Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

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

千田浩司[†1]　五十嵐大[†1]　高橋克巳[†1]

3主体の協調計算により，入力値を秘匿しつつ効率良く関数計算を行う方式を提案する．提案方式は，2主体の取得情報を得ない限り入力値を秘匿できる特徴を持つ．また，提案方式をプライバシ保護データマイニング（Privacy-Preserving Data Mining）技術として利用する場合の運用モデルの考察を行う．

Efficient 3-Party Secure Function Evaluation and Its Application

Koji Chida,[†1] Dai Ikarashi[†1]
and Katsumi Takahashi[†1]

We propose a simple three-party protocol that efficiently evaluates a function keeping input data secret. The proposed protocol can obtain the result of function without disclosing input data as long as no party conspires with any other party. Moreover, we consider *Privacy-Preserving Data Mining* as an application of the proposed protocol.

1. はじめに

近年，個人に関する様々な情報を容易に取得できる環境の進歩が目覚ましいが，個人に関する情報の利活用については，個人情報保護・プライバシの観点から，国内外で法制度・ガイドラインや標準化の整備と併せた技術的対策が検討されている[1)–5)]．そして技術的対策に注目すると，いわゆる広義の匿名化が有効な手段として検討が進められている．ここで広義の匿名化とは，個人に関する情報が個人と結び付くことの無いように情報を加工する手段全般を指す．しかし一般に匿名化による対策は，情報の利用が限定的になることに加え，個人に関する情報が個人と結び付かないことの保証が容易では無い．その理由の一つとして，個人に関する情報と個人の結び付けが，予め備えている知識や情報に大きく左右されることが挙げられる．例えば，氏名，住所，年齢，性別，職業，購買履歴（日時，場所，商品名）からなるデータをマーケティング用途に氏名，住所を ID 番号に置き換えて第三者提供や一般公開を行った場合，年齢，性別，職業，および一部の購買履歴から ID 番号に対応する個人を特定できる者がいないとも限らず，その ID 番号から特定の個人の購買履歴が全て紐付いてしまう可能性があり，これはプライバシの観点から望ましくない．このように，個人に関する情報をどの程度まで開示可能かという問題は，情報の利活用と保護の両方の観点から慎重に考える必要がある．

上述の開示問題の有力な解決手段として，各種計算の入力となるデータを秘匿しつつ計算結果を求める**秘匿関数計算**（Secure Function Evaluation）[21)] が近年注目されている．特に情報を保護しつつデータマイニングを行う**プライバシ保護データマイニング**（Privacy-Preserving Data Mining）技術は，Lindell, Pinkas による研究成果[15)] を端緒に秘匿関数計算を利用した手法が数多く提案されている[6),20)]．しかしながら秘匿関数計算は一般に通常の関数計算と比べ処理時間が著しく増加し，特にデータマイニングは入力データ数や計算量が膨大となる場合があるため，秘匿関数計算を利用する場合は処理時間の軽減が特に大きな課題となる．なお本稿では，情報を保護しつつ統計処理を行うための技術も合わせてプライバシ保護データマイニングと呼ぶことにする[⋆1]

本稿では，プライバシ保護データマイニング技術として利用可能とすることを目指した，軽量かつ汎用型の秘匿関数計算方式を提案する．対象とする計算を限定して処理軽減を図る方式は数多く提案されているものの，代表的な演算への適用に留まっているのが現状である．提案方式は，3主体の協調計算により入力値を秘匿しつつ各種計算を効率良く実行し，2主体の取得情報を得ない限り入力値を秘匿できる特徴を持つ．また計算結果を復元する前に不正の有無を検証できるため，単独の主体が不正を働いたとしても入力値の秘匿性は保証される．

†1 NTT 情報流通プラットフォーム研究所
NTT Information Sharing Platform Laboratories

⋆1 最近ではより広範の技術を指す，プライバシ保護データ分析（Privacy-Preserving Data Analysis）やプライバシ保護データ活用（Privacy-Preserving Data Utilization）といった用語も見受けられる．

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

以降，2 節で関連研究を紹介し，3 節で提案方式について説明する．4 節では提案方式に適した運用モデルについて考察する．最後に 5 節で本稿をまとめる．

2. 関 連 研 究

秘匿関数計算は一般に，計算対象の入力値を秘匿処理したうえで提供する主体と，その入力値を復元すること無く処理する主体の少なくとも 2 主体が関与し，秘匿方法は暗号化や秘密分散が代表的である．具体的な実現方法は Yao によって提案された[22]．これは論理回路演算を実行可能な状態のまま秘匿する主体と，その秘匿された論理回路演算を実行する主体によって構成され，実行には複数主体の協調計算が必要なことから**マルチパーティプロトコル**とも呼ばれる (2 主体に特化した方式は区別して **2 パーティプロトコル**と呼ぶ場合が多い)．論理回路演算を実行するマルチパーティプロトコルとして，紛失通信（Oblivious Transfer）を利用した方式[13] や，MIX-net を利用した方式[14] 等が提案されている．なお論理回路演算の実行を可能とする秘匿関数計算は**秘匿回路計算** (Secure Circuit Evaluation) と呼ばれる．特に最近では秘匿回路計算の実行を単独で行うことを可能とする準同型暗号[12] が注目を集めている．また Yao の方式に基づく，2 パーティプロトコル[17] や計算委託型の 2 パーティプロトコル[23] 等の実装報告も見られるようになり，実用に向けた動きが加速しつつある．ここで計算委託型とは，入力提供主体と計算主体が異なる主体構成を指し，2 パーティであれば計算主体が 2 主体であることを意味する．

一方，環 $\mathbb{Z}/m\mathbb{Z}$（m は適当な整数）上の算術演算を可能とする秘匿関数計算も提案されており，秘密分散に基づく方式[8),9] や準同型暗号に基づく方式[10),18] が知られている．なお秘匿回路計算は一般に処理効率が悪いため，秘匿関数計算を論理回路演算と算術演算に分けて全体の処理効率を上げる手法も提案されている[7),11),16),19]．

3. 提 案 方 式

提案方式は，計算対象の入力値を秘匿しつつ計算結果を求めるために，3 主体 X, Y, Z の協調計算を必要とする．なお対象の計算は処理効率を上げるために論理回路演算と算術演算に分けられているものとする．

入力値の秘匿処理は 3.1 節のとおり単純な秘密分散となる．復元は 2 主体の分散データの加算により実現され，単独の主体の分散データでは復元できない．ただし分散データを入出力とした加減算や定数倍演算は，既存方式[8),9] 同様，各主体が単独で計算できる．乗算や論理回路演算については，既存方式[8),9] 同様，各主体の協調計算が必要となる．更に提案方式は，2 進数変換や整数変換についても他の計算の単純な拡張として実現でき，これにより算術演算および論理回路演算の組合せ計算が可能となる．また計算結果を復元する前に不正の有無を検証できるため，単独の主体が不正を働いたとしても入力値の秘匿性は保証される．

はじめに，全ての主体が正しく手続きを実行することを仮定した基本方式（semi-honest モデル）を説明する．次に，何れかの主体が不正処理により計算結果を改竄した場合の検出方式（optimistic モデル）を説明する．なお計算結果を改竄した主体の特定方式（malicious モデル）については，任意のゼロ知識証明技術を用いるものとして本稿では言及しない．

何れの方式も，2 主体の取得情報を得ることはできないと仮定する．言い換えれば各主体は他の主体と結託しないものとする．また計算は $\mathbb{Z}/m\mathbb{Z}$ 上で行われるものとし，計算の効率化のため $m = 2^\ell$（ℓ は適当な整数）として話を進める．

3.1 基 本 方 式

[秘匿処理]

$a \in \mathbb{Z}/m\mathbb{Z}$ を以下のように秘密分散し，X, Y, Z の入力とする．

X： $a_x = a_y + a_z$
Y： $\vec{a}_y = (\hat{a}, a_y)$
Z： $\vec{a}_z = (\hat{a}, a_z)$

ただし $a_y, a_z \in_R \mathbb{Z}/m\mathbb{Z}$, $\hat{a} = a - a_x$.

上記の秘匿処理はデータ提供主体が実行する．データ提供主体は X, Y, Z の何れかであっても良いし，外部の主体でも良い．具体的なモデルを 4 節で与える．手続きとしては乱数 a_y, a_z を生成してから a_x, $\hat{a}$ を計算して各主体に分散データを送信すれば良い．

a_x, $\vec{a}_y$, $\vec{a}_z$ は何れもその取り方から a と独立の乱数とみなせるため，X, Y, Z 単独では a を復元できない．

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

[復元処理]

以下の何れかにより a を復元する．

- X, Y: $a_x, \hat{a}$ を共有し，式 (1) より a を求める．
- Y, Z: a_y, a_z を共有し，式 (2) より a を求める．
- X, Z: $a_x, \hat{a}$ を共有し，式 (1) より a を求める．

$$a = a_x + \hat{a} \tag{1}$$
$$a = a_y + a_z + \hat{a} \tag{2}$$

式 (1), (2) から a を正しく復元できることは $\hat{a} = a - a_x$, $a_x = a_y + a_z$ より明らか．

[加減算]

X, Y, Z は a, b の分散データ (a_x, b_x), $(\vec{a}_y, \vec{b}_y)$, $(\vec{a}_z, \vec{b}_z)$ から $c = a \pm b$ の分散データ $c_x, \vec{c}_y, \vec{c}_z$ を求める．

X： $c_x = a_x \pm b_x$
Y： $\vec{c}_y = (\hat{c}, c_y) = (\hat{a} \pm \hat{b}, a_y \pm b_y)$
Z： $\vec{c}_z = (\hat{c}, c_z) = (\hat{a} \pm \hat{b}, a_z \pm b_z)$

以下が成り立つことから，式 (1), (2) より $c_x, \vec{c}_y, \vec{c}_z$ の何れか二つから c を復元できることが分かる．

$$c_x + \hat{c} = (a_x \pm b_x) + (\hat{a} \pm \hat{b}) = (a_x + \hat{a}) \pm (b_x + \hat{b}) = a \pm b$$
$$c_y + c_z = (a_y \pm b_y) + (a_z \pm b_z) = (a_y + a_z) \pm (b_y + b_z) = a_x \pm b_x = c_x$$

[定数倍演算]

X, Y, Z は a の分散データ $a_x, \vec{a}_y, \vec{a}_z$ および定数 $e \in \mathbb{Z}/m\mathbb{Z}$ から $c = ea$ の分散データ $c_x, \vec{c}_y, \vec{c}_z$ を求める．

X： $c_x = e(a_x \pm b_x)$
Y： $\vec{c}_y = (\hat{c}, c_y) = (e\hat{a}, ea_y)$
Z： $\vec{c}_z = (\hat{c}, c_z) = (e\hat{a}, ea_z)$

$c_x, \vec{c}_y, \vec{c}_z$ の何れか二つから c を復元できることは加減算同様に確認すれば良い．

[乗算]

X, Y, Z は a, b の分散データ (a_x, b_x), $(\vec{a}_y, \vec{b}_y)$, $(\vec{a}_z, \vec{b}_z)$ から $c = ab$ の分散データ $c_x, \vec{c}_y, \vec{c}_z$ を求める．

(1) X は以下を行う．
  (a) $r_1, r_2, r_3, r_4, c_y \in_R \mathbb{Z}/m\mathbb{Z}$ を生成する．
  (b) $c_x = a_x b_x - r_3 - r_4$ を計算して c の分散データとする．
  (c) $c_z = c_x - c_y$ を計算する．
  (d) Y, Z にそれぞれ (r_1, r_2, r_3, c_y), $(a_x - r_1, b_x - r_2, r_4, c_z)$ を送信する．

(2) Y, Z はそれぞれ以下を計算して共有する．

$$y = \hat{a}\hat{b} + \hat{a}r_2 + r_1\hat{b} + r_3$$
$$z = \hat{a}(b_x - r_2) + (a_x - r_1)\hat{b} + r_4$$

(3) Y, Z は $\hat{c} = y + z$ を計算してそれぞれ $\vec{c}_y = (\hat{c}, c_y)$, $\vec{c}_z = (\hat{c}, c_z)$ を c の分散データとする．

以下が成り立つことから，式 (1), (2) より $c_x, \vec{c}_y, \vec{c}_z$ の何れか二つから c を復元できることが分かる．

$$\hat{c} = y + z = \hat{a}\hat{b} + \hat{a}b_x + a_x\hat{b} + r_3 + r_4$$
$$c_y + c_z = c_x = a_x b_x - r_3 - r_4$$
$$c_x + \hat{c} = (a_x + \hat{a})(b_x + \hat{b}) = ab$$

a, b の秘匿性については以下がいえる．

- X: データを受信しないため，乗算実行により a, b に関する有意な情報を得ることはできない．
- Y: 受信データ r_1, r_2, r_3, c_y, z は何れもその取り方から a, b と独立の乱数とみなせるため，乗算実行により a, b に関する有意な情報を得ることはできない．
- Z: 受信データ $a_x - r_1, b_x - r_2, r_4, c_z, y$ はその取り方から a, b と独立の乱数とみなせるため，乗算実行により a, b に関する有意な情報を得ることはできない．

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

[論理回路演算]

- 否定：X, Y, Z は $a \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ の分散データ a_x, $\vec{a}_y$, $\vec{a}_z$ から $\bar{a} = 1 - a$ の分散データ $\bar{a}_x$, $\vec{\bar{a}}_y$, $\vec{\bar{a}}_z$ を求める．
  X： $\bar{a}_x = -a_x$
  Y： $\vec{\bar{a}}_y = (\hat{\bar{a}}, \bar{a}_y) = (1 - \hat{a}, -a_y)$
  Z： $\vec{\bar{a}}_z = (\hat{\bar{a}}, \bar{a}_z) = (1 - \hat{a}, -a_z)$
- 論理積：X, Y, Z は $a, b \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ の分散データ (a_x, b_x), $(\vec{a}_y, \vec{b}_y)$, $(\vec{a}_z, \vec{b}_z)$ から $c = a \wedge b = ab$ の分散データ c_x, $\vec{c}_y$, $\vec{c}_z$ を求める．すなわち乗算を実行すれば良い．
- 論理和：X, Y, Z は $a, b \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ の分散データ (a_x, b_x), $(\vec{a}_y, \vec{b}_y)$, $(\vec{a}_z, \vec{b}_z)$ から $c = a \vee b = a + b - ab$ の分散データ c_x, $\vec{c}_y$, $\vec{c}_z$ を求める．すなわち加減算および乗算を実行すれば良い．
- 排他的論理和：X, Y, Z は $a, b \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ の分散データ (a_x, b_x), $(\vec{a}_y, \vec{b}_y)$, $(\vec{a}_z, \vec{b}_z)$ から $c = a \oplus b = a + b - 2ab$ の分散データ c_x, $\vec{c}_y$, $\vec{c}_z$ を求める．すなわち加減算，定数倍演算および乗算を実行すれば良い．

否定演算について，以下が成り立つことから，式 (1), (2) より $\bar{a}_x$, $\vec{\bar{a}}_y$, $\vec{\bar{a}}_z$ の何れか二つから $\bar{a} = 1 - a$ を復元できることが分かる．

$$\hat{\bar{a}} = 1 - \hat{a} = 1 - (a - a_x)$$
$$\bar{a}_x + \hat{\bar{a}} = -a_x + (1 - (a - a_x)) = 1 - a$$
$$\bar{a}_y + \bar{a}_z = -a_y - a_z = -a_x = \bar{a}_x$$

[2 進変換処理]

X, Y, Z は $a \in \mathbb{Z}/m\mathbb{Z}$ の分散データ a_x, $\vec{a}_y$, $\vec{a}_z$ から $a[i] \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ $(i = 1, \ldots, \ell)$ の分散データ $a[i]_x$, $\vec{a}[i]_y$, $\vec{a}[i]_z$ を求める．ここで $a[i]$ は a の下位 i 番目のビットとする．

(1) X, Y はそれぞれ $a_x[i]$, $\hat{a}[i]$ $(i = 1, \ldots, \ell)$ の秘匿処理を実行する．

(2) X, Y, Z は式 (1) および以下の全加算回路演算式に基づき $a_x[i]$, $\hat{a}[i]$ の分散データから $a[i]$ の分散データを求める．

$$c_1 = 0$$
$$d_i = a_x[i] + \hat{a}[i] - 2a_x[i]\hat{a}[i]$$
$$\text{(sum) } a[i] = a_x[i] \oplus \hat{a}[i] \oplus c_i = d_i + c_i - 2d_ic_i \quad (3)$$
$$\text{(carry out) } c_{i+1} = (a_x[i] \wedge \hat{a}[i]) \vee (a_x[i] \wedge c_i) \vee (\hat{a}[i] \wedge c_i)$$
$$= a_x[i]\hat{a}[i] + d_ic_i - a_x[i]\hat{a}[i]d_ic_i \quad (4)$$

式 (3), (4) は単純な $a_x + \hat{a}(= a)$ の論理式であり，$a_x[i]\hat{a}[i]$, d_ic_i, $(a_x[i]\hat{a}[i])(d_ic_i)$ の乗算を実行する．全体では加減算および乗算がともに $3\ell - 2$ 回，そして定数倍が $2\ell - 1$ 回となる．乗算は並列化できないため，通信回数は $O(\ell)$ となる．なお上記手続き (2) の全加算回路演算を文献[11] の Protocol $[d]_B$ (Section 6.2) に置き換えれば，通信回数を $O(1)$ とできるが，乗算の回数が $55\ell \log_2 \ell$ となる．

[整数変換処理]

X, Y, Z は $a[i] \in \mathbb{Z}/2\mathbb{Z} \subseteq \mathbb{Z}/m\mathbb{Z}$ $(i = 1, \ldots, \ell)$ の分散データ $a[i]_x$, $\vec{a}[i]_y$, $\vec{a}[i]_z$ から $a \in \mathbb{Z}/m\mathbb{Z}$ の分散データ a_x, $\vec{a}_y$, $\vec{a}_z$ を求める．$a = \sum_{i=1}^{\ell} 2^{i-1}a[i] \bmod m$ が成り立つことから，加算および定数倍演算を実行すれば良い．

3.2 不正検出方式

3.1 節で述べた基本方式に以下の手続きを追加する．

A. X, Y, Z は対象の計算について各主体の処理を入れ替えて 3 回実行する．対象の計算は 3.1 節で述べた加減算，定数倍演算，乗算，論理回路演算，2 進変換処理，整数変換処理の組み合わせとなる．入力値 a の分散データ a_x, $\vec{a}_y$, $\vec{a}_z$ も実行ごとに異なるものを用いる．

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

B. 対象の計算を行う前に，各主体の入力となる $a^{(j)}$ の分散データ $a_x^{(j)}$, $\vec{a}_y^{(j)}$, $\vec{a}_z^{(j)}$ $(j = 1,2,3)$ が正しく生成されていることを，入力値 $a(= a^{(j)})$ を復元すること無く確認する．不正入力を検出するために必要となる．具体的には以下を行い，正当性を確認できない場合は不正検出として処理を終了する．

- 【$\hat{a}^{(j)}$ の正当性確認】Y, Z は $\hat{a}^{(j)}$ を互いに送信し，等しいことを確認する．
- 【$a_x^{(j)}$ の正当性確認】Y, Z は $r_{yz} \in_R \mathbb{Z}/m\mathbb{Z}$ を共有し，X にそれぞれ $u_y = a_y^{(j)} + r_{yz}$, $u_z = a_z^{(j)} - r_{yz}$ を送信する．X は $a_x^{(j)} = u_y + u_z$ が成り立つことを確認する．
- 【$a_y^{(j)}$ の正当性確認】X, Z は $r_{xz} \in_R \mathbb{Z}/m\mathbb{Z}$ を共有し，Y にそれぞれ $u_x = a_x^{(j)} + r_{xz}$, $u_z = a_z^{(j)} + r_{xz}$ を送信する．Y は $a_y^{(j)} = u_x - u_z$ が成り立つことを確認する．
- 【$a_z^{(j)}$ の正当性確認】X, Y は $r_{xy} \in_R \mathbb{Z}/m\mathbb{Z}$ を共有し，Z にそれぞれ $u_x = a_x^{(j)} + r_{xy}$, $u_y = a_y^{(j)} + r_{xy}$ を送信する．Z は $a_z^{(j)} = u_x - u_y$ が成り立つことを確認する．
- 【$a_x^{(1)} + \hat{a}^{(1)} = a_x^{(2)} + \hat{a}^{(2)} = a_x^{(3)} + \hat{a}^{(3)}$ の確認】X, Y, Z はそれぞれ $r_x, r_y, r_z \in_R \mathbb{Z}/m\mathbb{Z}$ を生成して秘匿処理を行い，$a_x^{(1)} + \hat{a}^{(1)} + r$, $a_y^{(2)} + \hat{a}^{(2)} + r$, $a_z^{(3)} + \hat{a}^{(3)} + r$ を計算して復元し，復元結果が等しいことを確認する．ここで $r = r_x + r_y + r_z$ とする．

C. 復元処理の前に，対象の計算の全ての計算結果が等しいことを，計算結果を復元すること無く確認する．なお X, Y による復元，Y, Z による復元，そして X, Z による復元全てについて確認する．したがって 9 つの計算結果について等号判定を行う．なお確認方法は手続き B の最終段の処理を行えば良い．

次に，何れかの主体によって計算結果が改竄された場合，上記の手続きがその不正を検出できることを示す．先ず証明の方針を以下に示す．

I. 計算結果を改竄するための不正は，

a. 入力の分散データを不正値とする，

b. 乗算または 2 進変換処理の秘匿処理の手続きで他の主体に不正値を送信する，または

c. 復元処理で不正値を開示する

の何れかに帰着されることに着目する．なお手続き B において何れかの主体が手続きを偽った場合は，不正 a の検出ができなくなるが，その場合は不正 b または c の問題に帰着されることに注意する．

II. 不正 b は $1/2^{\ell}$ 以下の確率を除き手続き C で不正検出されることを示す．

不正 a は各主体が手続き B を正しく行えば防ぐことができる．不正 c は手続き C によって防ぐことができる．したがって以下では不正 b について考える．

2 進変換処理の秘匿処理の手続きで他の主体に不正値を送信する場合は，各主体が手続き B を正しく行えば手続き B で検出され，そうでない場合は，次段以降の乗算または 2 進変換処理の秘匿処理の手続きで他の主体に不正値を送信することに帰着される．次段以降に乗算および 2 進変換処理の秘匿処理が無い場合は不正 c の問題に帰着される．

次に，乗算の手続きで他の主体に不正値を送信する場合を考える．各主体は乗算 $c = ab$ において主体 X の手続きを行う際，主体 Y, Z にそれぞれ (r_1, r_2, r_3, c_y), $(a_x - r_1, b_x - r_2, r_4, c_z)$ を送信する．r_1, r_2, r_3, r_4, c_y は乱数であるから，$a_x - r_1, b_x - r_2, c_z$ をそれぞれ $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ に置き換えたとき，Z が得る乗算結果の分散データは $\vec{\tilde{c}}_z = (\tilde{\tilde{c}}, \tilde{c}_z) = (y + \hat{a}\tilde{b} + \tilde{a}\hat{b} + r_4, \tilde{c})$ となる．すなわち，$(\hat{c} - \tilde{\tilde{c}}, c_z - \tilde{c}) = (\hat{a}(b_x - r_2 - \tilde{b}) + (a_x - r_1 - \tilde{a})\hat{b}, c_z - \tilde{c})$ の差分が生じる．しかし X は $\hat{a}$, $\hat{b}$, c_z を知らないため，この差分をもう 2 回の乗算 $c = ab$ の実行時の Z の分散データに付加することができず，手続き C で不正検出されない確率は高々 $1/2^{\ell}$ となる．

最後に，本提案の基本方式と不正検出方式について，情報理論的安全性に基づく既存方式との比較を行う．既存方式[8),9)] では算術演算および論理回路演算を独立に実行することができ，2 進変換処理[11),16)] によってそれらを組み合わせた計算が可能となる．算術演算および論理回路演算は既存方式，提案方式ともに，$O(\ell)$ の計算量となり乗算のみ $O(1)$ の通信回数および $O(\ell)$ の通信量を必要とするが，提案方式は乗算の計算がより単純となる．また既存方式[8)] では，不正検出のために Generalized Reed-Miller Code に基づくエラー訂正処理が行われているが，4 主体以上が計算に関与する必要があり，提案方式とは主体構成が異なる．

4. 考　　察

本節では提案方式をプライバシ保護データマイニング技術として利用する場合の運用モデルの考察を行う．

提案方式は，3 主体の協調計算により入力値を秘匿しつつ各種計算を効率良く実行し，2 主体の取得情報を得ない限り入力値を秘匿できる特徴を持つ．そこで 2 主体が保持する情

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

報を用いてプライバシ保護データマイニングを行う場合は，図 1 のように第三者機関を介入させた構成（第三者機関介入型）が挙げられる．すなわち，2 つの情報保持主体と第三者機関による 3 主体によって提案方式を実行する．最終的に計算結果を復元するのは各情報保持主体とすれば良い．各情報保持主体は保持情報を秘匿処理し，自身を含めた 3 主体に秘密分散する．なお各情報保持主体からみれば，第三者機関と他の情報保持主体の結託により保持情報が知られ得るため，第三者機関はプライバシ保護データマイニングをサポートするサービス主体として社会的信頼を得ていることが望ましい．

第三者機関介入型と同様の主体構成として，情報保持主体とは異なる分析主体を第三者機関と置き換えても良い（図 2）．この場合，どちらかの情報保持主体が計算結果の復元に必要な分散データを分析主体に送信することで，分析主体のみ計算結果を復元することができる．

3 主体が保持する情報を用いてプライバシ保護データマイニングを行う場合は，図 3 のように提案方式の直接的な利用（全参加型）が挙げられる．ただし他の情報保持主体の結託により保持情報が知られ得るため，図 4 のように 3 つの異なる第三者機関に保持情報を秘密分散して提供し，計算を委託する主体構成（計算委託型）も考えられる．計算委託型の場合は，秘匿関数計算を実行する 2 主体が結託しなければ良いため，秘匿関数計算を実行する主体の 1 つは情報保持主体や分析主体としても良い．

4 以上の主体が保持する情報を用いてプライバシ保護データマイニングを行う場合は，3 主体の場合と同様に図 4 の主体構成を適用できる．

5. おわりに

個人に関する情報の保護と利活用を両立させる汎用的なプライバシ保護データマイニング技術として，3 主体の協調計算により，2 主体の取得情報を得ない限り入力値を秘匿できる秘匿関数計算方式を提案した．提案方式は，単純な計算により秘匿関数計算を実現しており，入力や各主体の処理の不正を効率良く検出できる特徴を持つ．また算術演算と論理回路演算の組み合わせにも適用できるため，算術演算とともに絶対値や大小比較といった論理回路演算が必要となるマイニングや統計処理に適していると考えられる．

参考文献

1) 健康情報活用基盤構築のための標準化及び実証事業，
https://microsite.accenture.com/meti/Pages/default.aspx.
2) 情報大航海プロジェクト，http://www.igvpj.jp/index/.
3) 「地理空間情報サービス産業の将来ビジョン」及び「G空間プロジェクト」の公表について（METI/経済産業省），
http://www.meti.go.jp/press/20080703007/20080703007.html.
4) Geographic Privacy-aware Knowledge Discovery and Delivery,
http://www.geopkdd.eu/.
5) Electronic Health Information Laboratory – KnowledgeBase,
http://www.ehealthinformation.ca/knowledgebase/.
6) C.C. Aggarwal and P.S. Yu, Privacy-Preserving Data Mining: Models and Algorithms, ISBN 978-0-387-70991-8, Springer-Verlag, Jul. 2009.
7) J. Algesheimer, J. Camenisch, and V. Shoup, Efficient computation modulo a shared secret with application to the generation of shared safe-prime products, CRYPTO 2002, LNCS 2442, pp. 417–432, Springer-Verlag, 2002.
8) M. Ben-Or, S. Goldwasser, and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, STOC ’88, pp. 1–10, ACM Press, 1988.
9) D. Chaum, C. Crepeau, and I. Damgård, Multiparty unconditionally secure protocols, STOC ’88, pp. 11–19, ACM Press, 1988.
10) R. Cramer, I. Damgård, and J. B. Nielsen, Multiparty computation from threshold homomorphic encryption, EUROCRYPT 2001, LNCS 2045, pp. 280–300, Springer-Verlag, 2001.
11) I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, Unconditionally secure constant-rounds multi-party computation for equality, comparison bits and exponentiation, TCC 2006, LNCS 3876, pp. 285–304, Springer-Verlag, 2006.
12) C. Gentry, Fully homomorphic encryption using ideal lattices, STOC ’09, pp. 169–178, ACM Press, 2009.
13) O. Goldreich, S. Micali, and A. Wigderson, How to play any mental game, or a completeness theorem for protocols with honest majority, STOC ’87, pp. 218–229, ACM Press, 1987.
14) M. Jakobsson and A. Juels, Mix and match: secure function evaluation via ciphertexts, ASIACRYPT 2000, LNCS 1976, pp. 162–177, Springer-Verlag, 2000.
15) Y. Lindell and B. Pinkas, Privacy preserving data mining, CRYPTO 2000, LNCS 1880, pp. 36–54, Springer-Verlag, 2000.
16) T. Nishide and K. Ohta, Multiparty computation for interval, equality, and comparison without bit-decomposition protocol, PKC 2007, LNCS 4450, pp. 343–360, Springer-Verlag, 2007.
17) B. Pinkas, T. Schneider, N.P. Smart, and S.C. Williams, Secure two-party computation is practical, ASIACRYPT 2009, LNCS 5912, pp. 250–267, Springer-Verlag,

Vol.2010-DPS-142 No.1
Vol.2010-CSEC-48 No.1
2010/3/4

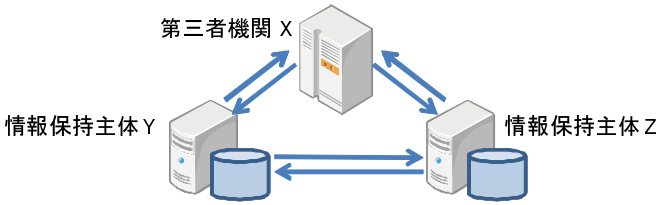

図 1 第三者機関介入型

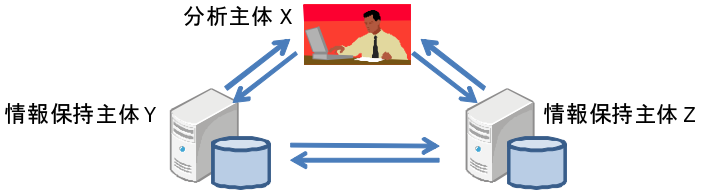

図 2 分析主体介入型

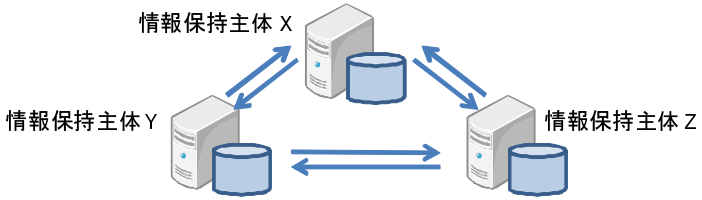

図 3 全参加型

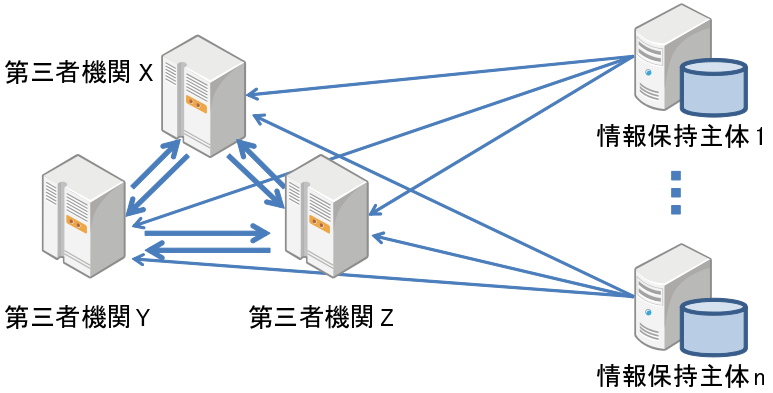

図 4 計算委託型

2009.
18) B. Schoenmakers and P. Tuyls, Practical two-party computation based on the conditional gate, ASIACRYPT 2004, LNCS 3329, pp. 119–136, Springer-Verlag, 2004.
19) B. Schoenmakers and P. Tuyls, Efficient binary conversion for Paillier encrypted values, EUROCRYPT 2006, LNCS 4004, pp. 522–537, Springer-Verlag, 2006.
20) J. Vaidya, C.W. Clifton, and Y.M. Zhu, Privacy Preserving Data Mining, ISBN 0-387-25886-8, Springer-Verlag, Nov. 2005.
21) A. C. Yao, Protocols for secure computations, FOCS '82, pp. 160–164, IEEE Press, 1982.
22) A. C. Yao, How to generate and exchange secrets, FOCS '86, pp. 162–167, IEEE Press, 1986.
23) 柴田 賢介，千田 浩司，五十嵐 大，山本 太郎，高橋 克巳，表計算ソフトをフロントエンドとした委託型 2 パーティ秘匿回路計算システム，CSS2009, 2009.