

プライバシーを保護したデータ突合プロトコル

長尾 佳高^{1,a)} 宮地 充子^{1,2,b)}

概要: 近年、様々な機関が大量のデータを保有しており、一般的にそれらのデータは複数の属性、例えば名前、住所、病名等により構成される。そのため同じ人物に関するデータをいくつかの異なる機関が保有しており、これらのデータをプライバシーを侵害することなく統合することができれば原因解析や各属性間の関係などの分析が実現できる。また、統合処理では、データの種類によらず実行できることや、必ず正しい統合データが得られることが必要となる。既存研究である PSI を用いたアプリケーションプロトコルでは付随する解析データに制限があり、また多機関の場合に対応していないものが多い。本論文では、データを突合属性、解析属性、その他の三つの属性に分類し、各属性データを適切に秘匿したデータ突合方式を提案する。この方式では計算量が機関の数に依存せず、目的の統合データを必ず入手することができる。また、統合データが得られることの実験データも提示する。

キーワード: Privacy, data integration, PSI

Privacy Preserving Data Integration Protocol

YOSHITAKA NAGAO^{1,a)} ATSUKO MIYAJI^{1,2,b)}

Abstract: Recently, large amount of data is collected by various organizations. Generally, data consists of various attributes such as name, address, medical term, etc. Related to the same person, different organizations often possess data with different attributes. If we can integrate data kept in different organization related to the same person without violating privacy, detailed analyzes such as cause investigation or relations among attributes could be realized. In such a scenario, we do not need personal information while it should be protected securely. Importantly, the data exactly integrates data associated with the same person. In this paper, we classify attributes in data into three of matching attributes, analyzing attributes, and others. Then, we propose a privacy preserving data integration protocol while handling data privacy appropriately according to classification of matching, analyzing attributes, and others.

Keywords: Privacy, data integration, PSI

1. 序章

近年、様々な機関が大量のデータを保有している。一般に、それらのデータは名前、住所や医療用語など様々な属性から構成される。異なる機関はそれぞれ別の属性を持っており、同じ人物の情報として付随している場合がある。

もしプライバシーを侵害することなく異なる機関の同じ人物に対応する情報を統合できたなら、原因解析や属性間の関係分析など、詳細な解析に役立てることができる。本シナリオでは、保護されるべき個人情報が必要とせず、個人の情報に基づいて統合されたデータを必ず入手することができる。例えば、表 1 と表 2 にあるように、2 つの病院が同じ人物に対する医療情報を持っているという状況を考える。各病院はそれぞれ $Data_1 = \{(\text{名前}, \text{血液型}, \text{病名}, \text{日付})\}$ と $Data_2 = \{(\text{名前}, \text{病名}, \text{日付})\}$ を所有している。これらの情報を突合する場合、突合データとしてある患者が 2016 年 4 月に胃がんになり、2018 年 5 月に脳梗塞になっ

¹ 大阪大学
Osaka University

² JAIST
北陸先端科学技術大学院大学

a) nagao@cy2sec.comm.eng.osaka-u.ac.jp

b) miyaji@comm.eng.osaka-u.ac.jp

ているという情報が得られる．もし同じ人物に対する情報を統合できた場合，異なる病気の詳細に解析することが可能になる本シナリオでは，名前などの秘匿されるべき個人情報を必要としない．同じ人物の情報が統合されることが重要であり，これは第三者機関，例えば解析機関は個人情報を除いた統合データを必要とする．

表 1 病院 1(Data₁)

名前	血液型	病名	日付
田中	A	胃がん	2016/2
林	B	肝臓がん	2016/3
佐藤	AB	胃がん	2016/4

表 2 病院 2(Data₂)

名前	病名	日付
田中	脳梗塞	2018/5
鈴木	脳出血	2018/6
高橋	くも膜下出血	2018/7

PDDI プロトコル [1] は本方式において重要な役割を担っている．PDDI [1] などの PSI プロトコルは複数の機関が保有するデータセットの共通集合を安全に計算するプロトコルである．最終的に，各機関は共通集合のみを取得し，それ以外のデータセットに関する一切の情報を得ることはできない．しかし，PSI はデータの共通集合を取得することを目的としており，共通集合に付随するデータを安全に統合する手法については提供していない．また，以前我々が提案したデータ突合プロトコル [2,3] では PDDI [1] の分散復号処理の通信制約による問題を解決し，安全なデータ突合プロトコルを提案している．しかし，偽陽性による統合データへの影響について詳細に触れておらず，突合データが偽陽性によらず正しく取得できることが実験データから示されていない．また暗号化方式として拡張 ElGamal 暗号を使用しているが，データサイズの削減と高速化の観点から，本論文では楕円曲線上の ElGamal 暗号が利用できることを説明し，新しく突合プロトコルを提案する．

本論文は次のように構成される．まず，2 章ではプロトコルを提案する上で必要となる諸定義や基礎的な知識について説明し，3 章でいくつかの既存研究を紹介する．また，4 章で本提案方式であるデータ突合プロトコルについて説明を行う．

2. 準備

ここでは，本研究に使用する諸定義やデータ構造，暗号方式について説明する．

定義 1 (ECDLP) 有限体 $\mathbb{F}_p$ 上の楕円曲線 $E(\mathbb{F}_p)$ について，2つの元 Y, G とする．このとき， $Y = xG = G + \dots + G$ となる x が存在するならばその値を求める問題を楕円曲線離

散対数問題 (Elliptic Curve Discrete Logarithm Problem, ECDLP) という．

2.1 Bloom filter

Bloom filter (以後 BF と表記) は確率的データ構造の一つである．BF は長さ m の配列と k 個の独立なハッシュ関数 $H_1, \dots, H_k$ により構成され，Const と ElementCheck の二つの関数でデータの格納・チェックを行う．BF には偽陽性があり，その確率 FPR は式 $\text{FPR} = \{1 - (1 - \frac{1}{m})^{kw}\}^k \approx \{1 - e^{-kw/m}\}^k$ で与えられる．

Algorithm 1 Const(S)

Input: A set S

Output: A Bloom filter $\text{BF}(S)$

```

for  $i = 0$  to  $m - 1$  do
   $\text{BF}(S)[i] \leftarrow 1$ 
end for
for all  $x \in S$  do
  for  $i = 0$  to  $k - 1$  do
     $j = H_i(x)$ 
    if  $\text{BF}(S)[j] = 1$  then
       $\text{BF}(S)[j] \leftarrow 0$ 
    end if
  end for
end for
output  $\text{BF}(S)$ . stop.

```

Algorithm 2 ElementCheck(BF, x)

Input: A Bloom filter $\text{BF}(S)$, an element x

Output: 1 if $x \in S$ and 0 if $x \notin S$

```

for  $i = 0$  to  $k - 1$  do
   $j = H_i(x)$ 
  if  $\text{BF}(S)[j] = 1$  then
    output 0. stop.
  end if
end for
output 1. stop.

```

次に，BF のアルゴリズムについて説明する．一般に用いられている BF の構成とは異なり，本方式では BF の全ての配列要素を 1 で初期化し， $\text{BF}[H_i(x)]$ に対応する要素を 0 にセットする．本方式において BF は $\cap_{i=1}^n S_i$ の計算に用いられるが，これは各 $\text{BF}(S_i)$ を足し合わせた IBF により実現する．従来の BF を用いた場合， $x \in S_i$ において $\text{IBF}(\cap S_i)[H_i(x)] = n$ となる．一方，Algorithm 1 を用いることにより $x \in S_i$ において $\text{IBF}(\cap S_i)[H_i(x)] = 0$ となる．これにより，PDDI [1] での逆元計算を削減することができる．また，本提案では楕円曲線上の ElGamal 暗号の準同型性を利用するために用いられる．

2.2 楕円曲線

暗号で用いられる楕円曲線は次の式で与えられる 3 次曲線と無限遠点 $\mathcal{O}(\infty, \infty)$ のことである。

$$E: y^2 = x^3 + ax + b \pmod{p}$$

ここで, $a, b \in \mathbb{Z}_p$ であり, $x, y \in \mathbb{F}_p$ である。ただし, $p \leq 5$, $4a^3 + 27b^2 \neq 0$ とする。有限体 $\mathbb{F}_p$ 上の有理点の集合は楕円曲線上で可換群となり, $E(\mathbb{F}_p)$ と表す。また, このときの単位元は $\mathcal{O}$ となる。楕円曲線上の 2 点 $A = (x_1, y_1)$, $B = (x_2, y_2)$ に対し, $A + B$ は A と B を結ぶ直線と楕円曲線とのもう一つの交点を R としたとき, x 軸に対称な点 $R' = (x_3, y_3)$ で定義される。 $R' = (x_3, y_3)$ は次式で与えられる。

- $A \neq B$ のとき

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2$$

$$y_3 = \frac{y_2 - y_1}{x_2 - x_1} (x_1 - x_3) - y_1$$

- $A = B$ のとき

$$x_3 = \frac{3x_1^2 + a}{2y_1} - 2x_1 = \frac{x_1^4 - 2ax_1^2 - 8bx_1 + a^2}{4(x_1^3 + ax_1 + b)}$$

$$y_3 = \frac{3x_1^2 + a}{2y_1} (x_1 - x_3) - y_1$$

- $A = -B$ のとき

$A+B$ は無限遠点 $\mathcal{O}(\infty, \infty)$ となる。

$P(x_1, y_1)$ の逆元は $-P = (x_1, -y_1)$ となる。また, スカラー $n \in \mathbb{Z}_p$ に関する倍算については次のように定義する。

$$nP = \begin{cases} P + \dots + P (n \text{ 回の加算}) & n > 0 \\ -n(-P) & n < 0 \\ \mathcal{O} & n = 0 \end{cases}$$

2.2.1 楕円曲線上の ElGamal 暗号

ElGamal 暗号は公開鍵暗号の 1 つであり, 加法と乗法に関して準同型性がある。楕円曲線上の ElGamal 暗号は次のようにして実現される。

- 鍵生成・交換

1. 位数の大きな有限体 $\mathbb{F}_p$ 上の楕円曲線 $E(\mathbb{F}_p)$ を選択する。
2. 楕円曲線上の点 $G \in E(\mathbb{F}_p)$ を選択する。
3. 秘密鍵として $x \in \mathbb{Z}_p$ をとり, $Y = xG$ を計算する。
4. Y を公開鍵として共有する。また, 楕円曲線 $E(\mathbb{F}_p)$, 有限体 $\mathbb{F}_p$, 点 G は公開情報とする。

- 暗号化

1. 平文 m を楕円曲線上の点 $M \in E(\mathbb{F}_p)$ に対応付ける。
2. 乱数 $r \in \mathbb{Z}_p$ をとり, $U = rG$, $V = M + rY$ を計算する。
3. (U, V) を暗号文として送信する。

- 復号

1. 秘密鍵 s を用いて $Z = V - sU$ を計算する。復号は以下のように行われる。

$$\begin{aligned} V - xU &= (M + rY) - x(rG) \\ &= (M + xrG) - xrG \\ &= M \end{aligned}$$

2. 点 M から平文 m を取得する。

2.2.2 楕円曲線上の ElGamal 暗号による秘密分散

楕円曲線上の ElGamal 暗号は機関 $P_1, \dots, P_n$ 間の閾値 (n, n) の秘密分散に利用できる。この秘密分散は以下のように行われる。

1. 各公開鍵 Y_i からグループ公開鍵 $Y = \sum Y_i$ を計算し, 各 M_i の暗号化にはグループ公開鍵 Y を用いる。
2. 各暗号文 $(U_i, V_i), \dots, (U_n, V_n)$ を統合し, 暗号文 $(U, V) = (\sum U_i, \sum V_i)$ を得る。
3. 各機関は自身の秘密鍵 s_i を用いて部分シェア $Z_i = x_i U$ を計算する。
4. 部分シェア Z_i から完全シェア $Z = \sum Z_i$ を計算し, 復号データ $\sum M_i = V - Z$ を得る。

3. 既存研究

PSI (Private set intersection) は 2 機関の所有するデータセットの共通集合を安全に計算・取得するプロトコルである。本章では PSI に関する既存研究として, Circuit ベースの PSI プロトコル [4], OT ベースの PSI プロトコル [5], PSI を用いたアプリケーションである Private Intersection-Sum [6], 多機関データの PSI である PDDI [1] を紹介する。

Circuit ベースの PSI プロトコル [4] は疑似乱数関数として PRF の拡張である OPPRF を採用しており, 同じ入力に対して互いの入力を秘匿したままある特定の値を入手することができる。入力として Cuckoo Hashing をベースとするデータ構造を, 出力の比較として Circuit を用いることにより, 安全に $O(n)$ の計算量で共通集合を求めることができる。

OT ベースの PSI プロトコル [5] は新しいデータ構造である PaXoS (probe-and-XOR of strings) を用いた高速な PSI であり, malicious なモデルにおいて安全である。PaXoS は n 個のバイナリ文字列を m 個のバイナリ文字列にマッピングするランダム化関数であり, それぞれのオリジナル文字列は m 個の文字列をある特定の法則で排他的論理和をとることで取得することができる。PaXoS は $m \times n$ の行列であり, これを用いた暗号化・復号処理は一般に n が大きくなるにつれ計算量が膨大になる。そのため Cuckoo Graph を導入し, 1 つのオリジナル文字列に対しその情報を Cuckoo hashing を用いて PaXoS の要素に埋め込む。復号の際にはこの要素の位置を示すインデックスを取得することができ, 排他的論理和をとればオリジナル文字列となるよう PaXoS を構築することで効率的に暗号文を計算し, 暗号化・復号処理を高速に行うことを可能としている。

Private Intersection-Sum [6] は PSI を用いたアプリケーションプロトコルであり、機関の所有するデータセットの共通集合ではなく、共通集合に付随するデータの総和を安全に計算・取得する。P₁, P₂ はそれぞれ m₁, m₂ 個のデータセット $V = \{v_i\}_{i=1}^{m_1}$ と $W = \{(w_i, t_i)\}_{i=1}^{m_2}$ を所有する。このとき、 $S = V \cap W$ とするとプロトコルを実行することで一方は共通集合の数 $|S|$ を取得し、もう一方は共通集合 S に付随するデータの総和 $\sum t_i$ を取得する。また、各機関の役割を入れ替えてプロトコルを実行することで両機関とも共通集合の個数とそれに付随するデータの総和が取得できる。

PDDI [1] は Bloom filter を用いた PSI プロトコルであり、2 機関ではなく多機関の所有するデータセット S_i の共通集合を計算・取得することができる。このプロトコルでは各機関の所有する Bloom filter を拡張 ElGamal 暗号で暗号化し、分散復号することで各機関の情報を秘匿しながら $\cap S_i$ を取得できる。このプロトコルは semi-honest なモデルにおいて安全であり、分散復号の際に必要なシェアの共有は P2P 通信を前提としている。

4. 提案方式

本章では提案プロトコルの詳細について説明する。

4.1 コンセプト

使用する記号は以下の通りである。

- P_i: データセット Data_i を保有する機関。
- C: S_i の共通集合 $\cap S_i$ に付随する属性を取得するクライアント。
- O: 計算の負担と各機関の通信の中継を行うが、データに関する情報を得ることのないサーバ。
- |S|: 集合 S に含まれる要素数。
- Data_i = {(s_{ij}, α_{ij})}: P_i の保有するデータセット。s_{ij} と α_{ij} はそれぞれ突合属性、解析属性。
- S_i = {s_{i,1}, ..., s_{i,ω_i}}: P_i のデータセット Data_i (|S_i| = ω_i) に含まれる解析属性の集合。
- $\cap S_i = \{s_1, \dots, s_h\}$: 全 S_i の共通集合。(| $\cap S_i$ | = h)
- PEnc(Y, m): 楕円曲線上の ElGamal 暗号による暗号化。(公開鍵 Y, メッセージ m)
- PDec(x, C): 楕円曲線上の ElGamal 暗号による復号。(秘密鍵 x, 暗号文 C)
- SEnc(K, m): 共通鍵暗号による暗号化。(共通鍵 K, メッセージ m)
- SDec(K, c): 共通鍵暗号による復号。(共通鍵 K, 暗号文 c)
- Hybrid(Y, m): PEnc(Y, K), SEnc(K, m) を用いたハイブリッド暗号方式による暗号化。(公開鍵 Y, メッセージ m)
- m, k: BF 長とハッシュ関数の個数。

- BF(S_i) = [BF_i[0], ..., BF_i[m - 1]]: S_i を埋め込んだ Blom filter。
- IBF($\cap S_i$) = [$\sum_{i=1}^n$ BF_i[0], ..., $\sum_{i=1}^n$ BF_i[m - 1]]: S₁, ..., S_n を統合した Bloom filter。
- r = [r₁, ..., r_m]: BF のランダムサイズに用いる m 次元配列。

PSI に関する既存研究 [1, 4, 5, 7–11] では共通集合の取得を目的としている。一方、多くの場合、2 のようにデータセットにデータが付随している。Private Intersection-Sum [6] は 2 機関の所有するデータセット S₁ と S₂ から共通集合 |S₁ ∩ S₂| とこれに付随する値の総和を計算・取得するプロトコルである。しかし、2 機関でのプロトコルであることと付随するデータについて数値計算のみしか行うことができない。一方、本研究の目標であるデータの突合処理では数値以外にも様々なデータを扱うことが想定される。

例えば、1 章で紹介したシナリオにおいて、データの属性を突合属性、解析属性、その他の 3 つのカテゴリに分類する。表 1 において、名前を突合属性、病名と日付を解析属性、血液型をその他に対応付ける。これらの属性の種類によって、保護されるべきプライバシー要件は異なっている。本論文では、プライバシーを侵害することなく異なる機関が保有する同じ突合属性に付随する解析属性を統合するプロトコルを提案する。これは多機関が所有するデータを突合属性に従って安全に統合する新しいプロトコルであり、本方式で扱うデータは突合属性、解析属性、その他に分類され、それぞれ適切に秘匿される。解析属性は原因解析や市場動向の調査などのデータ解析に利用できる属性であり、突合属性は例えば氏名など、データを統合する指標となる属性である。もし異なる機関に所有されている、同じ突合属性に付随する解析属性を統合したデータが得られれば、より有用なデータ解析への利用が期待できる。本研究の目標は、この統合処理を実現することである。また、統合されたデータを取得する機関をクライアント C とする。本プロトコルでは突合属性に対し PSI を利用し、その後突合属性に従って解析属性を統合することで、プライバシーを侵害することなく統合処理を実現する。

次に、機関 P_i のプライバシーについて説明する。突合属性について PSI を用いる際、各機関は共通集合以外の他機関に関する情報を得られないことが必要となる。また、解析属性を統合する際には、クライアントのみが統合された解析属性を入手し、各機関は他機関のデータを得られないことが必要となる。また、データのプライバシーに加え、機関のプライバシーも考える必要がある。特に、どの機関がどの解析属性を保有しているかという情報をクライアントに秘匿しなければならない。本プロトコルでは PDDI [1] や O のランダムサイズ・シャッフル処理によってデータのプライバシーと機関のプライバシーを保証しながら、分散管理されたデータを突合し、クライアントが入手することを可能に

する。

4.2 提案プロトコル

本プロトコルは3機関 P , O , C により実行され、大まかに次の2ステップに分けられる。

1. MPSI [1] を利用し、各機関の所有するデータセットの共通集合を計算する。
2. 1 で得た共通集合とそれに付随するデータをそれぞれプライバシー要件に従って適切に暗号化し、データの突合を行い C が入手する。

ステップ1 はさらに Bloom filter の暗号化フェーズと共通集合の計算フェーズに分けられる。本プロトコルに必要な要件は以下の通り。

- 拡張性：突合属性・解析属性の種類に関わらずプロトコルを実行できる。
- 最小限の通信制約：各機関 P_i 間の P2P 通信を必要としない。これは実環境において各機関の直接通信を確立するのは困難であるためである。P2P 通信の代替として、本プロトコルでは P , C 間の通信に簡単なクライアントサーバモデルを導入する。つまり、 $P_i \leftrightarrow O$, $O \leftrightarrow C$ の通信のみを前提とする。
- 耐障害性：プロトコルが実行されれば、 C は確実に目的となる解析属性データを得られる。

また、プライバシー要件は以下の通り。

定義 2 以下のプライバシー要件を S が満たしている時、 S は party-private であるとする。

- P_i は $P_j (i \neq j)$ のデータセット Data_j に関して、共通集合 $\cap S_i$ 以外の情報を入手することができない。また、 P_j の所有する解析属性に関する情報も入手することができない。
- C は共通集合 $\cap S_i$ に付随する解析属性のみを取得でき、それ以外の突合属性・解析属性の情報を入手することができない。
- O は共通集合のサイズ $|\cap S_i|$ を除く突合属性・解析属性に関する一切の情報を入手することができない。

次に、共通集合の計算と突合属性・解析属性の暗号化の詳細について記述する。PDDI [1] では共通集合を復号する際、各機関の所有するシェア z_i から完全シェア $z = \Pi z_i$ を計算し、 z を用いて暗号文を復号する。一方本方式では各機関 P_i 間の P2P 通信は前提としていない。図1 はそれぞれのプロトコルの分散復号に用いる通信を示している。

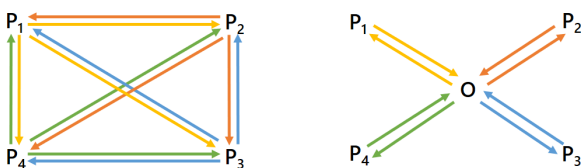


図1 分散復号 PDec に必要な通信

左図は P2P 通信を用いた分散復号処理を表しており、PDDI [1] ではこのように分散復号を行う。一方クライアントサーバ方式を採用している本方式では右図のようになり、各機関 P_i 間の通信を O が中継するため、シェア z_i から安全に z を計算する処理を考える必要がある。ここで、各 P_i の中で最も計算能力の高い機関を P_1 とし、他の機関 P_i は同程度の計算能力を持つものとする。本方式では暗号の準同型性と P_1 の計算能力を利用することで、シェア z_i から z を安全に計算する方法を提案する。

次に、統合した解析属性を取得する方法を説明する。機関 P_i の所有するデータセットを秘匿しながら C が突合データを取得するために必要な条件は次の4つに分類される。

1. 突合属性の共通集合 $\cap S_i$ は P_i のみ復号可能。
2. 解析属性はその属性を所有する P と C のみ復号可能。
3. O は暗号化された突合属性を識別し、解析属性を統合することが可能。
4. C はある特定の解析属性を所有する機関を判別することが不可能。

この4つの条件を満たすため、以下の方法で各属性データを秘匿する。

1. P_i により突合属性の共通集合 $\cap S_i = \{s_j\}$ を計算したのち、 $\cap S_i = \{s_j\}$ の各要素 s_j は全機関 P_i で共有する共通鍵 K_P により $\text{SEnc}(K_P, s_j)$ に暗号化する。 O は暗号文から s_i を復号できないが、同一要素に対する暗号文が等しいことから突合処理を行うことができる。
2. P_i の解析属性 $\{\alpha_{ij}\}_j$ はハイブリッド鍵暗号方式により $\text{Hybrid}(y_c, \alpha_{i,j})$ に暗号化し、 C に送信する。解析属性は固定長でないため、公開鍵暗号ではなく共通鍵暗号による暗号化が望ましい。
3. O は同じ突合属性に対応する解析属性を統合し、シャッフルした後 C に送信する。
4. C は統合された解析属性のみを得る。

また、本方式では PDDI [1] とは異なり、閾値暗号として楕円曲線上の ElGamal 暗号を用いる。PDDI [1] では拡張 ElGamal 暗号を用いているが、これは閾値による分散復号が可能であることと加法に対して準同型であることを利用している。楕円曲線上の ElGamal 暗号はこの2つの性質を満たしており、また、拡張 ElGamal 暗号と比較してより高速な暗号化・復号処理が期待できる。楕円曲線上の ElGamal 暗号はメッセージ m を楕円曲線上の点 M に対応付ける必要があり、 M に対しては準同型性が存在するが一般にメッセージ m に対して準同型性があるとは限らない。本方式では $x \in \text{BF}$ に対し $\text{BF}[H_i(x)] = 0$ としており、 $m = 0$ に対して準同型性があればよい。そのためメッセージ $m = 0$ を無限遠点 O に対応付けることにより、 $\sum r_i O = O$ であることから加法・スカラー倍の演算について準同型性が利用できる。また、 $m = 1$ については簡単に点 G を対応付けるが、準同型性がないため $\sum m$ の復号

は不可能である。

以下にプロトコルの詳細を記載する。また、2-4 は各フェーズを表している。

Initialization:

システムパラメータとして有限体 $\mathbb{F}_p$ と $\mathbb{F}_p$ 上の楕円曲線 $E(\mathbb{F}_p)$, $E(\mathbb{F}_p)$ 上の点 P を用いる。 P_i , C は楕円曲線上の ElGamal 暗号 PEnc/PDec と共通鍵暗号 SEnc/SDec を利用でき、ハイブリッド暗号方式 Hybrid も同様に利用できる。また、Bloom filter を構成する関数である Const と SetCheck は P_i のみ利用できる。

1. P_i は秘密鍵として $x_i \in \mathbb{Z}_p$ をとり、 $Y_i = x_i G$ を計算し、 Y_i を公開鍵として共有する。
2. P_i は n 機関の公開鍵 $Y = \sum Y_i \pmod{p}$ を計算する。
3. 各機関 P_i は共通鍵 K_P を互いに共有する。
4. C は秘密鍵として $x_c \in \mathbb{Z}_p$ をとり、 $Y_c = x_c G$ を計算し、 Y_c を公開鍵として共有する。

Phase I: IBF の暗号化

1. P_i はデータセット $S_i = \{s_{ij}\}$ について以下のように Bloom filter を構築する。
 $\text{Const}(S_i) \rightarrow \text{BF}(S_i) = [\text{BF}_i[0], \dots, \text{BF}_i[m-1]]$ (Algorithm 1).
2. P_i は楕円曲線上の ElGamal 暗号により $\text{BF}_i(S_i)$ を暗号化して O に送信する。
 $\text{PEnc}(\text{BF}_i[j]) = (U_i[j], V_i[j])$ (ただし、 $U_i[j] = r_i G$, $V_i[j] = M_i + r_i Y_i$)
3. O は暗号化された BF を以下の式で統合する。

$$\begin{aligned} & \text{PEnc}(\text{IBF}(\cap S_i)) \\ &= \sum \text{PEnc}(\text{BF}(S_i)) \\ &= [(U(0), V(0)), \dots, (U(m-1), V(m-1))] \end{aligned}$$

(ただし、 $U(j) = \sum U_i(j) \pmod{p}$, $V(j) = \sum V_i(j) \pmod{p}$)

4. O は $\mathbf{r}=[r_0, \dots, r_{m-1}]$ により $\text{PEnc}(\text{IBF}(\cap S_i))$ をランダム化する。

$$\begin{aligned} & \text{PEnc}(\mathbf{r} \text{IBF}(S_i)) \\ &= [(r_0 U(0), r_0 V(0)), \\ & \quad \dots, (r_{m-1} U(m-1), r_{m-1} V(m-1))] \end{aligned}$$

5. O はランダム化された $\text{PEnc}(\mathbf{r} \text{IBF}(S_i))$ を P_i に送信する。

Phase II: O による共通集合の計算

1. $P_i (\neq P_1)$ は部分シェアである $\mathbf{Z}_i$ を次の式で計算する。

$$\begin{aligned} \mathbf{Z}_i &= [Z_i[1], \dots, Z_i[m]] \\ &= [(x_i r_0 U(0), x_i r_0 V(0)), \\ & \quad \dots, (x_i r_{m-1} U(m-1), x_i r_{m-1} V(m-1))] \end{aligned}$$

2. P_i は $\mathbf{Z}_i$ をハイブリッド暗号 $\text{Hybrid}(Y_1, \mathbf{Z}_i)$ で暗号化

し、 O を経由して P に送信する。

3. P_1 は $\text{Hybrid}(Y_1, \mathbf{Z}_i)$ を復号して得られた部分シェア $\mathbf{Z}_i$ から完全シェア $\mathbf{Z}$ を計算する。

$$\mathbf{Z} = \sum \mathbf{Z}_i = [\sum \mathbf{Z}_i[0], \dots, \sum \mathbf{Z}_i[m-1]]$$

4. P_1 は $\mathbf{Z}$ をハイブリッド暗号 $\text{Hybrid}(Y_i, \mathbf{Z})$ で暗号化し、 O を通じて P に送信する。
5. 各 P_i は $\text{Hybrid}(Y_i, \mathbf{Z})$ を復号して得られた $\mathbf{Z}$ から $\text{IBF}(\cap S_i)$ を復号し、共通集合 $\cap S_i = \{s_1, \dots, s_h\}$ を取得する。

Phase III: 突合データの取得

1. P_i は $\cap S_i$ に含まれる突合属性と、それに付随する解析属性をそれぞれ $\text{SEnc}(K_P, s_j)$, $\text{Hybrid}(Y_c, \alpha_{i,j})$ に暗号化する。また、このとき突合属性と解析属性の関係を保つため、各暗号文をまとめて $\{\text{SEnc}(K_P, s_j) \parallel \text{Hybrid}(Y_c, \alpha_{i,j})\}_j$ とし、 O に送信する。
2. O は取得した暗号文 $\{\text{SEnc}(K_P, s_j) \parallel \text{Hybrid}(Y_c, \alpha_{i,j})\}_{i,j}$ から突合属性について n 個の同一暗号文を識別し、同一暗号文が n 個存在しない場合はその暗号文を突合対象から除外する。
3. O は突合属性の暗号文 $\text{SEnc}(K_P, s_i)$ から同一暗号文について付随する解析属性の暗号文 $\text{Hybrid}(Y_c, \alpha_{i,j})$ を統合し、データの順序に対してシャッフル処理を行う。
4. C は解析属性の暗号文を復号し、統合データ $\{\alpha_{i_1,j} \parallel \dots \parallel \alpha_{i_n,j}\}$ を得る。ここで、各データはシャッフルされているため、どの機関がどの情報を保有していたかは C に秘匿される。

5. 性能評価

本章では提案プロトコルの実行結果について記載する。プログラミング言語として Python3 を利用し、Bloom filter に使用するハッシュ関数には PyCryptodome(PyCrypt) ライブラリの SHA-256 を用いた。使用する k 個のハッシュ関数はそれぞれ独立でなければならないため、 k 個の独立な乱数をソルトとして使用した。また、実験は単一のデスクトップ PC で行われ、CPU は Intel(R) Core(TM) i7-9700 3.00GHz、メモリは 16GB である。楕円曲線に使用するパラメータは NIST P-256 を使用し、実行時間はすべて各エンティティ間の通信に必要な時間を除いている。

図 5 はデータ数 $10^3, 10^4, 10^5$ において各機関の実行時間を示している。FPR は各々 0.0065 に設定した。本プロトコルにおける各エンティティの計算量は BF 長に依存するので、BF 長を最小にするためにハッシュ関数の個数は $k = 7$ とした。また、図 6 は機関数 $n = 3, 11, 18, 21$ における実行時間を示している。この時、各機関が所有するデータ数とその共通集合の要素数はともに 10^3 個である。これらより、 C の計算量が P_i や O と比較して少ないこと、 P_i の計算量が機関数 n に依存しないことが確認できる。ま

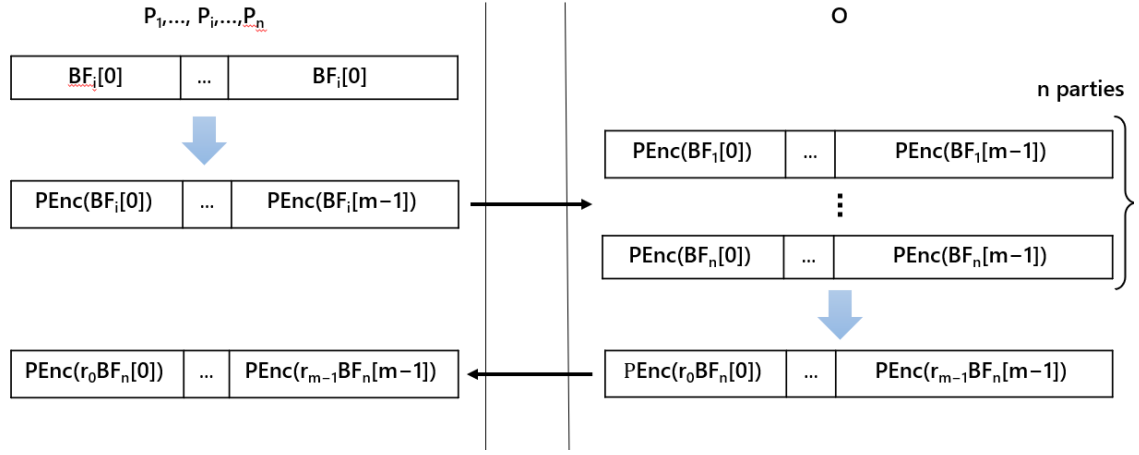


图 2 Phase I

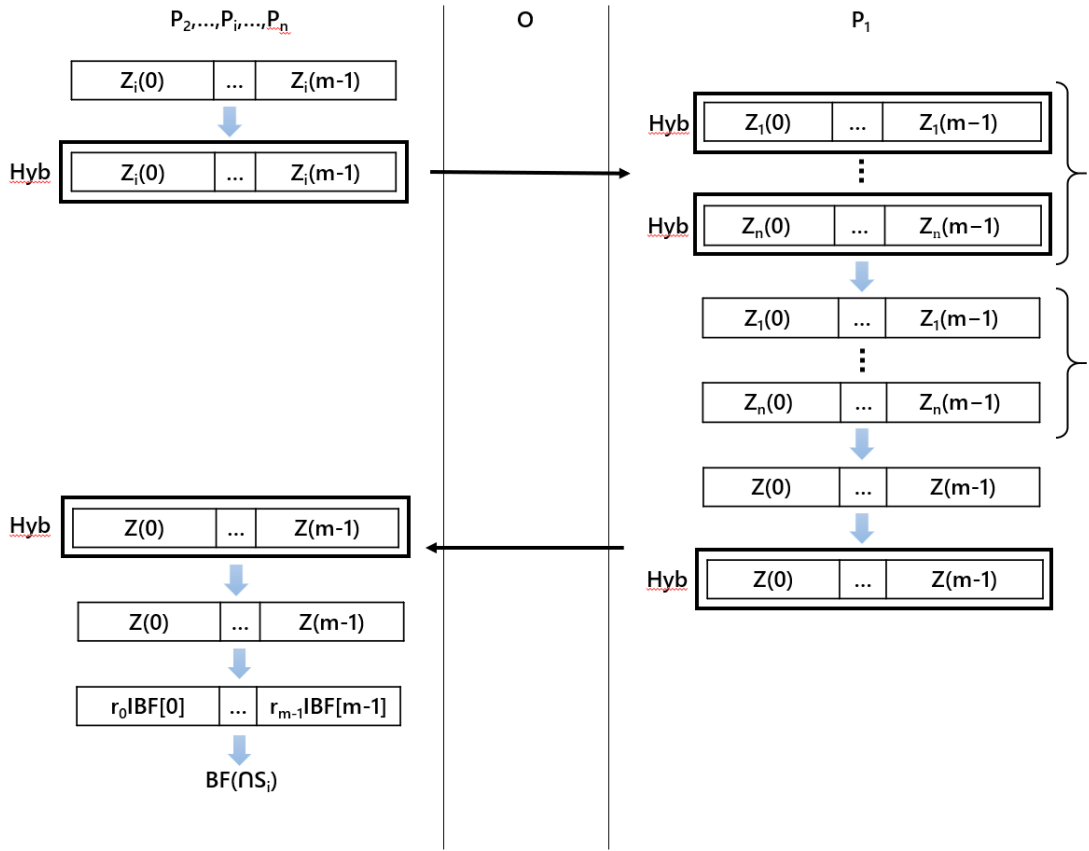


图 3 Phase II

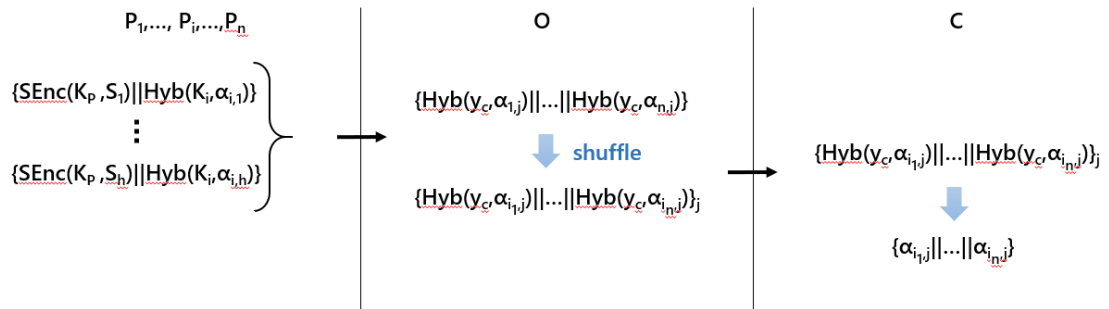


图 4 Phase III

た、表 3 は得られる突合データが偽陽性によらないことを示している。P₁, P₂, P₃ はそれぞれ 5×10^3 , 10^4 , 10^5 個のデータを持つ機関であり、その共通集合は 10^4 個である。この時、ハッシュ関数の個数は FPR が最小になるよう設定している。BF 長が小さい、つまり FPR が大きくなると各機関が得る共通集合の個数*1は本来の値より多くなるが、突合データの数には常に 10^3 個のままであり、偽陽性により誤った共通集合の数を得ることになっても突合データに影響がないことを示している。

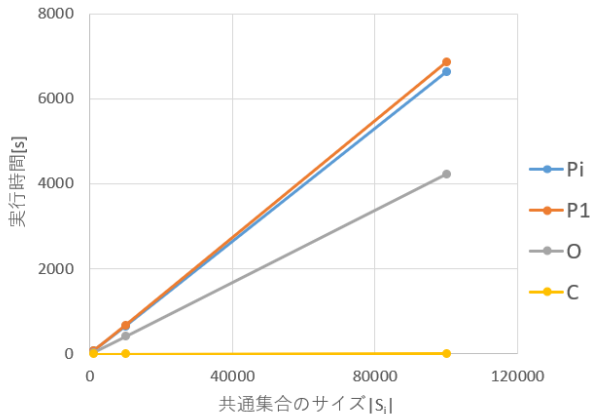


図 5 共通集合のサイズと実行時間の関係

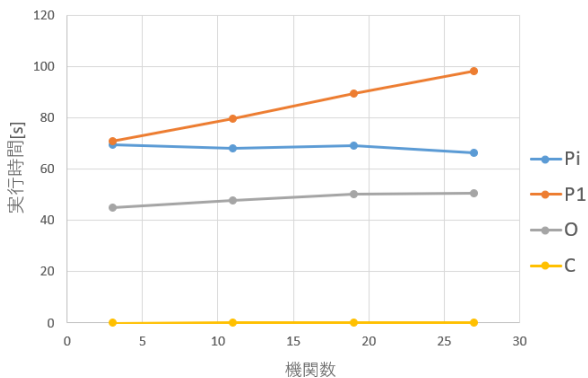


図 6 機関数と実行時間の関係

表 3 偽陽性による共通集合の個数と突合データ数

BF 長	共通集合の要素数 $ \cap S $			突合データ数		
	10^4	2×10^4	4×10^4	10^4	2×10^4	4×10^4
P ₁	3536	2544	1793	1000	1000	1000
P ₂	4551	2937	1963	1000	1000	1000
P ₃	28343	11860	5254	1000	1000	1000

*1 MPSI により取得した共通集合の個数。

6. まとめ

本論文では、データを突合属性、解析属性などの属性に分類し、それぞれの属性に応じてデータを適切に秘匿する突合プロトコルを提案した。本プロトコルはデータ構造として利用する Bloom filter の偽陽性によらず C が必ず突合データを取得でき (耐障害性)、かつ本プロトコルはデータの種類・サイズによらない (拡張性)。今回の提案ではより実環境に近いモデルとして P2P 通信を前提とせず、外部機関を導入している。データを適切に暗号化することにより、外部機関は共通集合の個数以外の情報を得ることはできず、シンプルなクライアントサーバモデルで安全に実現することができる。

6.1 参考文献・謝辞

謝辞 本研究の一部は文部科学省「Society5.0 に対応した高度技術人材育成事業成長分野を支える情報技術人材の育成拠点の形成 (enPiT)」さらに文部科学省の平成 30 年度「Society 5.0 実現化研究拠点支援事業」の助成を受けています。

参考文献

- [1] A. Miyaji, K. Nakasho, and S. Nishida. Privacy-preserving integration of medical data - A practical multiparty private set intersection. *J. Medical Systems*, 41(3):37:1–37:10, 2017.
- [2] A. Miyaji and Y. Nagao. Research on privacy preserving data retrieval by using psi.
- [3] A. Miyaji and Y. Nagao. Privacy preserving data integration protocol.
- [4] B. Pinkas et al. Efficient circuit-based PSI with linear communication. In *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 122–153. Springer.
- [5] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Psi from paxos: Fast, malicious private set intersection. *Cryptology ePrint Archive*, Report 2020/193, 2020. <https://eprint.iacr.org/2020/193>.
- [6] M. Ion et al. On deploying secure computing commercially: Private intersection-sum protocols and their business applications. *IACR Cryptology ePrint Archive*, 2019.
- [7] R. Egert et al. Privately computing set-union and set-intersection cardinality via bloom filters. In *ACISP 2015*, volume 9144 of *LNCS*, pages 413–430. Springer, 2015.
- [8] L. Kissner and D. X. Song. Privacy-preserving set operations. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 241–257. Springer, 2005.
- [9] D. Many, M. Burkhart, and X. Dimitropoulos. Fast private set operations with sepia, 2012.
- [10] Y. Sang and H. Shen. Efficient and secure protocols for privacy-preserving set operations. *ACM Trans. Inf. Syst. Secur.*, 13(1):9:1–9:35, 2009.
- [11] Atsuko Miyaji and Tomoaki Mimoto. *Security Infrastructure Technology for Integrated Utilization of Big Data Applied to the Living Safety and Medical Fields: Applied to the Living Safety and Medical Fields*. 01 2020.