報を正しく復元できないので，以下の式を満たす．

$$H(S|V_A)=H(S)\ (A\notin\Gamma)$$

4.1.2 定理 2

提案方式は k 個以上の分散情報から秘密情報を復元することが可能なことを示す．

[定理 2 の証明]

提案方式において k 個の分散情報について以下の式を満たすよう U と V 行列を定義する．

$$W=U\cdot r+V\cdot S=\left(W_{(t_0,0)},\cdots,W_{(t_0,n_p-2)},\cdots,W_{(t_{k-1},0)},\cdots,W_{(t_{k-1},n_p-2)}\right)^T$$

ここでr,Sはそれぞれ以下のように表される．

$$S=\left(S_0,S_1,\cdots,S_{n_p-1}\right)^T,\ r=\left(r_0^0,\cdots,r_{n_p-2}^0,r_0^1,\cdots,r_{n_p-1}^1,\cdots,r_0^{k-2},\cdots,r_{n_p-1}^{k-2}\right)^T$$

[U V]行列において，行列変換を行うことで以下を生成することが出来る．

$$[U\ V]\cdot\begin{bmatrix}r\\S'\end{bmatrix}=\left(*,\ldots,*\left|(S_\alpha\text{-}S_\beta),(S_{\alpha+1}\text{-}S_{\beta+1}),\ldots,(S_{\alpha+m}\text{-}S_{\beta+m})\right.\right)^T$$

$$S'=\left(S_1,\cdots,S_{n_p-1}\right)^T$$

ただしα,βは次の式により決定される．$t_0,\cdots,t_{k-1}\in GF(n_p)$は,i≠jにおいて$t_i\neq t_j$を満たすような任意の数であり，m は $0\le m\le n_p\text{-}2$を満たす．

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

従って，k 個の分散情報を加減算することにより，以下に示される集合 X の全ての要素を得ることが出来る．

$$X=\{S_{\alpha+m}\text{-}S_{\beta+m}\mid 0\le m\le n_p\text{-}2\}$$

部分秘密情報を復元するために，以下に示される集合 X’を考える．

$$X'=\{x_m=S_{\alpha+m}\text{-}S_{\beta+m}\mid 0\le m\le n_p\text{-}1\}$$

次に，集合 X’にm=-αを代入する．

$$x_\alpha=S_0\text{-}S_{\beta-\alpha}$$

S_0の値は 0 であり，$\beta\text{-}\alpha=2\left(t_{k-2}\text{-}t_{k-1}\right)=C$とすると以下のように示せる．

$$S_C=\text{-}x_{\text{-}\alpha}$$

同様に，m にそれぞれの値を代入することで全ての部分秘密情報を復元することが出来る．

$m=-\alpha$	$S_C=-x_{-\alpha}$
$m=C-\alpha$	$S_{2C}=S_C-x_{C-\alpha}$
$m=2C-\alpha$	$S_{3C}=S_{2C}-x_{2C-\alpha}$
⋮	⋮
$m=(n_p-2)c-\alpha$	$S_{(n_p-1)C}=S_{(n_p-2)C}-x_{(n_p-2)C-\alpha}$

次に示す集合は$GF(n_p)$上の有限体であり，以下のようになる．

$$\{C,2C,\ldots,(n_p-1)C\}=\{1,2,\ldots,n_p-1\}=GF(n_p)\backslash 0$$

これより，次式が成立する．

$$\left\{S_C,S_{2C},\ldots,S_{(n_p-1)C}\right\}=\{S_1,S_2,\ldots,S_{n_p-1}\}$$

よって，k 個の分散情報より，部分秘密情報をすべて求められたので，(k,n)閾値秘密分散法のアクセス構造Γが$\Gamma=\{A\in 2^P\,||A|\ge k\}$を満たしたこととなり，次式が成り立つ．ここで$V_A$は A 内の各ユーザに与えられている分散時の確率変数を表す．

$$H(S|V_A)=0$$

提案方式にコンパクト法を適用した場合，分散情報から復元される必要がある秘密情報はαS, αの 2 つである．定理 2 より，次式を得る．

$$H(\alpha S|V_A)=0\ (A\in\Gamma)$$
$$H(\alpha|V_A)=0\ (A\in\Gamma)$$

よって提案方式では k 個以上の分散情報から秘密情報を正しく復元することができるので，以下の式を満たす．

$$H(S|V_A)=0\ (A\in\Gamma)$$

5. 秘匿部分一致検索への応用

次にコンパクト方式を適用した提案方式において，秘匿部分一致検索を適用するアルゴリズムを説明する．

<準備>

(1)5.2(3)以降の分散アルゴリズムを利用して，キーワード$K=(0,K_1,\dots,K_{n-1})$(0 を除く各成分K_iを部分キーワードとする)，キーワードに関連する秘密情報$S=(0,S_1,\dots,S_{n-1})$を以下のようにサーバに分散する．

$$[K]_j=\left(\overline{[\beta K]}_j,\overline{[\beta_0]}_j,\dots,\overline{[\beta_{k-1}]}_j\right)$$

$$[S]_j=\left(\overline{[\alpha S]}_j,\overline{[\alpha_0]}_j,\dots,\overline{[\alpha_{k-1}]}_j\right)$$

(2)キーワードを保存するユーザは，以下の 1 の分散値集合を生成し，サーバに分散する．

$$[1]_j^{\delta}=\left(\overline{[\delta]}_j,\overline{[\delta_0]}_j,\dots,\overline{[\delta_{k-1}]}_j\right)$$

(3)(2)と同様に，検索をするユーザは，以下の 1 の分散値集合を生成し，サーバに分散する．

$$[1]_j^{\eta}=\left(\overline{[\eta]}_j,\overline{[\eta_0]}_j,\dots,\overline{[\eta_{k-1}]}_j\right)$$

<検索>

(1)ユーザは，検索キーワード$K'=\left(0,K'_1,\dots,K'_{n-1}\right)$(0 を除く各成分$K'_i$を部分検索キーワードとする)を，以下のように分散する．

$$[K']_j=\left(\overline{[\varepsilon K']}_j,\overline{[\varepsilon_0]}_j,\dots,\overline{[\varepsilon_{k-1}]}_j\right)$$

(2)サーバP_j(j∈GF(k))は$[K]_j$と$[1]_j^{\eta}$に関して秘匿乗算い(一時的にβKが復元される)を行い，以下に示される$[K]_j^{\eta}$を生成する．

$$[K]_j^{\eta}=\left(\overline{[\beta\eta K]}_j,\overline{[\beta_0\eta_0]}_j,\dots,\overline{[\beta_{k-1}\eta_{k-1}]}_j\right)$$

(4)同様にサーバP_j(j∈GF(k))は$[K']_j$と$[1]_j^{\delta}$に関して秘匿乗算(一時的にεK'が復元される)を行い，以下に示される$[K']_j^{\delta}$を生成する．

$$[K']_j^{\delta}=\left(\overline{[\beta\delta S']}_j,\overline{[\beta_0\delta_0]}_j,\dots,\overline{[\beta_{k-1}\delta_{k-1}]}_j\right)$$

(2)サーバ P_j(j∈GF(k))は$\overline{[\beta\eta K]}_j$と$\overline{[\varepsilon\delta K']}_j$を 3.5.4 に示された TUS 法を適用した秘匿減算を行い，以下を得る．このときサーバ P_j によって生成される乱数δ_j^tは，この処理を行うたびに生成される．

$$[D]_j^t=\left(\overline{\left[\lambda^t(K-K')\right]}_j,\overline{[\lambda_0^t]}_j,\dots,\overline{[\lambda_{k-1}^t]}_j\right)$$

(3)サーバP_jは$\overline{[\alpha S]}_j$と$\overline{\left[\lambda^t(K-K')\right]}_j$を 3.4.1 に示された提案方式の加法準同型性を利用した秘匿加算を行い，以下をユーザに送信する．

$$[R]_j^t=\left(\overline{\left[\alpha S+\lambda^t(K-K')\right]}_j,\overline{[\alpha_0]}_j,\dots,\overline{[\alpha_{k-1}]}_j\right)$$

(4)ユーザはαS+λ^t(K-K')を復元し，同じく復元したαで除する．

$$S+\frac{\lambda^t}{\alpha}(K-K')$$

部分検索キーワードK'_iが部分キーワードK_iと一致したとき，$\frac{\lambda^t}{\alpha}(K_i-K'_i)=0$となり，キーワードに関連する秘密情報$S_i$が復元できる．キーワードの並び順を考慮する場合は，検索キーワードの順番を入れ替えて同様の手順を繰り返す．

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 XOR 法を応用した，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方式である多値化法について，その加法準同型性を用いた秘匿検索(部分一致)への応用の一例や，TUS 法を適用した四則演算の手法を提案した．

参考文献

[1] 鈴木良介: “ビッグデータビジネスの時代”，翔泳社, pp.14 (2011).
[2] 総務省: “平成 24 年版情報通信白書”, pp.153 (2012).
[3] 岡本栄司: “暗号理論入門[第 2 版]”，共立出版株式会社, pp. 91-95 (2006).
[4] A. Shamir: “How to Share a Secret”, Commun.ACM,vol.22, no.11, pp.612-613(1979).
[5] Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka: “A New (k, n)-Threshold Secret Sharing Scheme and Its Extension”, ISC 2008 conference, (2008).
[6] 高橋加寿子, 須賀祐治, 岩村惠市: “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”，第 65 回 CSEC 研究会 (2014).
[7] 神宮武志, 岩村惠市: “除算を含む四則演算に適用可能な秘密分散法を用いた秘匿計算手法の提案”，信学技報 115(122), pp.51-57(2015).
[8] Joel H. Ferziger, Milovan Perić: “コンピュータによる流体力学(小林敏雄,谷口伸行,坪倉誠訳)”，シュプリンガー・フェアラーク東京, pp.88 (2003).
[9] 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(2008).
[10] 青井健, 神宮武志, 岩村惠市:“n<2k-1 における秘匿計算の安全性検討及び非対称秘密分散との応用”，信学技報 116(130), pp.237-243(2016).
[11] 中原将貴, 辻下健太郎, 金田北洋, 岩村惠市: “n<2k-1 において実行可能なパスワード付秘密分散法及びその秘匿検索への応用”, CSS2016(2016).