Computer Security Symposium 2014
22 - 24 October 2014

効率的な多機関の Private Set Intersection

西田 昌平 †　　宮地 充子 †

† 北陸先端科学技術大学院大学
923-1292 石川県能美市旭台 1-1
{s1310053,miyajig}@jaist.ac.jp

あらまし データ集合を持つ通信者同士が互いにその内容を開示することなく，共通する要素に関する情報だけを得るプロトコルを，Private Set Intersection(PSI) Protocol という．多機関における PSI は 2 機関の PSI と異なり，全機関の SI や 2 機関以上に共通する要素など様々に出力を拡張できることが望まれる．しかし，既存研究で提案された Multi-Party PSI プロトコルでは後者が実現されていなかった．そこで本研究では，多機関の各種 SI で最終データについて各プレイヤーが入力したデータに関する情報以外は秘匿して出力する方法を提案する．

Efficient Multi-Party Private Set Intersection Protocol

Shohei Nishida†　　Atsuko Miyaji†

†Japan Advanced Institute of Science and Technology
923-1292，Asahidai 1-1，Nomi-shi，Ishikawa，Japan
{s1310053,miyaji}@jaist.ac.jp

Abstract Private Set Intersection Protocol(PSI) is necessary when several players with each data sets want to compute some information about their common data without disclosing them. PSI among more than two parties is called a Multi-Party PSI. Previous PSI by L. Kissner and D. Song is limited to compute all-set intersections with the same number of sets w and with $O(n^2w+nw^2)$ complexity where n is the number of the players, w is the data size. In this paper, we propose a Multi-Party PSI among players with any data size, which can compute any set arithmetic with $O(w^2n^2)$.

1　はじめに

近年の急速な情報化社会の進展に伴い，様々なエンティティとの通信が必要不可欠となってきている．そのような通信において個人情報等が含まれた重要な情報をやり取りする際，データ漏洩の防止とプライバシー保護の技術は必須である．そこで近年その技術として Private Set Intersection (PSI) という通信プロトコルが注目されている．PSI とは，異なるデータを持つクライアントとサーバのデータの積集合を求めるプロトコルである．このプロトコルに従うことにより通信者の持つ重要な情報を外部に漏らすことなく，必要な情報のみやり取りを行うことが可能である．例えばデータマイニングの場などにおいて，ある重要な情報の集合をもつ多数の機関が，その複数機関の間で共通するデータ集合の要素に関する情報を得たい場合などに PSI は有効である．PSI プロトコルに従って通信を行うことで，それぞれの機関が入力したデータ集合に関する情報は保護される．既存研究では，様々な PSI プロトコルが提案されている．

通信者の入力データ集合の要素数も秘匿する方式 [ACT11] や，プロトコルに参加するプレイヤーに任意の行動をする者がいた場合でも安全に通信をおこなえる方式 [CKT10]，複数者間での通信を可能とした方式 [KS07] などがある．

本研究では既存の Multi-Party プロトコル [KS07] における問題点を考え，それを補うプロトコルの構築を目指す．既存のプロトコルでは全プレイヤーの入力データ数を一致させなければならないという制限があるのに対し，本研究ではプロトコルに参加するプレイヤー数や，入力のデータ数に制限や条件はない．さらに本提案方式では Bloom Filter という空間効率の良い確率的データ構造を用いることで既存手法と比べ，計算量・通信量なども削減されている．このプロトコルの更なる特徴として，参加プレイヤーの全データ集合のうち，任意の人数以上に共通する要素に関する情報を出力することができる．

本稿は 2 章で，取り扱う記号の表記，攻撃者モデルやプロトコルを構成する技術について述べる．続く 3 章で既存研究で提案された比較対象である Malti-Party PSI プロトコルについて説明した後，4 章で提案プロトコルの概要を述べ，5 章でその評価を行う．最後に 6 章で成果や今後の展望についてまとめる．

2 準備

2.1 表記

ここでは本稿で扱う記号について説明する．以後特に記述がない場合は以下の表記を用いるものとする．

- n : プレイヤー数
- P_i : プレイヤー i
- $S_i = \{s_{i,1}, s_{i,2}, ..., s_{i,w_i}\}$: プレイヤー i が持つデータ集合．各プレイヤーのデータ数 w_i は任意である．
- m : Bloom Filter のサイズ
- k : Bloom Filter に用いるハッシュ関数の数
- $H = \{h_0, ..., h_{k-1}\}$: Bloom Filter に用いるハッシュ関数群
- E_{pai} : Paillier 暗号の暗号化
- D_{pai} : Paillier 暗号の復号処理
- $E_{pk,i}$: プレイヤー i の公開鍵 pk を用いた暗号化
- $D_{sk,i}$: プレイヤー i の秘密鍵 sk を用いた復号処理
- BF_i : プレイヤー i が生成した Bloom Filter
- $\mathrm{BF}_i[j](0 \leq j < m)$: プレイヤー i が生成した Bloom Filter の配列位置 j に格納されている数
- IBF : 統合フィルター
- $\mathrm{IBF}[j](0 \leq j < m)$: 統合フィルターの配列位置 j に含まれる数
- CBF : 比較フィルター
- RF_i : Reduced Filter
- $\cap^d S$: 参加プレイヤーの全データ集合中 $d(\leq n)$ 人が所持している要素の集合
- $\cap^d S_i$: 参加プレイヤーの全データ集合中 $d(\leq n)$ 人が所持している要素の内，プレイヤー i が所持しているデータの集合

2.2 攻撃者モデル

PSI プロトコルには，Semi-honest adversary model と Malicious adversary model という二種類の攻撃者モデルがある．Semi-honest adversary model は攻撃者がプロトコルに従うことを前提とし，得られる情報から更なる情報を得ようと試みる．Malicious adversary model は攻撃者がプロトコルを任意に逸脱をする可能性があることを前提としたモデルであり，攻撃者はプロトコル通りの入力をしない，途中で通信を切断するといった任意の振る舞いをする可能性がある．

2.3 Paillier 暗号

本研究で提案するプロトコルと，比較対象である既存研究 [KS07] のプロトコルで使用する Paillier 暗号について説明する．Paillier 暗号とは加法準同型性を有する準同型暗号である．準同型暗号とは，性質として暗号文 $E(m_1)$, $E(m_2)$ が与えられたときに，平文 m_1, m_2 や秘密鍵の情報なしで $E(m_1 \circ m_2)$ を計算できる．ここで，$\circ$ とは加法や乗法のような二項演算子とする．Paillier 暗号の暗号方式は以下のアルゴリズムで構成する．

鍵生成 大きな素数 p, q をランダムに選び，$n = pq$ とする．次に $x \in \mathbb{Z}_n$ をランダムに選び，$g = 1 + xn \mod n^2$ を計算する．$pk = (n, g)$，$sk = (p, q)$ をそれぞれ公開鍵，秘密鍵として出力する．

暗号 $m \in \mathbb{Z}_n$ を暗号化するとき，次のように計算する．$r \leftarrow \mathbb{Z}^*_{n^2}$ を選ぶ．次に暗号文 $c = g^m \cdot r^n \mod n^2$ を出力する．

復号 暗号文の復号は次を計算する．$m = L(m^\lambda \mod n^2)/L(g^\lambda \mod n^2) \mod n$ ここで，$\lambda = \mathrm{lcm}(p-1, q-1)$，$L(g^\lambda \mod n^2)$，関数 $L(u) = (u-1)/n$ である．

2.4 Bloom Filter

ブルームフィルタ [B70] は要素が集合の一部かどうかの確認をする場合などに利用され，必要な領域が要素の大きさに依存しない非常に効率のよい確率的データ構造である．ブルームフィルタは，m ビットの配列と，$[0, ..., m-1]$ にマッピングする k 個のハッシュ関数 $h_0, ..., h_{k-1}$ からなるハッシュ関数群 H から構成される．データは次のように格納される．まず用意した m ビットの配列の全てに 0 を格納する．次に図 1 にあるようにデータ集合 S の要素を $h_0, ..., h_{k-1}$ を用いてマッピングする．その後得られた値と同じ配列内に 1 を格納する．ハッシュ関数は一意にマッピングを行うため，確認したい要素に対して同じハッシュ関数を用いて，全ての要素が 1 であることを確認すればよい．また，Bloom Filter の特性として同じサイズのデータ構造と

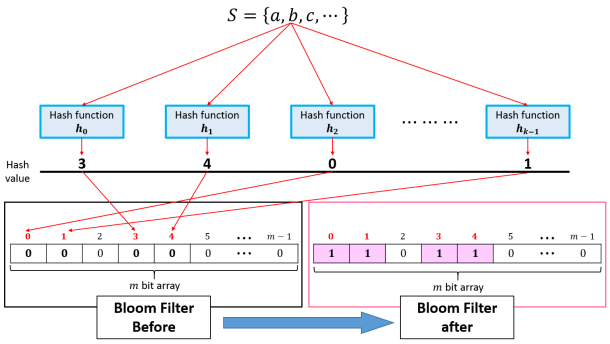

図 1: Bloom Filter での要素の登録

同じハッシュ関数群を持つフィルター同士の論理和と論理積はそれぞれビット単位の OR と AND で実現できる．なおブルームフィルタには偽陰性はないが偽陽性がある．偽陰性とは実際に存在するデータについて存在するかどうかを確認した場合に，存在しないという誤った結果が出ることである．一方で偽陽性とは実際に存在しないデータについて存在するかどうかを確認した場合に，存在するという誤った結果が出ることである．ブルームフィルタでは m ビット配列を用いるが，その任意の個所に 1 が格納される確率は $1/m$ であり，要素の数を w とすると，偽陽性が生じる確率 p_1 は次のようになる．

$$\begin{aligned} p_1 &= \left\{1 - (1 - \frac{1}{m})^{kw}\right\}^k \\ &\approx \left\{1 - e^{-kw/m}\right\}^k \end{aligned}$$

上記より，明らかに m が大きければ偽陽性確率は低くなり，逆に w が大きければ偽陽性確率は高くなることが分かる．これより，k と m,w の関係は以下のように表せる．

$$k = \frac{m}{w} \ln 2$$

上記の式を使い，要素数 w と Bloom Filter のサイズ m が決まっているとき，偽陽性の発生を最小に抑える最適なハッシュ関数の個数 k を求めることができる．

3　既存プロトコル [KS07]

3.1　プロトコル概要

次に比較対象となる Multi-Party PSI を紹介する．プレイヤーは P_1,..., P_n であり，それぞれが $S_i = \{s_{i,\,1}, s_{i,\,2}, ..., s_{i,\,w_i}\}$ をデータ集合として持ち，$\forall_{i\in[n],\,j\in[w_i]} s_{i,\,j} \in U$ とする．次に Paillier 暗号での暗号化を $E : U \to V$，つまり $\forall x \in U$ に対して $E_{pai}(x) \in V$ とする．V は U を含む十分に広いドメインであり，ランダムな $a \in V$ を選んだとき，それが $a \in U$ である確率は無視出来るほど小さい．またここで用いる Paillier 暗号の復号には全プレイヤーが参加する必要がある．

3.2　アルゴリズム

まず各プレイヤー $P_i(i = 1, ..., n)$ は Paillier 暗号の公開鍵に対応する秘密鍵を分散して持つ．続いて $f_i = (x - (s_{i,1})) ... (x - (s_{i,w_i}))$ を生成し，各係数を Paillier 暗号を用いて $E_{pai}(f_i)$ としてから $P_{i+1}, ..., P_{i+c}$ に送る．その後 $c+1$ 個の k 次多項式 $r_{i,0}, ..., r_{i,c}$ をランダムに生成する．但し，これらの多項式の係数 $\forall c$ に対して $c \in V$ とする．さらに各プレイヤー P_i は $\Phi_i = f_{i-c} * r_{i,c} + ... + f_{i-1} * r_{i,1} + f_i * r_{i,0}$ を計算する．

続いて，P_1 は $\lambda_1 = \Phi_1$ を P_2 に送る．その後 $P_i(i = 2, ..., n)$ は P_{i-1} から送られた λ_{i-1} を用いて $\lambda_i = \lambda_{i-1} + \Phi_i$ を計算し，$P_{i+1 \mod n}$ に送る．P_1 が $p = \lambda_n = \sum_{i=1}^{n} f_i * \left(\sum_{j=0}^{c} r_{i+j,j}\right)$ を受け取ると，それを全てのプレイヤーに送る．

最後に全てのプレイヤーで p の復号を行い，自らの入力を引数として与える．出力が 0 となれば，その引数が共通集合である．

3.3　問題点

既存プロトコルにおける問題点は，プロトコルに参加する全プレイヤーの入力データ数を一致させる必要がある点である．つまり，あるプレイヤーは他のプレイヤーの入力セットのサイズを知ることができる．また，各プレイヤーの多項式化した入力を複数人のプレイヤーに送信する必要があるので，計算量・通信量も大きくなる．

4　提案プロトコル

4.1　プロトコル概要

提案プロトコルは複数の参加プレイヤーである Clients(C) と Trusted Third Party(TTP) である Server(S) 間で行われ，参加プレイヤー数に関して制限や条件は無く，各プレイヤー P_i はそれぞれデータ集合 $S_i = \{s_{i,1}, ..., s_{i,w_i}\}(w_i$ は任意) を入力とし，このプロトコルの実行後 P_i は参加プレイヤーの全データ集合中 $d(\leq n)$ 人が所持している要素の集合 $\cap^d S$ の内，プレイヤー i が所持しているデータの集合 $\cap^d S_i$ を得ることができる．

この提案プロトコルは，Bloom Filter 生成，フィルター統合，Intersection 計算の 3 つのフェーズからなる．Bloom Filter 生成フェーズでは，各プレイヤー i は入力データから Bloom Filter を生成し Server に送信する．フィルター統合フェーズでは，Server が Clients から送られてきた Bloom Filter を配列毎に加算し統合フィルター IBF を出力する．また統合フィルターで d 以上の数が格納されている配列内の数を “1” に置き換え，それ以外を “0” に置き換えたフィルターを比較フィルター CBF として出力し各プレイヤー i に送信する．Intersection 計算フェーズでは，受け取った CBF から $\cap^d S_i$ を計算する．

4.2　プロトコル構成要素

Bloom Filter の生成について

各プレイヤーが Bloom Filter の生成時に使用するハッシュ関数群 H は Rndom Oracle Model における理想的な暗号学的ハッシュ関数であると仮定する．つまり，原像計算困難性，第 2 原像計算困難性，衝突困難性の性質をもつハッシュ関数とする．

公開鍵暗号の安全性について

Clients は Bloom Filter 生成フェーズで，Server はフィルター統合フェーズで，それ

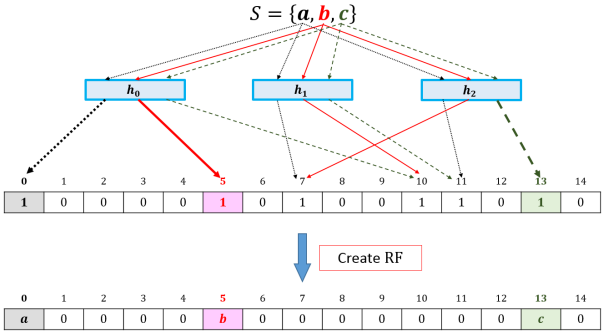

図 2: RF_i の生成

ぞれデータを送信する際に強秘匿性を満たす公開鍵暗号を用いると仮定する．

RF_i : Reduced Filter

Bloom Filter の偽陽性発生確率 p_1 の式から，フィルターに登録する各要素に対して，他の要素と重複が発生しないような “1” が格納されている配列を持つ確率 p_2 は以下のように表すことができる．また Bloom Filter 生成フェーズにおいて，図 2 のようにその配列内に該当する要素を格納し，他の配列には “0” を格納したフィルター Reduced Filter を生成する．Intersection 計算フェーズで RF_i と CBF の AND を取ることで，$\cap^d S_i$ が計算できる．

$$\begin{aligned} p_2 &= (1 - \frac{1}{m})^{(k-1)w} \\ &\approx e^{(1-k)w/m} \end{aligned}$$

4.3 アルゴリズム

前提条件として，通信は公開ネットワーク上で行われ，Bloom Filter 生成に必要な各パレメータ (m, k, H) はプロトコル参加プレイヤー間で共通の知識とする．Bloom Filter 生成パラメータは偽陽性発生確率である確率 p_1 を低くしつつ，RF_i を生成できるようにするため確率 p_2 が高くなるように設定する．また，各プレイヤー i は RF_i が生成できることを前提とする．図 3,4 に $n = 5$ としたときの提案プロトコルの流れを示す．

Bloom Filter 生成 (Clients)

1. $\mathrm{BF}_i[j] = 0$ $(j = 0, ..., m-1)$
2. $\mathrm{BF}_i[h_l(S_i)] = 1$ $(l = 0, ..., k-1)$
3. Construct RF_i
4. $C \to S : E_{pai}(\mathrm{BF}_i[j])$ $(j = 0, ..., m-1)$

フィルター統合 (Server)

1. Compute $E_{pai}(\mathrm{IBF}[j]) = \sum_{i=1}^{n} E_{pai}(\mathrm{BF}_i[j]) = E_{pai}(\sum_{i=1}^{n} \mathrm{BF}_i[j])$ $(j = 0, ..., m-1)$
2. $\mathrm{IBF}[j] = D_{pai}(E_{pai}(\mathrm{IBF}[j]))$ then Output $\mathrm{IBF}(j = 0, ..., m-1)$
3. if $d \leq \mathrm{IBF}[j]$, $\mathrm{IBF}[j]$ Replace “1”, else $\mathrm{IBF}[j]$ Replace “0” then Output $\mathrm{CBF}(j = 0, ..., m-1)$
4. $S \to C : E_{pk,i}(\mathrm{CBF})$

Intersection 計算 (Clients)

1. $D_{sk,i}(E_{pk,i}(\mathrm{CBF}))$
2. Compute $\mathrm{RF}_i \wedge \mathrm{CBF}$, Output $\cap^d S_i$

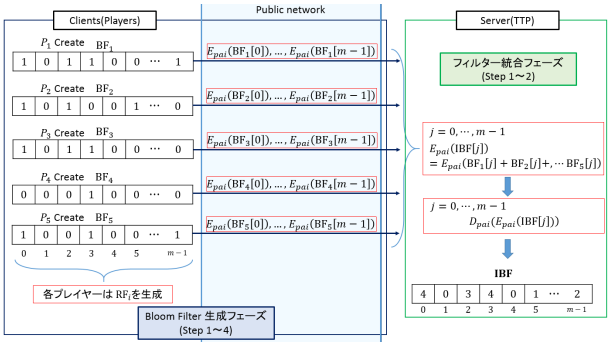

図 3: 提案プロトコル 1

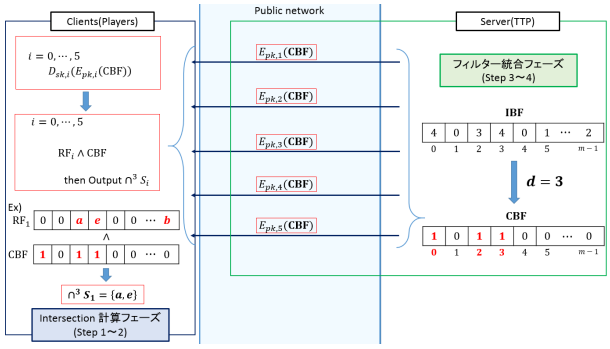

図 4: 提案プロトコル 2

4.4 議論

このプロトコルはBloom Filterベースの PSIであり，参加プレイヤー数に制約はなく，プレイヤーが入力するデータ数も任意でよい．またBloom Filterを用いることで，Clientsにも Serverにも入力データ数は漏れない．$d = n$ であるとき，既存の Multi-party PSIと同等の出力を得ることができる．

Bloom Filter生成フェーズで各プレイヤーは RF_i を保存し，フィルター統合フェーズで Serverは IBFを保存しておくことにより，$d(\leq n)$ の閾値を変えて再度 Intersectionを得ようとしたときに，もう一度プロトコルを始めから実行する必要はなく，フィルター統合フェーズの Step 3から開始すれば良いことが分かる．また，メンバーの追加も容易であり，新たにプロトコルに参加した Clientは Bloom Filter生成フェーズをおこない，Serverは受信したフィルターを統合フィルターに追加し IBFを更新すればよい．以上の点が既存の PSIと大きく異なる部分である．

5 評価

5.1 安全性

定理 (プロトコルの安全性)

提案するプロトコルは，Semi-honest adversary modelにおいて，S_i，$|S_i|$，$\cap^d S \backslash \cap^d S_i$，$|\cap^d S|$，これらが保護される．

証明 (プロトコルの安全性)

Intersection計算フェーズにおいて，プレイヤー i が受け取った CBFからは，Bloom Filterの性質と，ランダムオラクルモデルにおいてハッシュ関数群 H が暗号学的ハッシュ関数の安全性を満たすことから，$\cap^d S_i$ 以外の情報は得られない．

5.2 計算量・通信量

各フェーズにおける計算量は以下の通りである．ここでは，計算を簡単にするため各プレイヤーのデータ集合のサイズを w で統一する．

Bloom Filter生成 (Clientsの計算量・通信量)

各プレイヤーは Bloom Filterの生成にデータ集合の要素 w 個を k 回ハッシュ関数に入力するので，kw の計算量がかかる．通信量に関しては，Bloom Filterを送信するので，フィルターサイズである m を要する．

・計算量

kw

・通信量

m

フィルター統合 (Serverの計算量・通信量)

Serverは n 個のフィルターに対して領域毎の加算を m 回おこない統合フィルターを生成する．次に統合フィルターの領域内を全てチェック (m 回) して，比較フィルターを生成する．通信量に関しては，各プレイヤー n 人に CBFを送信するので，nm を要する．

・計算量

$(n-1)m + m$

・通信量

nm

Intersection出力 (Clientsの計算量)

各プレイヤーは CBFと RF_i を配列ごとに論理積するので，m 回の計算量を要する．

・計算量

m

以上より，Clientsの計算量は

$$k = \frac{m}{w}\ln 2$$

の式を用いて，

$$\begin{aligned} & kw + m \\ = \ & w\frac{m}{w}\ln 2 + m \\ = \ & m\ln 2 + m \end{aligned}$$

以下に上記計算量と通信量をまとめた表を示す．

5.3 比較

既存研究 [KS07] とのプロトコル参加プレイヤーの計算量・通信量の比較，プライバシーに

表 1: 提案プロトコル計算量・通信量

	計算量	通信量
Server	$O(nm)$	$O(nm)$
Clients	$O(m)$	$O(m)$

関する評価などを以下に示す．既存研究では方式・手法が異なるため，通信量が一致する ($m = w^2n^2$) と仮定し，計算量の比較を行う．ここで，n はプレイヤー数，m は Bloom Filter のサイズ，k は Bloom Filter 生成時に使用するハッシュ関数の数，w はデータ集合のサイズである．

表 2: 計算量・通信量に関する評価

	計算量	通信量
既存研究 [KS07]	$O(n^2w + nw^2)$	$O(w^2n^2)$
提案プロトコル	$O(w^2n^2)$	$O(w^2n^2)$

表 3: プライバシーに関する評価

	入力セットサイズ	漏れる情報
既存研究 [KS07]	$\|S_i\| =, ..., = \|S_n\|$	$\|S_i\|, ..., \|S_n\|$
提案プロトコル	任意のサイズ	なし

6 まとめ

6.1 成果

入力データサイズ，プロトコル参加プレイヤー数に制約がない Multi-Party PSI Protocol を提案した．この提案プロトコルは参加プレイヤーの全データ集合中 $d(\leq n)$ 人が所持している要素の内，各プレイヤーが所持しているデータを得ることができる．また，公開ネットワークでの通信，出力結果，Server や Clients が受け取るデータから漏れる情報はない．

6.2 今後の展望

- 統合フィルターから比較フィルターを生成する段階において，暗号化したまま $d(\leq x)$ 以上の数が格納されている配列を見つける手段の考案．
- 計算量・通信量の更なる削減

参考文献

[B70] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. Commun. ACM, 13(7):422–426, 1970.

[ACT11] G. Ateniese, E. De Cristofaro, and G. Tsudik. (if) sizematters： Size-hiding private set intersection. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi,editors,Public Key Cryptography U PKC 2011,volume 6571 ofLecture Notes in Computer Science,pages 156–173. Springer Berlin / Heidelberg, 2011.

[KS07] L. Kissner and D. Song. Privacy-Preserving Set Operations. In CRYPTO 2005, Springer-Verlag (LNCS 3621), pages 241-257, 2007.

[CKT10] E. De Cristofaro, J. Kim, and G. Tsudik. Linear-complexity private set intersection protocols secure in malicious model. In ASIACRYPT, 2010.