計算主体を限定しない汎用的で軽量な秘匿関数計算の提案

布川 敦史† 須賀 祐治‡ 岩村 恵市†

†東京理科大学
102-0073 東京都千代田区九段北 1-14-6
nunokawa @sec.ee.kagu.tus.ac.jp, iwamura@ee.kagu.tus.ac.jp
‡株式会社インターネットイニシアティブ
101-0051 東京都千代田区神田神保町 1-105
suga@iij.ad.jp

あらまし 秘匿関数計算は入力情報を秘匿して計算が行えるので情報保護の観点から、データマイニングや電子投票などにおいてプライバシ情報を保護するために利用される。秘匿関数計算にはいくつかの種類があるが，低計算量かつ、汎用性がある方式としては千田らの方式がある。この方式は分散計算をする主体数が3つに限定され、復元処理はそのうちの2主体が行う。本稿では素因数分解の計算量的困難性を用いることにより、分散できる主体数を3つ以上に拡張した方式を提案する。

Proposal of general and light Secure Function Evaluation that is Arbitrary number calculation entity

Atsushi Nunokawa† Yuji Suga‡ Keiichi Iwamura†

†Tokyo University of Science
1-14-6 Kudankita, Chiyoda, Tokyo 102-0073, JAPAN
{nunokawa, iwamura}@sec.ee.kagu.tus.ac.jp
‡Internet Initiative Japan Inc.
1-105 Kandajinbo-cho 101-0051 JAPAN
suga@iij.ad.jp

Abstract Secure Function Evaluation is used to protect privacy information like data mining and electronic voting, etc. Secure Function Evaluation that has low computation and general-purpose is Chida's method. Data is corrected from arbitrary two entities in three calculation entity. Therefore, addition, subtraction, constant multiplication, and multiplication are possible. In this paper, it proposes method to increase the number of entity to three or more by using computational complexity of prime decomposition.

1 はじめに

近年、インターネットの普及によりデータの活用が増えている。なかでも代表される技術としてデータマイニング(data mining)がある。データマイニングとは大量のデータ(観測データ、購買履歴など)から法則や有益な情報を導く手法である。

データマイニングは今まで個人情報などのセンシティブな情報をそのまま用いることが主流であった。しかし、これでは個人情報の観点からよくないとして、プライバシ

保護データマイニング[3](以降、PPDM: Privacy Preserving Data Mining)が提案された。PPDM においてセンシティブな情報を秘匿しながらデータマイニングを行う方式として秘匿関数計算(Secure Function Evaluation)がある。

秘匿関数計算の既存の研究では準同型公開鍵暗号[2]を用いる方式や、CIC48 方式[4]などがある。一般に、準同型公開鍵暗号を用いる方式は計算量が多い。

それに対して CIC48 方式は今まで提案された手法の中で最も軽量な手法である。

CIC48 方式の特徴は情報量的安全性に基づいた手法であるが、計算主体は 3 主体に限定され、これらが協調演算を行うことで、データを秘匿し分散する。秘匿されたデータは計算主体の任意の 2 つから集めることで加減算、定数倍、乗算を行える。しかし、2 つの計算主体に結託された場合には弱いという欠点がある。

本稿ではより汎用で軽量的な秘匿関数計算方式を提案する。CIC48 方式との相違は主体数を限定せず汎用性を実現することができ、任意の n 主体の協調演算により入力を秘匿し、k 主体のデータを集めることで加減算、定数倍、乗算の計算が実現できる。ただし、安全性は素因数分解の計算量的困難性に依存する。

以降、2 節では関連研究、3 節では提案方式、4 節では証明、5 節では復元法の汎用化、6 節ではまとめという流れで説明する。

2 関連研究

2.1 準同型公開鍵暗号

準同型公開鍵暗号[2]は秘匿関数計算の一つで、データを暗号化し、暗号文のまま計算ができる方式。特徴として、平文の情報、鍵を知らずに計算をすることができる。例として Paillier 暗号について簡単に書く。

Paillier 暗号は、平文 m_1,m_2 を暗号化した二つの暗号文 c_1,c_2 の積を取ることにより、m_1+m_2 の暗号文が計算できる暗号方式である。

$c_1=g^{m1}\cdot r_1^n \bmod n^2$　$c_2=g^{m2}\cdot r_2^n \bmod n^2$

c_1,c_2 二つの積を取る。

$c_1\cdot c_2=g^{m1+m2}\cdot(r_1\cdot r_2)^n$

m_1+m_2 の加算になることが確かめられる。

計算は加減算もしくは乗算が可能であるが、暗号文での平文の改竄ができ、また、べき乗の計算を必要とするので、データマイニングに適用するには難しいと考えられる。

2.2 CIC48 方式

CIC48 方式[4]は 2010 年 3 月の CSEC で NTT の千田らによって提案された秘匿関数計算である。

3 主体に情報を分散する。分散した情報を用いて演算を行い、復元データを作る。復元時は任意の 2 つの主体からデータを集める。加減算、定数倍、乗算が提案され、これらを用いて、論理回路演算、2 進数整数変換の提案もしている。特徴として情報量的安全性を持つが 3 主体に限定されているので汎用的でなく、かつ 2 主体の結託により情報が漏れる危険性がある。

3 提案方式

秘匿、復元処理、乗算プロトコルの紹介をする。加減算、定数倍は CIC48 方式と同様の処理で実現できるのでここでは割愛する。

Semi-honest モデルで行い、また、分散する数は $n\geqq 3$ の自然数とし、乱数は一様乱数、素数は大きな素数とする。計算は Z/pZ 上で行われるものとする。

3.1 秘匿処理、復元処理

―秘匿処理―

情報提供者 A、計算主体 E_{n-1} とし情報 $a\in Z/pZ$ の秘匿を行う。

A:素数 $a_k \in Z/pZ$ $(1 \leqq k < n)$を生成し、以下の計算を行う。

$$a_0 = a - a_1 - a_2 - \cdots - a_{n-1}$$

A は計算主体主 E に分散情報を提供する。

$E_0 : D_0 = a_0$

$E_1 : D_1 = a_1$

⋮

$E_{n-1} : D_{n-1} = a_{n-1}$

—復元処理—

復元者 B として計算主体から集めたデータで復元を行う。

B:$D_0 + D_1 + \cdots + D_{n-1} = a_0 + a_1 + \cdots + a_{n-1}$

$a_0 = a - a_1 - a_2 - \cdots - a_{n-1}$より

$= a$

3.2 乗算

情報提供者 A1,A2 とし、それぞれが計算する値を $a,b \in Z/pZ$ とする。

－分散－

A1:素数 $a_k \in Z/pZ$ $(1 \leqq k < n)$を生成し、以下の計算を行う。

$$a_0 = a - a_1 - a_2 - \cdots - a_{n-1}$$

また、協調演算で用いる素数 $p_k \in Z/pZ$ $(1 \leqq k < n)$を生成する。また、乱数 $r_{ajk} \in Z/pZ$ を生成する。j=,2,･･･,n-1,のとき k=1,2,･･･,j-1 ただし、j=0 のとき k=1,2,･･･,n-1 とし、j＝1 のときは生成しない。

これらの素数と乱数を用いて R1 の値を作る。

$R1=(r_{a01}+\sum_{m=2}^{n-1} r_{am1})/p_1+(r_{a02}+\sum_{m=3}^{n-1} r_{am2})/p_2+$･･･$+(r_{a0k}+\sum_{m=k+1}^{n-1} r_{amk})/p_k+$･･･$+r_{a0n-1}/p_{n-1}$

A2:素数 $b_k \in Z/pZ$ $(1 \leqq k < n)$を生成し、以下の計算を行う。

$$b_0 = b - b_1 - b_2 - \cdots - b_{n-1}$$

また、協調演算で用いる素数 $q_k \in Z/pZ$ $(1 \leqq k < n)$を生成する。また、乱数 $r_{bjk} \in Z/pZ$ を生成する。j=,2,･･･,n-1,のとき k=1,2,･･･,j-1 ただし、j=0 のとき k=1,2,･･･,n-1 とし、j＝1 のときは生成しない。

これらの素数と乱数を用いて R2 の値を作る。

$R2=(r_{b01}+\sum_{m=2}^{n-1} r_{bm1})/q_1+(r_{b02}+\sum_{m=3}^{n-1} r_{bm2})/q_2+$･･･$+(r_{b0k}+\sum_{m=k+1}^{n-1} r_{bmk})/q_k+$･･･$+r_{b0n-1}/q_{n-1}$

A1,A2 は計算主体に分散情報、乱数、素数、生成した値を渡す。

$E_0 : D_0 = (a_0, b_0, R1, R2)$

$E_1 : D_1 = (a_1, b_1\ p1, q1)$

$E_2 : D_2 = (a_2, b_2\ p2, q2, r_{a21}, r_{b21})$

⋮

$E_k : D_k = (a_k, b_k, p_k, q_k, r_{ak1}, r_{ak2}, \cdots, r_{akk-1}, r_{bk1}, r_{bk2}, \cdots, r_{bkk-1})$

⋮

$E_{n-1} : D_{n-1} = (a_{n-1}, b_{n-1}, p_{n-1}, q_{n-1}, r_{an-11}, r_{an-12}, \cdots, r_{an-1n-2}, r_{bn-11}, r_{bn-12}, \cdots, r_{bn-1n-2})$

—協調演算—

計算は全て計算主体で行われる。

・主体 E_1 は以下の計算を行う。

$E_1 : N_1 = a_1 p_1$

E_1 以外の主体 E_j(j=0,2,3,…,n-1)にN_1を送る。

E_j(j=0,2,3,…,n-1)は以下の計算をする。

$E_j : X_{j \Leftrightarrow 1} = N_1 b_j + r_{aj1} = a_1 b_j p_1 + r_{aj1}$

E_j は $X_{j \Leftrightarrow 1}$を E_1 に送り、E_1 は p_1 で割る。

$$E_1 : \frac{X_{j \Leftrightarrow 1}}{p_1} = \frac{a_1 b_j p_1 + r_{aj1}}{p_1} = a_1 b_j + \frac{r_{aj1}}{p_1}$$

$E_0, E_2 \sim E_{n-1}$ は同様の計算をし、得た値で X'_1 を計算する。

$$X'_1 = a_1 b_1 + \frac{X_{0 \Leftrightarrow 1}}{p_1} + \frac{X_{2 \Leftrightarrow 1}}{p_1} + \cdots + \frac{X_{n-1 \Leftrightarrow 1}}{p_1}$$

$$= a_1 b_1 + a_1 \left(b_0 + \sum_{m=2}^{n-1} b_m \right) + \frac{1}{p_1} \left(r_{a01} + \sum_{m=2}^{n-1} r_{am1} \right)$$

$E_1 : M_1 = b_1 q_1$

E_1 以外の主体 E_j(j=0,2,3,…,n-1)にM_1を送る。

E_j(j=0,2,3,…,n-1)は以下の計算をする。

$E_j : x_{j \Leftrightarrow 1} = M_1 a_j + r_{bj1} = a_j b_1 q_1 + r_{bj1}$

E_j は $x_{j \Leftrightarrow 1}$を E_1 に送り、E_1 は q_1 で割る。

$$E_1 : \frac{x_{j \Leftrightarrow 1}}{q_1} = \frac{a_1 b_j q_1 + r_{bj1}}{q_1} = a_1 b_j + \frac{r_{bj1}}{q_1}$$

$E_0, E_2 \sim E_{n-1}$ は同様の計算をし、得た値で X''_1 を計算する。

$$X''_1 = \frac{x_{0 \Leftrightarrow 1}}{q_1} + \frac{x_{2 \Leftrightarrow 1}}{q_1} + \cdots + \frac{x_{n-1 \Leftrightarrow 1}}{q_1}$$

$$= b_1 \left(a_0 + \sum_{m=2}^{n-1} a_m \right) + \frac{1}{q_1} \left(r_{b01} + \sum_{m=2}^{n-1} r_{bm1} \right)$$

以上の計算から $D_0=X'_1+X''_1$ としておく。
・任意の主体 E_k は(k≧2)以下の計算を行う。
$E_k: N_k = a_k p_k$
E_k は主体 E_j(j=0,k+1,⋯,n-1)にN_kを送る。E_j(j=0,k+1,⋯,n-1)は以下の計算をする。
$E_j: X_{j\Leftrightarrow k} = N_k b_j + r_{ajk} = a_k b_j p_k + r_{ajk}$
E_j は $X_{k\Leftrightarrow j}$を E_k に送り、E_k は p_k で割る。

$$E_k: \frac{X_{j\Leftrightarrow k}}{p_k} = \frac{a_k b_j p_k + r_{ajk}}{p_k} = a_k b_j + \frac{r_{ajk}}{p_k}$$

E_0,E_2～E_{n-1} は同様の計算をし、得た値で X'_1 を計算する。

$$X'_k = a_k b_k + \frac{X_{0\Leftrightarrow k}}{p_k} + \frac{X_{k+1\Leftrightarrow k}}{p_k} + \cdots + \frac{X_{n-1\Leftrightarrow k}}{p_k}$$

$$= a_k b_k + a_k\left(b_0 + \sum_{m=k+1}^{n-1} b_m\right) + \frac{1}{p_k}\left(r_{a0k} + \sum_{m=k+1}^{n-1} r_{amk}\right)$$

$E_k: M_k = b_k q_k$
E_k は主体 E_j(j=0,k+1,k+2,⋯,n-1)にM_kを送る。E_j(j=0,k+1,k+2,⋯,n-1)は以下の計算をする。
$E_j: x_{j\Leftrightarrow k} = M_k a_j + r_{bjk} = a_j b_k q_k + r_{bjk}$
E_j は $x_{j\Leftrightarrow k}$を E_k に送り、E_k は q_k で割る。

$$E_k: \frac{x_{j\Leftrightarrow k}}{q_k} = \frac{a_j b_k q_k + r_{bjk}}{q_k} = a_j b_k + \frac{r_{bjk}}{q_k}$$

E_0,E_{k+1}～E_{n-1} は同様の計算をし、得た値で X''_k を計算する。

$$X''_k = \frac{X_{k+1\Leftrightarrow k}}{q_k} + \frac{X_{k+2\Leftrightarrow k}}{q_k} + \cdots + \frac{X_{n-1\Leftrightarrow k}}{q_k}$$

$$= b_k\left(a_0 + \sum_{m=k+1}^{n-1} a_m\right) + \frac{1}{q_k}\left(r_{b0k} + \sum_{m=k+1}^{n-1} r_{bmk}\right)$$

以上の計算から $D_k=X'_k+X''_k$ としておく。
・E_0 は以下の計算を行う。
$E_0: D_0 = a_0 b_0 - R1 - R2$
D_0 を保持する。

—復元処理—

情報を集計する B はすべての主体から E_k (k=0,1,⋯,n-1)D_k(k=0,1,⋯,n-1)の値を集め,すべてを足す。

$B: D_0+D_1+\cdots+D_k+\cdots+D_{n-1}$

$= a_0b_0-R1-R2+a_1b_1 + a_1(b_0 + \sum_{m=2}^{n-1} b_m) +$

$\frac{1}{p_1}(r_{a01} + \sum_{m=2}^{n-1} r_{am1})+b_1(a_0 + \sum_{m=2}^{n-1} a_m) +$

$\frac{1}{q_1}(r_{b01} + \sum_{m=2}^{n-1} r_{bm1})+\cdots+a_k b_k +$

$a_k(b_0 + \sum_{m=k+1}^{n-1} b_{mk}) +$

$\frac{1}{p_k}(r_{a0k} + \sum_{m=k+1}^{n-1} r_{amk}) + b_k(a_0 +$

$\sum_{m=k+1}^{n-1} a_m) + \frac{1}{q_k}(r_{b0k} + \sum_{m=k+1}^{n-1} r_{bmk})$

$+\cdots+a_{n-1}b_{n-1}+a_{n-1}b_0+a_0b_{n-1}+ \frac{1}{p_{n-1}} r_{a0n-1}+$

$\frac{1}{q_{n-1}} r_{b0n-1}$

$R1=(r_{a01}+\sum_{m=2}^{n-1} r_{am1})/p_1+(r_{a02}+\sum_{m=3}^{n-1} r_{am2})/p_2+\cdots+(r_{a0k}+\sum_{m=k+1}^{n-1} r_{amk})/p_k+\cdots+r_{a0n-1}/p_{n-1}$

$R2=(r_{b01}+\sum_{m=2}^{n-1} r_{bm1})/q_1+(r_{b02}+\sum_{m=3}^{n-1} r_{bm2})/q_2+\cdots+(r_{b0k}+\sum_{m=k+1}^{n-1} r_{bmk})/q_k+\cdots+r_{b0n-1}/q_{n-1}$

R1,R2 の 2 式より乱数が取り除かれる。

$=a_0b_0+a_1b_1 + a_1(b_0 + \sum_{m=2}^{n-1} b_m)+b_1(a_0 + \sum_{m=2}^{n-1} a_m)+\cdots$

$+a_k b_k + a_k(b_0 + \sum_{m=k+1}^{n-1} b_m) + b_k(a_0 + \sum_{m=k+1}^{n-1} a_m)$

$+\cdots+a_{n-1}b_{n-1}+a_{n-1}b_0+a_0b_{n-1}$

a_i(i=0,1,2,⋯,n-1)でくくる

$=a_0(b_0+b_1+\cdots+b_k+\cdots+b_{n-1})+ a_1(b_0+b_1+\cdots+b_k+\cdots+b_{n-1})+\cdots+ a_k(b_0+b_1+\cdots+b_k+\cdots+b_{n-1})+\cdots+ a_{n-1}(b_0+b_1+\cdots+b_k+\cdots+b_{n-1})$

$=(a_0+a_1+\cdots+a_k+\cdots+a_{n-1})\ (b_0+b_1+\cdots+b_k+\cdots+b_{n-1})$

$a_0=a-(a_1+\cdots+a_k+\cdots+a_{n-1})$, $b_0=b-(b_1+\cdots+b_k+\cdots+b_{n-1})$より

$=ab$

4 証明

乗算プロトコルの安全性についての証明を行う。最初に Semi-honest モデルの完全秘匿性し定義を示してから証明を行う。

最初に E_k の計算主体の入力と出力を整理し、そのデータから予想される逸脱した計算について考える。

計算主体を n 個とし任意の計算主体 E_k の入力と出力を考え、入力 X に対して F(X)の計算をする。ユーザは入力 X を以下のように分散する。

$X=(X_{E1},X_{E2},\cdots,X_{En})$
ユーザが分散させた入力を主体に送る。
また、X_{Ek}と一緒にユーザは計算主体E_kにR_{Ek}を送る。
$R_{Ek}=\{R_{k1},R_{k2},\cdots,R_{kj}\}$
R_{Ek} の例として元情報の値ではなく、乱数や素数などの演算に用いる値とする。
R_{Ek}から任意に選んだものをR'_{Ek}とする。
$R'_{Ek}\subseteq\{R_{k1},R_{k2},\cdots,R_{kj}\}$
任意の主体 E_k が他の計算主体から得る値をw_kとする。
$w_k=\{w_{k1},w_{k2},\cdots w_{kj}\}$
w_kの中から任意に選んだものをw'_kとする。
$w'_k\subseteq\{w_{k1},w_{k2},\cdots w_{kj}\}$
計算主体E_kはユーザと他の計算主体から受け取った値$X_{Ek,}$ w'_k, R'_{Ek}で決められたプロトコルfの計算し、計算結果をf_jとする。
$f(X_{Ek,}\ w'_k,\ R'_{Ek})=\{f_1,f_2,\cdots,f_j\}$
また、逸脱した計算をgとし計算結果をg_jとする。
受け取った値 $X_{Ek,}$ w'_k, R'_{Ek} で元情報を導こうとするものとする。
$g(X_{Ek,}\ w'_k,\ R'_{Ek})=\{g_1,g_2,\cdots,g_j\}$
逸脱した計算は素因数分解や乱数を取り除く計算など元情報を導く計算とする。
以上より定義 4.1 でVIEW関数の定義をする。

定義 4.1　VIEW 関数の定義

ここでは VIEW 関数が確率的関数であることを示し、Semi-honest モデルでの完全秘匿性を定義する。また、VIEW 関数が情報量的安全と確率的関数の 2 つの性質を持つなら十分な秘匿性を持つことを定義する。
計算主体E_kの入力$X_{Ek,}$ w_k, R_{Ek}よりfとgが計算する。
Semi-honest モデルで完全秘匿性を持つとき、以下のような確率的関数 S が存在することである。
$S(X_{Ek},\ w'_k,R'_{Ek},\ f(X_{Ek,}\ w'_k,\ R'_{Ek}),\ g(X_{Ek,}\ w'_k,\ R'_{Ek}))=VIEW_k(X_{Ek},\ w'_k,R'_{Ek},)$
上記で完全秘匿性を定義したが、素因数分解など計算量的困難性に基づいた安全性があるとき完全秘匿性を持たないが、計算量的安全性をもつので十分な秘匿性をもつとする。

定義 4.2　素因数分解について

ある素数$p_1,q_1\in Z/pZ$の2つの合成数は多項式時間で素因数分解 が出来る。このとき大きな素数を用いることで素因数分解を困難にすることが出来る。 このとき合成数 p_1q_1 は計算量的安全性をもつとする。

定義 4.3　関数について

確率変数 $s\in S$ と独立であるような集合 S から集合 D への確率的関数 F が完全秘匿性を持つとは、任意の実際の秘密データ $a\in S$ と出力 $d\in D$ に対して 以下の式が成り立つ。

$$Pr(s=a\mid F(s)=d)=Pr(s=a)$$

s:元データ,F(s):開示されるデータ

定義 4.4　配布データについて

ユーザから送られる元データは n 個(n>2)の素数を引いて秘匿される。このとき情報量的安全性があると仮定して、元情報については全ての素数と素数を元データから引いた値が集まらないと復元できないとする。

定義 4.5　一様乱数について

一様な乱数を足された値は一様乱数とする。

Proof　このプロトコルが通信路、各計算主体において情報量的安全性と計算量的安全性を持つことを確かめ、定義で示した Semi-honest モデルにおいて十分な秘匿性をもつことを確かめる。

通信路で得られる値は計算主体で交換される$X_{j\leftrightarrow k}$とN_k,M_kが得られる。
$X_{j\leftrightarrow k}$ は一様乱数が足され定義 4.5 より一様乱数とみなせるので情報量的安全性をもち、N_k,M_k は二つの素数の合成数であるので計算量的安全性を持つ。
計算主体E_0では元情報が得られるのはユーザから素数を n 個引いた元データa_0,b_0と任意の計算主体$E_j(1\leqq j\leqq n\text{-}1)$から得る$N_j$、$M_j$の 2 つの素数の合成数である。
a_0,b_0 は定義 4.4 より情報量的安全性を持ち、

N_j、M_j は同様に計算量的安全性を持つ。
計算主体 E_1 で他の計算主体 $E_j(2 \leqq j \leqq n-1)$から得るのは $X_{j \Leftrightarrow 1}$ である。$X_{j \Leftrightarrow 1}$ は一様乱数が足されるので定義 4.5 より情報量的安全性をもつ。
計算主体 $E_k(1 < k \leqq n-1)$では $X_{j \Leftrightarrow k}(k+1 \leqq j \leqq n-1)$と $N_j, M_j(1 \leqq j \leqq k-1)$を得る。$X_{j \Leftrightarrow k}$ は一様乱数が足されるので情報量的安全性を持ち、N_j, M_j は 2 つの素数の合成数なので計算量的安全性をもつ。
以上より、通信路と計算主体 $E_k(0 \leqq k \leqq n-1)$が情報量的安全性と計算量的安全性に基づく値を受け取るのでこのプロトコルは Semi-honest モデルにおいて十分な秘匿性を持つ。

5　復元法の汎用化

前章は 4 主体中 4 主体が情報復元に必要であった。そこで、n 主体中 k 主体が集まれば情報復元が可能な復元法の汎用化を考える。今回はページ数の都合で原理や証明、一般化に関する説明は省略し、計算主体が 4 つの場合の例で説明する。4 主体中 1 主体で情報復元できるためには各主体 E_k は自分がもつ D_k を他の全主体に送ればよい。すると、各主体は D_1～D_k の情報がそろうため 1 主体で情報復元可能である。また、4 主体中 2 主体で情報復元可能にするためには各主体が各々異なる 1 つのデータを持たないようにすればよい。すなわち、E_1 は E_2 と E_3 に D_1 を送り、E_2 は E_3 と E_4 に D_2 を送り、E_3 は E_4 と E_1 に D_3 を送り、E_4 は E_1 と E_2 に D_4 を送る。すると、E_1 は D_1, D_4, D_3 を、E_2 は D_2, D_1, D_4 を、E_3 は D_3, D_2, D_1 を、E_4 は D_4, D_3, D_2 を得る。前章の方式は情報復元に 4 主体のデータ D_1～D_4 が必ず必要なため、各主体は自分だけでは情報復元できず、任意の 2 主体が集まれば情報復元可能になる。同様の原理では、4 主体中 3 主体で情報復元を可能にするためには各主体が他の 1 主体にデータを送り、各主体が 2 つデータを持つようにすればよいように思えるが、場合によっては 4 主体中 2 主体が集まれば情報復元できる。よって、それを避けるために、誤り訂正符号または秘密分散を用いる。誤り訂正符号の場合、(12, 8) 符号を構成し、各主体 E_k が情報となる D_k を 2 分割して 2 シンボルとし、他にパリティ P_k を1つもつように構成する。すると、4 主体中 3 主体が集まれば 9 個のシンボルがそろうので、消失訂正をすることにより集めなかった 1 主体のデータが復元でき、その後情報復元を行う。パリティの計算法も秘匿計算によって行うがページ数の都合上詳細は次回の発表に回す。

6　まとめ

本稿では秘匿関数計算における乗算プロトコルの提案と証明を行った。提案プロトコルは計算量的安全性と情報量的安全性にもとづくプロトコルであることを確認した。今後は復元法の汎用化を具体的に示し、さらに実装によりここで取り上げた 3 つの方式の計算量や処理速度の比較を正確に行いたいと思う。

7　参考文献

[1]Agrawal, R. and Srikant, R.: Privacy-preserving data mining, ACM SIGMOD Record, Vol. 29, No. 2, pp. 439–450 (2000)
[2]Public-Key Cryptosystems Based on Composite Degree Residuosity Classes Pascal Paillie Published in J. Stern, Ed., Advances in Cryptology EUROCRYPT '99, vol. 1592 of Lecture Notes in Computer Science, pp. 223-238 Springer-Verlag, 1999
[3]佐久間 淳 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)
[4]千田 浩司,五十嵐 大 効率的な 3 パーティ秘匿関数計算の提案とその運用モデルの考察 Vol.2010-CESC-48 No.1
[5]高効率 3 パーティ秘匿関数計算の情報理論的安全性, / 五十嵐 大, 千田浩司, 高橋克己 Vol.2010-CSEC-50 No.46