情報処理学会論文誌 Vol.61 No.6 1175–1189 (June 2020)

推薦論文

匿名加工をともなう2パーティ秘匿クロス集計の完全準同型暗号による実現と性能比較

片山 源太郎[1,a)] 吉浦 裕[1]

受付日 2019年7月5日, 採録日 2020年3月13日

概要：複数の組織が持つデータを統合して利用することで，単一組織のみでは困難なサービスの実現が期待できる．このような組織間データ利用において，データのプライバシ保護は必要不可欠である．本稿では特に匿名加工をともなう 2 パーティ秘匿クロス集計を取り上げ，通信量の削減が期待できる完全準同型暗号を用いた方式を提案した．このとき，完全準同型暗号を用いることによる計算量の増加をハッシングや加法準同型暗号へのパッキングなどの工夫により抑制した．そして，先行研究の方式と提案方式との理論比較および実装比較を行った．理論比較では，計算量のオーダを変えずに通信量のオーダから支配的な変数を除去することができた．実装比較では，総実行時間について LAN 環境において 86.1%，WAN 環境において 77.7%に削減することができた．

キーワード：完全準同型暗号，加法準同型暗号，秘匿積集合，クロス集計

Performance Comparison of Two-Party Secure Anonymized Cross Tabulation Protocol with Fully Homomorphic Encryption

Gentaro Katayama[1,a)] Hiroshi Yoshiura[1]

Received: July 5, 2019, Accepted: March 13, 2020

Abstract: Inter-organization data mining is important because it allows an organization to develop services that cannot be developed by a single organization. However inter-organization data mining has privacy issues. This paper discusses two-party secure anonymized cross tabulation protocol and proposes a new protocol with fully homomorphic encryption to reduce communication complexity. The proposed method can remove a dominant parameter from communication complexity of a representative previous method while maintaining the computational complexity. It was implemented and evaluated to show that it decreased the total execution time of the previous method to its 86.1% in LAN environment and to its 77.7% in WAN environment.

Keywords: homomorphic ecnryption, private set intersection, cross tabulation

1. まえがき

複数の組織が持つデータを統合して利用する組織間データ利用は，単一組織のみでは知り得ないデータの利用を可能にし，単一組織では困難なサービスや製品の実現を可能とする．たとえば，性別や住所などの個人の属性情報を持つ組織と商品の販売情報を持つ小売店が協力し，共通の ID を用いて名寄せしたうえで解析を行い，個人へ商品を推薦するサービス [1] がある．

組織どうしが互いを信頼して自組織のデータを他方の組織に提供することができれば，組織間データ利用は容易に実現できる．しかし，現実には組織どうしが互いに信頼できるとは限らない．たとえば，他方の組織による情報漏洩・不正利用の懸念がある場合やデータの内容（個人情報や企

[1] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[a)] g.katayama@uec.ac.jp

本稿の内容は 2019 年 3 月の第 84 回 CSEC 研究発表会にて報告され，同研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

© 2020 Information Processing Society of Japan

業機密）により第三者へ提供することが適当でない場合などがある．このため，互いを信頼できない組織どうしでは互いのデータを秘匿して利用することが求められる．さらに，データの統合結果を一方の組織に開示すると，統合結果から他方の組織のデータの一部が推定されるおそれがある．これを防止するため，データの利用と統合結果への匿名加工をデータ秘匿状態で行い，匿名加工後の統合結果のみを一方または両方の組織に開示する方式が必要となる．

このような匿名加工をともなう組織間の秘匿データ利用のなかでも，本稿ではクロス集計（2章で詳述）を取り上げる．クロス集計はナイーブベイズ法や決定木分析の前段階で使用され，推薦アルゴリズム [1] や識別アルゴリズム [2] などで用いられる基本的かつ重要な集計法である．また，組織間秘匿クロス集計には2パーティや3パーティ以上の集計があるが，本稿では2パーティの秘匿クロス集計を取り上げる．

匿名加工をともなう2パーティ秘匿クロス集計は千田ら [3] によって，提案されている．千田ら [3] の方式では，通信量のオーダが2つの組織がそれぞれ持つデータのレコード数に依存する．例として，顧客情報管理組織が日本人の数割程度の属性情報を持ち，小売店は来店した顧客に関する販売情報のみを持つような場合を考える．このとき，顧客情報管理組織が持つレコード数と小売店が持つレコード数には大きな差が生じるが，千田ら [3] の方式では，通信量が顧客情報管理組織が持つレコード数に依存するため，通信がクリティカルな設定では総実行時間が大きくなる．

現実のユースケースのなかには，秘匿クロス集計を行う2パーティのうち一方の組織がIDの発行や管理を行う場合がある．このように一方の組織がID管理組織である場合，他方の組織が持つIDはID管理組織の持つIDの部分集合であることが自明であるため，IDの共通集合が他方の組織に漏洩しても実用上問題がない．

そこで，本稿ではこのようなユースケースを想定して安全性の要件を緩めたうえで，通信がクリティカルな設定において効率的な匿名加工をともなう2パーティ秘匿クロス集計の方式を提案する．提案方式では，秘匿計算の過程において通信量の削減が期待できる完全準同型暗号 [4] を用いる．秘匿計算にはGarbled Circuit [5] や秘密分散に基づく秘匿計算 [6]，完全準同型暗号などがあるが，なかでも完全準同型暗号はGarbled Circuitや秘密分散に基づく秘匿計算とは異なり，乗算やその他の操作の過程において通信が発生しないという特徴を持ち，通信量の削減が期待できる．一方，完全準同型暗号は暗号文のサイズが大きいことや乗算やその他の操作の計算量が大きいといった問題を持つ．本稿では，完全準同型暗号を用いることによる計算量の増加を抑制したうえで，通信量の削減をはかる．そして，千田らの方式 [3] との理論比較および実装比較を行う．比較では，提案方式と千田らの方式の安全性を揃えたうえで公平な比較を行う．

2. 匿名加工をともなう2パーティ秘匿クロス集計問題

2つの組織が互いのデータを組み合わせてクロス集計表を算出し，匿名加工するまでを秘匿計算し，匿名加工後のクロス集計表を一方または両方の組織に開示するという問題を匿名加工をともなう2パーティ秘匿クロス集計問題と呼ぶ．その例を**図 1** に示す．ここで，クロス集計とはデータの複数の属性に着目し，着目した属性に関して，それらの属性値の組合せに該当する度数を集計することをいう．たとえば図1のように性別，年齢（顧客情報管理組織のデータ）と商品A，商品B，商品Cの購入の有無（小売店のデータ）に着目し，クロス集計すると図1に示すクロス集計表のようになる．ここで，クロス集計は各組織のID集合の共通部分を用いて名寄せすることにより実現する．この問題の適切な実装や実行時間は2つの組織の持つデータのサイズや形式に依存する．

そこで，本稿ではクロス集計のユースケースとして，山口らの組織間プライバシ保護推薦システム [1] におけるデータのサイズや形式を用いる．プレーヤーやデータテーブル，クロス集計表，匿名加工後のクロス集計表の例は図1のとおりである．2つの組織として顧客情報管理組織と小売店を考え，各組織のパラメータとその大きさを**表 1** に示す．顧客情報管理組織（組織 A）は1千万人（N_A）の性別，年齢，都道府県からなるデータを持ち，それぞれ2カ

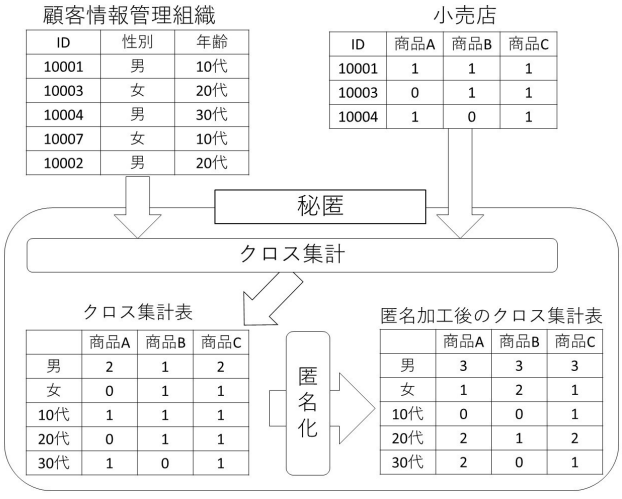


図 1 匿名加工をともなう2パーティの秘匿クロス集計
Fig. 1 Two-party secure anonymized cross tabulation.

表 1 各組織のパラメータとその大きさ
Table 1 Parameters of each organization.

顧客情報管理組織（組織 A）	ユーザ数	10,000,000（N_A）
	属性値総計	57（AT_A）
小売店（組織 B）	ユーザ数	10,000（N_B）
	属性値総計	10,000（AT_B）

© 2020 Information Processing Society of Japan

テゴリ（男，女），8 カテゴリ（10 代，20 代，...，80 代以上），47 カテゴリ（北海道，...，沖縄県）とし，属性値の総計 AT_A は 57 となる．小売店（組織 B）は 1 万人（N_B）の 1 万個の商品について，各個人の商品購入の有無のデータを持ち，1 商品につき 1 カテゴリとし，属性値の総計 AT_B は 10,000 となる．

このように，顧客情報管理組織が小売店の持つ ID の管理組織である場合には，小売店の持つ ID は顧客情報管理組織が持つ ID の部分集合とする．そのため，匿名加工をともなう 2 パーティ秘匿クロス集計を実行する過程で小売店に対し，顧客情報管理組織と小売店の持つ ID の共通集合を開示することを本稿では許容する．たとえば，顧客情報管理組織としてはポイントカード会社や IT 基盤に基づくサービスを提供する会社などが考えられる．そして，小売店はそれらの組織が発行する ID と顧客の購買情報を紐づけて記録したうえで顧客情報管理組織が持つデータとクロス集計し，需要予測や顧客への商品推薦に活用するといったユースケースが考えられる．

3. 先行研究

3.1 概要

提案手法で用いる加法準同型暗号および完全準同型暗号と（3.1.1 項），匿名加工をともなう 2 パーティ秘匿クロス集計に関連する秘匿積集合および秘匿等結合（3.1.2 項）と匿名加工（3.1.3 項）について概説する．

3.1.1 加法準同型暗号および完全準同型暗号

暗号 E と平文 m_1，m_2，演算 $\circ$ に関して $E(m_1)$，$E(m_2)$ から $E(m_1 \circ m_2)$ を計算できるとき，E を準同型暗号と呼ぶ．このとき，$\circ$ が加算であれば加法準同型暗号，乗算であれば乗法準同型暗号，加算・乗算の両方に関して成り立つとき完全準同型暗号と呼ぶ．完全準同型暗号は 2009 年に Gentry ら [4] によって実現された．しかし，文献 [7] にあるように当初は乗算制限をなくすための処理である bootstrapping の計算には 30 分かかることが報告されており，実用性には乏しかった．その後，効率化が進められ，現在ではデータ活用への利用例 [8], [9] もある．また，計算コストの観点から現在研究されている多くの完全準同型暗号は LWE 仮定に基づく方式 [10] ではなく，RLWE 仮定に基づく方式 [11] である．

RLWE 仮定に基づく方式では，多項式を暗号化できる．そのため，複数の整数の平文を 1 つの暗号文に暗号化することができる．これはパッキングと呼ばれ，特に CRT パッキング [12] は中国人剰余定理を用いてパッキングを行う．CRT パッキングでは格納した複数の平文に関して完全準同型演算ができる．たとえば，平文 x_1, x_2, x_3 を 1 つの暗号文 $E(x_1, x_2, x_3)$ に暗号化する．同様に暗号化された $E(y_1, y_2, y_3)$ に関して，$E(x_1 + y_1, x_2 + y_2, x_3 + y_3)$ や $E(x_1 \cdot y_1, x_2 \cdot y_2, x_3 \cdot y_3)$ を計算することができる．また，CRT パッキングされた暗号文は平文の右シフトを実行できるという特徴を持つ．たとえば，$E(x_1, x_2, x_3)$ を $E(x_3, x_1, x_2)$ のように右シフトすることができる．このため，乗算や加算，右シフトによって実現可能な処理を暗号化したまま実行することができる．

また，その他の完全準同型暗号の周辺技術として，暗号文の法を変換する Modulus switching [13] もある．これは，演算によるノイズ増加の抑制や暗号文の送受信における通信量削減などのために用いられる．完全準同型暗号のなかでも，RLWE 仮定に基づく方式の実装例として SEAL [14] や HElib [15] があり，CRT パッキングや CRT パッキングした平文の右シフトなどが実装されている．

加法準同型暗号の代表例として Paillier 暗号 [16] があり，暗号文どうしを乗算することで平文どうしの和の暗号文を計算できる．

3.1.2 秘匿積集合（PSI）および秘匿等結合

PSI（Private Set Intersection）は複数の組織が互いのデータを秘匿しながらその積集合をいずれかの組織に開示する．多者が参加する PSI [17] や 2 者のみが参加する PSI [8] がある．

秘匿等結合は互いのデータを秘匿しながら共通の識別子で等結合する秘匿計算である．出力する等結合表を平文でいずれかの組織に開示する方式として，可換ハッシュ関数（冪剰余など）を用いる方式 [18] がある．また，出力する等結合表を暗号化したうえでいずれかの組織に開示する方式として，加法準同型暗号を用いる方式 [19] がある．

3.1.3 匿名加工

匿名性の指標として k–匿名性 [20] や l–多様性 [21]，差分プライバシ [22] などがある．匿名加工では，これらの指標を満たすように匿名加工を行う．ここでは数学的に安全性を定式化でき，背景知識に依存しない差分プライバシを取り上げる．

一般に，個人情報を含むデータの統計量を開示すると，統計量から元の個人情報が部分的に漏洩する．たとえば，年収の平均値が 1 億円であれば，個々の年収に大きな値が含まれることが漏洩する．差分プライバシでは，統計量を加工することで，統計量からの個人情報の漏洩を制限する．この制限の度合はパラメータ ϵ により指定され，匿名加工をともなう統計量算出機構 F が ϵ–差分プライバシを満たすとは，F が下記の式 (1) を満たすことである．

$$\Pr[F(D_1) \in A] \leq \exp(\epsilon) \Pr[F(D_2) \in A] \tag{1}$$

ここで，Pr は確率を表し，D_1，D_2 は個人情報 1 人分だけ異なる任意の 2 つのデータであり，A は F の値域の任意の部分集合である．ϵ はデータを扱う組織のセキュリティポリシーを表すので，公開されることが多い．式 (1) を満たす代表的な加工方法は，式 (2) を満たすラプラスノイズを統計量に重畳することである．

© 2020 Information Processing Society of Japan

$$\frac{1}{2\sigma}\exp\left(-\frac{|x|}{\sigma}\right) \qquad \text{where} \quad \sigma \geq \frac{\Delta S}{\epsilon} \tag{2}$$

ここで，ΔS は統計量の感度と呼ばれ，統計量の定義および，F の定義域（データの形式と値の範囲）のみに依存し，データの現在の値には依存しない．σ はラプラス分布の尺度パラメータである．ノイズは，式 (2) のような期待値 0，分散 $2\sigma^2$ の確率密度関数に沿ってランダムに生成され，統計量に重畳される．

以上のように，ラプラスノイズは ϵ，F の定義，データの形式と値の範囲のみに依存するので，これらを公開する場合には，データの現在の値を秘匿したまま平文計算によってラプラスノイズを計算できる．

3.2 代表的な関連研究

代表的な先行研究である千田らの先行研究 [3] と，提案方式の計算量および通信量削減に関連がある完全準同型暗号を用いた秘匿積集合について概説する．

3.2.1 冪剰余の可換性と加法準同型暗号を用いる秘匿クロス集計

千田らは，秘匿等結合してから秘匿クロス集計する方式 [19] を改良し，暗号化状態でクロス集計表を匿名加工し，匿名加工後のクロス集計表を開示する方式を提案した [3]．文献 [19] では，秘匿等結合に Agrawal ら [18] の秘匿等結合を改良したものを用いている．

Agrawal らの秘匿等結合は可換ハッシュ関数を用いて照合する．組織 A，B が可換ハッシュ関数 f_{ka}，f_{kb} と ID 値 id_A，id_B をそれぞれ持つとき，式 (3) が成り立つことを利用し，ID そのものを明かさずに，id_A と id_B が等しいかを判定する．

$$id_A = id_B \Leftrightarrow f_{kb}(f_{ka}(id_A)) = f_{ka}(f_{kb}(id_B)) \tag{3}$$

また，ID 値が id_A, id_B であるデータを $data(id_A), data(id_B)$ とすると，式 (4) のように，$f_{ka}(id_A)$, $f_{kb}(id_B)$ に $data(id_A)$, $data(id_B)$ を紐付けることで，組織 A，B の持つデータの等結合が実現できる．

$$\begin{aligned} &data(id_A) \sim f_{kb}(f_{ka}(id_A)) \\ &\quad = f_{ka}(f_{kb}(id_B)) \sim data(id_B) \end{aligned} \tag{4}$$

可換ハッシュ関数としては冪剰余 $f_{ka}(x) = x^{ka} \mod p$ が代表的である．

文献 [19] では，紐づけるタプルデータを準同型暗号化し，タプルデータが暗号化された等結合表を秘匿計算する．そして，文献 [3] では暗号化された等結合表に準同型性を用いて匿名加工を加え，いずれかの組織に開示する．

3.2.2 完全準同型暗号を用いた秘匿積集合

Chen ら [8] は完全準同型暗号を用いた 2 パーティ PSI における効率化手法を提案している．この効率化手法は，秘匿クロス集計への応用が期待できる（4 章で詳述）．

アルゴリズム 1 文献 [8] の方式の雛形

Input: A の ID 集合 ID_A，B の ID 集合 ID_B
Output: ID_A と ID_B の積集合
1: B は ID_B の各要素を完全準同型暗号 FHE を用いて暗号化する．これを $c_1, \ldots, c_{|ID_B|}$ とし，A に送信する
2: A は $d_i \leftarrow r_i \prod_{id_A \in ID_A}(c_i - id_A)$ を暗号化状態で演算する
3: /* r_i は乱数である*/
4: A は $d_1, \ldots, d_{|ID_B|}$ を B に送信する
5: B は $ID_A \cap ID_B \leftarrow \{i : Dec_{FHE}(d_i) = 0\}$ を得る
6: /*Dec_{FHE} は FHE の復号関数とする*/

まず，文献 [8] の方式の雛形をアルゴリズム 1 に示す．アルゴリズム 1 のステップ 2 では，$d_i \leftarrow r_i \prod_{id_A \in ID_A}(c_i - id_A)$ を計算することで組織 A，B の ID の照合を行う．このとき，暗号文と平文，暗号文と暗号文どうしの演算は完全準同型暗号の準同型性を利用して演算する．文献 [8] では，アルゴリズム 1 に Cuckoo hashing [23]，Permutation-based hashing [24]，CRT パッキング [12]，Modulus switching [13] などを適用することで，計算量や通信量を削減する．

Cuckoo hashing は，複数のハッシュ関数を用いるハッシュテーブル作成方式である．ハッシュテーブル T，H 個のハッシュ関数 $h_1, \ldots, h_H$，格納したい値 x，乱数 $i \in \{1, \ldots, H\}$ のとき，$T[h_i(x)]$ に (x, i) を格納する．$T[h_i(x)]$ の格納前の値 w が空でなかった場合，w に関して再帰的に格納を行う．このため，Cuckoo hashing はある値の格納場所が H カ所に固定されるという特徴を持つ．これにより，ID 集合をハッシュテーブルに格納したうえで照合することで，アルゴリズム 1 のステップ 2 における積の数を N_A（ID_A の要素数）からハッシュ関数の値域の要素数に削減できる．詳細は 4.3 節で述べる．

Permutation-based hashing は，他のハッシュテーブル作成方式と組み合わせて，格納する要素の大きさを削減する．Cuckoo hashing と組み合わせる場合，格納する値 x を上位ビット x_L とそれ以外のビット x_R に分解し，$T[h_i(x_L) \oplus x_R]$ に (x_L, i) を格納する（$\oplus$ は排他的論理和）．これにより，ハッシュテーブルに格納する要素の大きさを $\log(x) + \log(H)$ ビットから $\log(x_L) + \log(H)$ ビットに削減できる．

CRT パッキングは，ステップ 2 における総積演算の並列実行による計算時間短縮やステップ 1，ステップ 4 における通信量削減が期待できる．また，Modulus switching はステップ 4 における通信量削減のために用いられる．

Cuckoo hashing のハッシュテーブル作成失敗や Permutation-based hashing による照合失敗（異なる ID どうしを等しいと判定する）が起こりうる．しかし，文献 [8] にあるように，格納したいデータ数に対するハッシュテーブルの大きさを調整することで，これらの確率を十分小さくすることができる．詳細については文献 [8] を参照されたい．

© 2020 Information Processing Society of Japan

4. 提案方式

完全準同型暗号を用いた匿名加工をともなう 2 パーティの秘匿クロス集計を行う方式を提案する．変数や記法を定義（4.1 節）したのち，提案方式の雛形を示し（4.2 節），Chen らの PSI プロトコルの効率化手法の応用（4.3 節）と破綻確率を考慮した加法準同型暗号へのパッキング（4.4 節）により，提案方式の雛形から計算量および通信量を削減する．

4.1 定義

変数や記法を以下のように定義する．

- 2 つの組織 A，B
- 組織 A，B のデータはタプル（テーブルの行）の集合とする．タプルは ID 部とデータ部から成る．タプルのデータ部をタプルデータと略す．2 つ以上のタプルが同じ ID を持つことはない．
- A の ID 集合 ID_A，B の ID 集合 ID_B
- ID_A の要素数 N_A，ID_B の要素数 N_B．N_A は組織 A のタプル数，N_B は組織 B のタプル数に等しい．
- $id_A \in ID_A$ について組織 A が持つタプルデータ $tup_A(id_A)$，組織 B についても同様．タプルデータは二値化されているものとする．例を**図 2** に示す．図 2 のように都道府県が北海道であれば，二値化したときには北海道のセルに 1 が立つ．
- 組織 A の持つタプルデータの属性数 AT_A，組織 B の持つタプルデータの属性数 AT_B．図 2 の下段の列数は，二値化後の AT_A の一例である．
- 加法準同型暗号の暗号化関数 AHE
- 加法準同型暗号の復号関数 Dec_{AHE}
- 加法準同型暗号の平文の法 p
- 完全準同型暗号の暗号化関数 FHE
- 完全準同型暗号の復号関数 Dec_{FHE}
- 完全準同型暗号の平文の法 p'
- FHE の 1 つの暗号文に格納できる平文の数 $slot$
- ハッシュ関数の個数 H
- ID_A，ID_B を格納するハッシュテーブル T_A，T_B
- ハッシュテーブルの大きさ（ハッシュ関数の値域の要素数）m
- 冪剰余の法 p''
- クロス集計表 CT

4.2 提案方式の雛形

単純に完全準同型暗号を用いて ID の秘匿，照合を行い，加法準同型暗号を用いてタプルデータの秘匿，集計を行う提案方式の雛形をアルゴリズム 2 に示す．アルゴリズム 2 は，ID 集合 ID_A，ID_B を入力として，匿名加工後のクロス集計表を出力する．

ステップ 3 で i と id_B の対応関係を記憶しておくことで，ステップ 16 で i から id_B を求められる．ステップ 7 では，完全準同型性を利用して等号判定を行う．ここで，$id_A = c_i$ は，平文である id_A と完全準同型暗号文である c_i（すなわち $FHE(id_B)$）を引数とし，id_A と id_B の等号判定結果（1 または 0）を返す演算を表す．演算は完全準同型暗号化された状態で行い，等号判定結果の完全準同型暗号文を返す．この処理には，フェルマーの小定理を応用する．すなわち，等号判定したい 2 つの値（id_A と id_B）の差を $\phi(p')$ 乗し，1 から引く処理を完全準同型性の利用により暗号化状態で行う．ここで，p' は完全準同型暗号の平文の法である．なお，id_A と c_i の 1 つのペアに対して，ステップ 7 を AT_A 回実行するが，$id_A = c_i$ の等号判定は 1 回だけ実行し，2 回目以降は結果を再利用する．

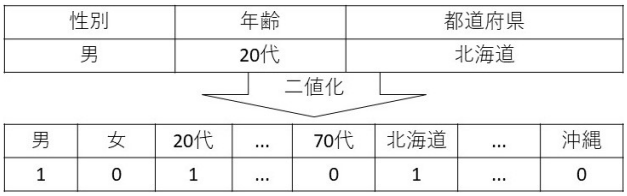


図 2 二値化の例
Fig. 2 An example of binarization.

また，ステップ 7 では $id_A = c_i$ が成り立つ $id_A \in ID_A$ に関して，id_A のタプルデータの加法準同型暗号文を完全準同型暗号により再暗号化して $d_{i,j}$ に格納する．つまり，$d_{i,j}$ に $FHE(AHE(tup_A(id_A)[j]))$ を格納する．再暗号化に際して，加法準同型暗号の暗号文の法は完全準同型暗号の平文の法よりも大きいため，加法準同型暗号の暗号文を $\log p'$ ビットごとに分割し，それぞれを完全準同型暗号の平文として CRT パッキングし，等号判定結果と乗算することで格納する．具体的には，$a = AHE(tup_A(id_A)[j])$ とすると，等号判定結果（$id_A = c_i$）に $[a_1, a_2, \ldots, a_{\log p/\log p'}, 0, \ldots, 0]$ を完全準同型性を利用して定数乗算する．ここで，a_k は a を $\log p'$ ビットごとに分割したときの k 番目の断片である．

ステップ 22 では，加法準同型暗号で暗号化された CT の値に乱数を加算する．その際，1 つ 1 つの加法準同型暗号文について，$[0, p-1]$ の一様乱数を加算し，剰余（$\bmod p$）を適用する．そのため，ステップ 24 において，組織 A が，加法準同型暗号の秘密鍵を用いて CT を復号しても，乱数により CT の値を知ることはできない．

アルゴリズム 2 はステップ 5〜9 において $N_B N_A$ 回の完全準同型暗号文どうしの等号判定の計算処理が必要であり，ステップ 10 において $N_B AT_A$ 個の完全準同型暗号文の通信処理が必要であるため，計算量および通信量が大きい．そこで，Chen らの PSI プロトコル [8] の効率化手法を応用（4.3 節で詳述）したうえで，破綻確率を考慮した加法準同型暗号へのパッキング（4.4 節で詳述）を行い，計算量および通信量を削減する．

© 2020 Information Processing Society of Japan

アルゴリズム 2 提案方式の雛形となる組織間クロス集計

Input: ID_A, ID_B
Output: 匿名加工後のクロス集計表 CT

1: 組織 B：
2: FHE の鍵を生成し，公開鍵を A に送信
3: すべての $id_B \in ID_B$ に関して $c_i \leftarrow FHE(id_B)$ を A に送信する，i と id_B の対応関係を記憶しておく
4: 組織 A：
5: **for all** $i \in \{1, \ldots, N_B\}$ **do in parallel**
6: **for all** $j \in \{1, \ldots, AT_A\}$ **do**
7: $d_{i,j} \leftarrow \sum_{id_A \in ID_A} (id_A = c_i) \times AHE(tup_A(id_A)[j])$
8: **end for**
9: **end for**
10: すべての $d_{i,j}$ を B に送信
11: 組織 B：
12: $AT_B \times AT_A$ の 2 次元配列 CT を用意し，$AHE(0)$ で初期化
13: **for all** $k \in \{1, \ldots, AT_B\}$ **do in parallel**
14: **for all** $j \in \{1, \ldots, AT_A\}$ **do**
15: **for all** $i \in \{1, \ldots, N_B\}$ **do**
16: **if** $tup_B(id_B)[k] = 1$ **then**
17: $CT_{k,j} \leftarrow CT_{k,j} + Dec_{FHE}(d_{i,j})$
18: **end if**
19: **end for**
20: **end for**
21: **end for**
22: CT のすべての要素に加法準同型性を利用して，それぞれ異なる $\lceil \log(p) \rceil$ ビットの乱数を加算して，A に送信
23: 組織 A：
24: CT を復号し，ラプラスノイズを重畳して B へ送信
25: 組織 B：
26: 受信したクロス集計表からステップ 22 で加算した乱数を減算し，匿名加工後のクロス集計表 CT を得る

4.3 Chen らの PSI プロトコルの効率化手法の応用

計算量を削減するために，ステップ 7 の総和を除去する．そのために Cuckoo hashing と Permutation-based hashing を応用する．組織 A, B が持つ ID_A, ID_B をそれぞれハッシュテーブル T_A，T_B に格納することを考える．このときの手順をアルゴリズム 3 に示す．

アルゴリズム 3 では，$N_A \gg N_B$ を想定して，ID_A を格納するハッシュテーブルは大きさ $m \times slot$ の 2 次元配列 T_A とし，ID_B を格納するハッシュテーブルは大きさ m の 1 次元配列 T_B とする．このとき，ある $id_A \in ID_A$ は T_A 内の H カ所のうち H カ所すべてに格納される．一方，ある $id_B \in ID_B$ は T_B 内の H カ所のうち 1 カ所に格納される．つまり，ある $id \in ID_A \cap ID_B$ が $id = T_B[i]$ であるとき，常に $id \in T_A[i]$ である．

ここで，このハッシュテーブルをアルゴリズム 2 に応用することを考えたとき，ある $id = T_B[i]$ を ID_A のすべての要素と照合する必要はなく，$T_A[i]$ の各要素とのみ照合を行えばよい．加えて，$T_B[i]$ と $T_A[i]$ の照合を CRT パッキングによる並列処理する．これにより，$slot \gg N_A/m$ が成り立つように $slot$ と m を設定することで，アルゴリズム 2 のステップ 7 の総和を除去することができる．

次にアルゴリズム 2 をハッシュテーブル，CRT パッキング，Modulus switching を用いて高速化したアルゴリズムをアルゴリズム 4 に示す．アルゴリズム 4 のステップ 7 では，通信量削減のため，CRT パッキングを用いて $slot$ 個の T_B の要素を 1 つの完全準同型暗号の暗号文に格納し，送信する．つまり，$FHE([T_B[1], \ldots, T_B[slot]])$ として送信する．ステップ 10 では，$FHE([T_B[1], \ldots, T_B[slot]]) \times [0, \ldots, 1, \ldots, 0]$ として，i 番目の要素 $T_B[i]$ を取り出す．

ステップ 11 では，等号判定の並列処理のために，Halevi と Shoup [25] の totalSums を用いて，$FHE([0, \ldots, T_B[i], \ldots, 0])$ を $FHE([T_B[i], \ldots, T_B[i]])$ にする．ステップ 14 では，4.2 節で示したフェルマーの小定理を応用した等号判定を並列処理する．具体的には $1 - \{FHE([T_B[i], \ldots, T_B[i]]) - [T_A[i][0], \ldots, T_A[i][slot]]\}^{p'}$ を計算する．これにより，$T_A[i][k] = T_B[i]$ が成り立つ k について，k 番目の要素にのみ 1 が立つ $FHE([0, \ldots, 1, \ldots, 0])$ が等号判定結果となる．つまり，ここでの演算 $=$ は平文の配列である $T_A[i]$ と完全準同型暗号文である d_i（すなわち $FHE([T_B[i], \ldots, T_B[i]])$）を引数とし，$T_A[i]$ のすべての要素と $T_B[i]$ の等号判定結果（$FHE([T_B[i] = T_A[i][0], \ldots, T_B[i] = T_A[i][slot]])$）を返す．

ステップ 16 では，等号判定結果をふまえ，$T_A[i][k] = T_B[i]$ となる ID のタプルデータの加法準同型暗号文のみを完全準同型暗号で再暗号化し，$d_{i,j}$ に格納する．このとき，組織 A は $T_B[i] = T_A[i][k]$ となる k や等号判定結果の 1 の立った場所を知ることはできない．このため，等号判定結果と $T_A[i]$ の各要素のタプルデータから $FHE([0, \ldots, a_1, a_2, \ldots, a_{\log p/\log p'}, \ldots, 0])$ を求める．ここで，$AHE(tup_A(T_A[i][k])[j]) = a_1 || \ldots || a_{\log p/\log p'}$ であり，$a_1, \ldots, a_{\log p/\log p'}$ は k 番目から $(k + \log p/\log p' \bmod slot)$ 番目に格納される．具体的には，すべての $x \in \{1, \ldots, \log p/\log p'\}$ について $A_x = [a_x^{(0+x) \bmod slot}, a_x^{(1+x) \bmod slot}, \ldots, a_x^{(slot-1+x) \bmod slot}]$ を計算する．ここで，a_x^y は $AHE(tup_A(T_A[i][y])[j])$ の x 番目の断片である．そして，$\sum_{x \in \{1, \ldots, \log p/\log p'\}} A_x(d_i \ll x)$ を計算することで，$FHE([0, \ldots, a_1, a_2, \ldots, a_{\log p/\log p'}, \ldots, 0])$ が求まる．$d_i \ll x$ は d_i の完全準同型暗号の平文の x 回の左シフトを意味する．ステップ 17 では，$d_{i,j}$ を Modulus switching を用いて暗号文の法の大きさを削減したうえで送信する．

4.4 破綻確率を考慮した加法準同型暗号へのパッキング

加法準同型暗号の秘匿計算の計算量や通信量の効率のため，複数の要素を 1 つの加法準同型暗号文に格納し，並列処理できるようにする加法準同型暗号へのパッキングの方法を示す．また，ラプラスノイズにより破綻する（正しい計算結果を復号できない）確率を示し，破綻確率を無視できるような各要素のビット長の決め方を示す．

© 2020 Information Processing Society of Japan

アルゴリズム 3 Cuckoo+Permutaion-based hashing

Input: A の ID 集合 ID_A，B の ID 集合 ID_B
Output: ID_A が格納されたハッシュテーブル T_A，ID_B が格納されたハッシュテーブル T_B

1: ハッシュテーブル T_A は大きさ $m \times slot$ の 2 次元配列
2: ハッシュテーブル T_B は大きさ m の 1 次元配列
3: 組織 A：
4: **for all** $x \in ID_A$ **do**
5: $x_R \leftarrow x$ の下位 $\log(m)$ ビット
6: $x_L \leftarrow x$ の下位 $\log(m)$ ビット以外
7: 乱数 $i \in \{0, \ldots, H\}$ を用意する
8: $T_A[h_i(x_L) \oplus x_R][j]$ が空であるような j を選ぶ
9: $T_A[h_i(x_L) \oplus x_R][j] \leftarrow (x_L, i)$
10: **end for**
11: 組織 B：
12: **for all** $y \in ID_B$ **do**
13: $y_R \leftarrow y$ の下位 $\log(m)$ ビット
14: $y_L \leftarrow y$ の下位 $\log(m)$ ビット以外
15: 乱数 $i \in \{0, \ldots, H\}$ を用意する
16: $w \leftarrow T_B[h_i(y_L) \oplus y_R]$
17: $T_B[h_i(y_L) \oplus y_R] \leftarrow (y_L, i)$
18: **if** w が空でない **then**
19: $y \leftarrow w$ とし，ステップ 13 から再帰的に処理
20: **end if**
21: **end for**

加法準同型暗号へのタプルデータのパッキングでは，ある $id \in ID_A$ の大きさ AT_A のタプルデータ $tup_A(id)$ を 1 つの加法準同型暗号文に暗号化する．パッキングは格納したい要素 $t_1, \ldots, t_n$ について式 (5) の t を計算し，t を加法準同型暗号の平文として暗号化する（$||$ はビットによる連接を意味する）．

$$t = t_1||1||t_2||1||\cdots||1||t_{n-1}||1||t_n||1 \tag{5}$$

このとき，$t_1, \ldots, t_n$ はそれぞれ $l-1$ ビットであるとする．

加法準同型演算による加算の結果が 2^{l-1} 以上の場合や -2^{l-1} 以下の場合には，t_i における繰り上がりや繰り下がりが t_{i+1} や t_{i-1} に伝搬し，正しい計算結果を復号することができない．そのため，l は計算結果の最大値，最小値よりも大きなビット数にする必要がある．しかし，ラプラスノイズを加算する場合には最大値，最小値が存在しないので，ラプラスノイズを加算した結果の絶対値が 2^{l-1} 以上となる確率を無視できる程度小さくするような l を設定する．

ラプラス分布 $\mathcal{L}(0, \sigma)$ に従う乱数の絶対値が $s\sigma$ 以上となる確率は $\Pr[|x| \geq s\sigma] = \exp(-s)$ である．そこで，ラプラスノイズを n 回生成したときに，すべてのノイズの絶対値が $s\sigma$ 未満となる確率 $Prob$ は以下の式 (6) で与えられる．

$$Prob = (1 - \exp(-s))^n \tag{6}$$

クロス集計表のセルの数を n，ノイズ重畳前のセルの最大値を max とする．式 (6) を満たす s を求め，$\log(s\sigma + max) \leq l - 1$ を満たすように l を設定すれば，すべてのセルの値の絶対値が 2^{l-1} 未満となる確率（すなわち破綻しない確率）を $Prob$ 以上にすることができる．よって，目標とする破綻確率を Pr_{fail} とすると，$Prob = 1 - Pr_{fail}$ として，上記を満たす最小の正整数として l を求めればよい．なお，$n = AT_A AT_B$，$max = N_B$ である．

アルゴリズム 4 アルゴリズム 2+CRT パッキング+Modulus switching

Input: ID_A, ID_B
Output: 匿名加工後のクロス集計表 CT

1: 組織 B：
2: FHE の鍵を生成し，公開鍵を A に送信
3: 組織 A：
4: AHE の鍵を生成し，公開鍵を B に送信
5: アルゴリズム 3 を用いて，ID_A，ID_B を T_A，T_B に格納
6: 組織 B：
7: $FHE(T_B)$ を A へ送信
8: 組織 A：
9: **for all** $i \in \{1, \ldots, m\}$ **do in parallel**
10: $d_i \leftarrow FHE([0, \ldots, T_B[i], \ldots, 0])$
11: $d_i \leftarrow totalSums(d_i)$
12: **end for**
13: **for all** $i \in \{1, \ldots, m\}$ **do in parallel**
14: $d_i \leftarrow (d_i = T_A[i])$ //$T_B[i] = T_A[i][k]$ とする
15: **for all** $j \in \{1, \ldots, AT_A\}$ **do**
16: $d_{i,j} \leftarrow AHE(tup_A(T_A[i][k])[j]) \times d_i$
17: $d_{i,j}$ を B に送信
18: **end for**
19: **end for**
20: 組織 B：
21: 大きさ $AT_B \times AT_A$ の 2 次元配列 CT を $AHE(0)$ で初期化
22: **for all** $k \in \{1, \ldots, AT_B\}$ **do in parallel**
23: **for all** $i \in \{1, \ldots, m\}$ **do**
24: **for all** $j \in \{1, \ldots, AT_A\}$ **do**
25: $c_{i,j} \leftarrow Dec_{FHE}(d_{i,j})$
26: **if** $tup_B(id_B)[k] = 1$ **then**
27: $CT[k][j] \leftarrow CT[k][j] + c_{i,j}$
28: **end if**
29: **end for**
30: **end for**
31: **end for**
32: CT のすべての要素に加法準同型性を利用して，それぞれ異なる $\lceil \log(p) \rceil$ ビットの乱数を加算して，A に送信
33: 組織 A：
34: CT を復号し，ラプラスノイズを重畳して B へ送信
35: 組織 B：
36: 受信したクロス集計表からステップ 32 で加算した乱数を減算し，匿名加工後のクロス集計表 CT を得る

4.5 提案方式のアルゴリズム

4.4 節の手法を用いてアルゴリズム 4 を改良した提案方式をアルゴリズム 5 に示す．破綻確率を考慮した加法準同型暗号へのパッキングにより，アルゴリズム 5 ではアルゴリズム 4 に比べ，ステップ 15〜17 のループ回数が AT_A から $\lceil lAT_A / \log p \rceil$ となる．

また，アルゴリズム 5 では匿名加工後のクロス集計表を組織 B にのみ開示するが，アルゴリズム 5 について以下

© 2020 Information Processing Society of Japan

アルゴリズム 5 FHE-PSI

Input: ID_A, ID_B
Output: 匿名加工後のクロス集計表 CT

1: 組織 B：
2: FHE の鍵を生成し，公開鍵を A に送信
3: 組織 A：
4: AHE の鍵を生成し，公開鍵を B に送信
5: アルゴリズム 3 を用いて，ID_A，ID_B を T_A，T_B に格納
6: 組織 B：
7: $FHE(T_B)$ を A へ送信
8: 組織 A：
9: **for all** $i \in \{1, \ldots, m\}$ **do in parallel**
10: $d_i \leftarrow FHE([0, \ldots, T_B[i], \ldots, 0])$
11: $d_i \leftarrow totalSums(d_i)$
12: **end for**
13: **for all** $i \in \{1, \ldots, m\}$ **do in parallel**
14: $d_i \leftarrow (d_i = T_A[i])$
15: **for all** $j \in \{1, \ldots, \lceil lAT_A / \log p \rceil\}$ **do**
16: $d_{i,j} \leftarrow AHE(tup_A(T_A[i][k]))[j] \times d_i$
17: **end for**
18: $d_{i,j}$ を B に送信
19: **end for**
20: 組織 B：
21: 大きさ $AT_B \times \lceil lAT_A / \log p \rceil$ の 2 次元配列 CT を $AHE(0)$ で初期化
22: **for all** $k \in \{1, \ldots, AT_B\}$ **do in parallel**
23: **for all** $i \in \{1, \ldots, m\}$ **do**
24: **for all** $j \in \{1, \ldots, \lceil lAT_A / \log p \rceil\}$ **do**
25: $c_{i,j} \leftarrow Dec_{FHE}(d_{i,j})$
26: **if** $tup_B(id_B)[k] = 1$ **then**
27: $CT[k][j] \leftarrow CT[k][j] + c_{i,j}$
28: **end if**
29: **end for**
30: **end for**
31: **end for**
32: CT のすべての要素に加法準同型性を利用して，それぞれ異なる $\lceil \log(p) \rceil$ ビットの乱数を加算して，A に送信
33: 組織 A：
34: CT を復号し，ラプラスノイズを重畳して B へ送信
35: 組織 B：
36: 受信したクロス集計表からステップ 29 で加算した乱数を減算し，匿名加工後のクロス集計表 CT を得る

の変形を行うことで組織 A にのみ開示するように変更することができる．アルゴリズム 5 のステップ 32 の $\lceil \log(p) \rceil$ ビットの乱数を加算している部分について，仮にここで $\lceil \log(p) \rceil$ ビットの乱数ではなく，組織 B が生成したラプラスノイズを重畳し，それを組織 A に送信し，組織 A が送られた暗号文を復号するという手順を行うと組織 A に匿名加工後のクロス集計表を開示することができる．このとき，ステップ 33 以降は実行しない．

5. 比較

提案方式と比較する方式（5.1 節）を示したうえで，2 つの方式の計算量，通信量を理論比較（5.2 節），実装比較（5.3 節）する．ここで，計算量については冪剰余や FHE，AHE の暗号化，FHE の準同型演算を計数する．これらは平文の四則演算や AHE の準同型演算に比べ，計算量が大きいためである．通信量については冪剰余の結果や FHE，AHE の暗号文の送信を計数する．これらは小さな整数や制御情報の送受信に比べ，通信量が大きいためである．

5.1 比較対象方式

比較対象方式には千田らの匿名加工をともなう 2 パーティ秘匿クロス集計 [3] を用いる．提案方式との公平な比較のため，以下を行う．(1) 破綻確率を考慮した加法準同型暗号へのパッキングを行う（4.4 節）．(2) 匿名加工は差分プライバシ基準を満たすようにラプラスノイズを重畳する．(3) 組織 B への共通集合の開示を許容する．

(1) については，破綻確率の考慮は差分プライバシを用いるうえで必須であり，提案方式でも取り入れている．このため，比較対象方式でも同様に取り入れる．また，パッキングは提案方式で取り入れた効率化手法であるが，公平な比較のため，比較対象方式でも同様のパッキングを取り入れる．(2) については，差分プライバシを用いるためには，必須である．(3) については，比較対象方式のセキュリティ条件を提案方式と同じレベルに緩和する．この緩和を受けて，比較対象方式を効率化する．

(1)〜(3) を適用した文献 [3] のアルゴリズムをアルゴリズム 6 に示す．f_{key} は鍵 key と事前に共有された法 p'' を用いて，$f_{key}(x) = x^{key} \mod p''$ と定義する．

(1) のパッキングにより，ステップ 15〜19 のループの回数が AT_A から $\lceil lAT_A / \log p \rceil$ に削減されている．これは提案方式のステップ 15〜17 の削減効果と同じである．(2) のため，ステップ 24 では，ラプラスノイズを重畳する．

(3) のため，ステップ 5 では文献 [3] において $(f_{kb}(h(id_B)), AHE(tup(id_B)))$ を送信する部分を比較対象方式では $(f_{kb}(h(id_B)), i)$ としている．ステップ 5 で i を送信することで，組織 B がステップ 11 で i から id_B を導出できるようにしている．具体的には，ステップ 11 において $f_{ka}(f_{kb}(h(id_B)))$ には i が紐づいている．ステップ 5 における i と id_B の対応関係を記憶しておけば，ステップ 11 において，i から id_B を導出できる．

$$f_{kb}(f_{ka}(h(id_A))) = f_{ka}(f_{kb}(h(id_B))) \tag{7}$$

式 (7) が成り立つ $f_{ka}(f_{kb}(h(id_B)))$ に紐づけられた i から，組織 B は id_B を知ることができる．また，$f_{kb}(f_{ka}(h(id_A)))$ にはステップ 7 において，$AHE(tup_A(id_A))$ が紐づく．ここで，$id_B \in ID_A \cap ID_B$ であり，組織 B は id_B および id_B に対応する $AHE(tup_A(id_A))$ を得られるため，クロス集計することができる．また，組織 B は $id_B \in ID_A \cap ID_B$ を得るため，組織 B に共通集合（$ID_A \cap ID_B$）が漏洩する．

(3) により，セキュリティ条件を緩めたことで，比較対象方式は文献 [3] に比べ，効率化している．ステップ 5 につ

© 2020 Information Processing Society of Japan

いて，文献 [3] では $(f_{kb}(h(id_B)), AHE(tup(id_B)))$ を送信するところで，比較対象方式では $(f_{kb}(h(id_B)), i)$ を送信する．ステップ 5 で $(f_{kb}(h(id_B)), AHE(tup(id_B)))$ を送信する場合，$\mathcal{O}(N_B)$ 個の冪剰余の値と $\mathcal{O}(N_B AT_B)$ 個の加法準同型暗号文を送信することになる．一方，$(f_{kb}(h(id_B)), i)$ を送信する場合，$\mathcal{O}(N_B)$ 個の冪剰余の値と $\mathcal{O}(N_B)$ 個の重複しない適当な乱数を送信することになる．つまり，比較対象方式では文献 [3] に比べ，$\mathcal{O}(N_B AT_B)$ 個の加法準同型暗号文の送信を取り除いている．また，$\mathcal{O}(N_B AT_B)$ 個の加法準同型暗号文の送信を行わないため，$\mathcal{O}(N_B AT_B)$ 個の加法準同型暗号の計算も行わない．

比較対象方式 [3] において，組織 A および B の保有する ID 数（N_A，N_B），タプルデータの属性数（AT_A，AT_B），属性の値域（2 値化されているので 0 または 1）は秘匿の対象ではない．組織 A および B のタプルデータの属性値を相手組織（B および A）に対して秘匿する点の安全性は，加法準同型暗号の安全性に帰着する．組織 A および B の ID 集合を相手組織に秘匿する点の安全性は，冪剰余の一方向性に帰着する．これらの点は，提案方式との比較のために修正された比較対象方式（アルゴリズム 6）でも同様である．ただし，組織 A のタプルデータの属性値を組織 B に秘匿する点の安全性のみ異なり，これは，加法準同型暗号および差分プライバシの安全性に帰着する．

比較対象方式 [3] において，組織 A と組織 B の共通 ID 集合（$ID_A \cap ID_B$）は組織 A と B の両者に秘匿される．この秘匿の安全性は，冪剰余と加法準同型暗号の安全性に帰着する．しかし，提案方式との比較のために修正された比較対象方式（アルゴリズム 6）では，上述のように，共通 ID 集合の組織 B への漏洩を許容する．

5.2 理論比較

5.2.1 パラメータの扱い

2 章で取り上げた山口らの組織間プライバシ保護推薦システム [1] を参考に，N_A，N_B，AT_B を変数，AT_A を定数と見なす．N_A，N_B については，組織ごとにユーザ数は変化することが想定されるため，変数とする．AT_B は商品数であり，小売店によって大幅に変化するため，変数とする．一方，AT_A については，組織 A が持つ属性情報は個人情報であり，会員から入手できる個人情報には限りがある．したがって，今回のユースケースにおける想定（表 1）に比べて大幅に増加することは考えにくいため，AT_A は定数とする．実際，文献 [1] でも，N_A は 10^7〜10^8，N_B は 10^2〜10^5，AT_B は 10^1〜10^4 とする一方，AT_A は 57 で固定されている．

p，p'，p''，H，l を定数と見なす．p，p'，p'' はそれぞれ加法準同型暗号，完全準同型暗号，冪剰余のパラメータであり，セキュリティ条件に従い定まる．H はハッシュ関数の個数であり，文献 [8] を参考に固定の値とする．

アルゴリズム 6 EXP-HE

Input: ID_A, ID_B
Output: 匿名加工されたクロス集計表

組織 A：
　鍵 ka と AHE の鍵 (pk, sk) を生成し，pk を B へ送信
　すべての $id_A \in ID_A$ に関して $(f_{ka}(h(id_A)), AHE(tup(id_A)[j]))$ を B に送信 $(j \in \{1, \ldots, \lceil lAT_A / \log(p) \rceil\})$
組織 B：
　鍵 kb を生成し，すべての $id_B \in ID_B$ に関して $(f_{kb}(h(id_B)), i)$ を A に送信
ここで，i は重複しない適当な乱数
　$(f_{kb}(f_{ka}(h(id_A))), AHE(tup(id_A))[j])$ を計算
組織 A：
　$(f_{ka}(f_{kb}(h(id_B))), i)$ を B に送信
組織 B：
　$f_{kb}(f_{ka}(h(id_A))) = f_{ka}(f_{kb}(h(id_B)))$ となる $tup_B(id_B)$ と $AHE(tup_A(id_A)[j])$ を見つけ，$c_{i,j} \leftarrow AHE(tup_A(id_A)[j])$ に格納
　大きさ $AT_B \times \lceil lAT_A / \log p \rceil$ の 2 次元配列 CT を $AHE(0)$ で初期化
　for all $k \in \{1, \ldots, AT_B\}$ **do in parallel**
　　for all $i \in \{1, \ldots, N_B\}$ **do**
　　　for all $j \in \{1, \ldots, \lceil lAT_A / \log(p) \rceil\}$ **do**
　　　　if $tup_B(id_B)[k] = 1$ **then**
　　　　　$CT[k][j] \leftarrow CT[k][j] + c_{i,j}$
　　　　end if
　　　end for
　　end for
　end for
　CT のすべての要素に準同型性を利用してそれぞれ異なる $\lceil \log(p) \rceil$ ビットの乱数を加算して，A に送信
組織 A：
　CT のすべての要素を復号し，ラプラスノイズを重畳して B へ送信
組織 B：
　受信したクロス集計表からステップ 21 で加算した乱数を減算し，匿名加工されたクロス集計表を得る

l は加法準同型暗号文へパッキングされた要素のビット長であり，$l = 32$ など計算機で扱いやすい値に固定できる．l はクロス集計表のセルの数（n）およびセルの最大値（max），ラプラス分布の尺度パラメータ（σ），破綻確率（Pr_{fail}）に応じて定まる．ここで，文献 [1] のユースケースにおけるクロス集計表のセルの数やセルの最大値のさらに 100 倍である $n = 57 \times 10^6$ および $max = 10^7$ を想定する（このとき統計量の感度 ΔS も 100 倍となり，$\Delta S = 6 \times 10^6$ となる．ΔS の決め方については 5.3.1 項にて詳述）．また，σ は式 (2) より ΔS と ϵ により定まるが，ϵ は多くの既存研究 [26], [27], [28] では 0.1〜10 の間で設定されている．ϵ が小さいほど l が大きくなるため，ここでは $\epsilon = 0.1$ に設定する．ゆえに，$\sigma = 6 \times 10^7$ とする．破綻確率 $Pr_{fail} = 10^{-9}$ とすると，式 (6) より $s \approx 38.58$ であり，$l \geq \log(s\sigma + max) + 1 \approx 31.11$ である．つまり，$l = 32$ と設定しておけば，十分である．このように，l は本稿で想

© 2020 Information Processing Society of Japan

定するユースケースにおいては，固定することができる．

また，$m, slot$ は変数であり，m は N_B の定数倍とし，$slot$ は $\mathcal{O}(N_A/N_B)$ とする．m は文献 [8] を参考に，十分大きな確率で N_B 個の要素をすべて格納できるハッシュテーブルの大きさとする．$slot$ は $slot \gg N_A/m$ が成り立つように設定するため，$\mathcal{O}(N_A/N_B)$ とする．

5.2.2 提案方式

計算量については，アルゴリズム 5 より，ステップ 7 において完全準同型暗号の暗号化 $\lceil m/slot \rceil$ 回，ステップ 10 において準同型演算の定数乗算を m 回，totalSums では 1 回につき準同型演算の右シフトを $\log(slot)$ 回行うのでステップ 11 において準同型演算の右シフト $m\log(slot)$ 回である．完全準同型暗号の等号判定は $\log(p')$ 回程度の乗算を行うので，ステップ 14 において準同型演算の乗算 $m\log(p')$ 回，ステップ 16 において準同型演算の定数乗算を $m\lceil lAT_A/\log(p) \rceil \log(p)/\log(p')$ 回，準同型演算の右シフトを $m\lceil lAT_A/\log(p) \rceil \log(p)/\log(p')$ 回，加法準同型暗号の暗号化 $N_A\lceil lAT_A/\log(p) \rceil$ 回，ステップ 25 において完全準同型暗号の復号 $m\lceil lAT_A/\log(p) \rceil$ 回，ステップ 34 において加法準同型暗号の復号 $AT_B\lceil lAT_A/\log(p) \rceil$ 回である．ゆえに，計算量のオーダは $\mathcal{O}(N_A + N_B + AT_B)$ である．

通信量については，アルゴリズム 5 より，ステップ 7 において完全準同型暗号の暗号文 $\lceil m/slot \rceil$ 個，ステップ 18 において完全準同型暗号の暗号文 $m\lceil lAT_A/\log(p) \rceil$ 個，ステップ 32 において加法準同型暗号の暗号文 $AT_B\lceil lAT_A/\log(p) \rceil$ 個である．ゆえに，通信量のオーダは $\mathcal{O}(N_B + AT_B)$ である．

5.2.3 比較対象方式

計算量についてはアルゴリズム 6 より，ステップ 3 において $\lceil \log(p'') \rceil$ ビットの冪剰余 N_A 回，加法準同型暗号の暗号化 $N_A\lceil lAT_A/\log(p) \rceil$ 回，ステップ 5 において冪剰余 N_B 回，ステップ 7 において冪剰余 N_A 回，ステップ 24 において加法準同型暗号の復号 $AT_B\lceil lAT_A/\log(p) \rceil$ 回である．ゆえに，計算量のオーダは $\mathcal{O}(N_A + N_B + AT_B)$ である．

通信量については，アルゴリズム 6 よりステップ 3 において $\lceil \log(p'') \rceil$ ビットの数 N_A 個，加法準同型暗号の暗号文 $N_A\lceil lAT_A/\log(p) \rceil$ 個，ステップ 5 において $\lceil \log(p'') \rceil$ ビットの数 N_B 個，ステップ 9 において $\lceil \log(p'') \rceil$ ビットの数 N_B 個，ステップ 22 において加法準同型暗号の暗号文 $AT_B\lceil lAT_A/\log(p) \rceil$ 個である．ゆえに，通信量のオーダは $\mathcal{O}(N_A + N_B + AT_A)$ である．

5.2.4 理論比較のまとめ

オーダによる比較をまとめて**表 2** に示す．提案方式と比較対象方式の計算量のオーダに差異はないが，通信量は異なる．提案方式（アルゴリズム 5）では，組織 A と B の等しい ID の組を見つけるのに，組織 B が各 $id_B \in ID_B$ を完全準同型暗号化して組織 A に送る（ステップ 7）．組織 A は完全準同型性を利用し，和と積の組合せ（フェルマーの小定理）を用いて，$id_A \in ID_A$ と $id_B \in ID_B$ の等号判定を秘匿状態で行う（ステップ 14）．ステップ 16 では，組織 A のタプルデータを加法準同型暗号化し，完全準同型性を利用して等号判定結果に乗じることで，id_B と等しい id_A についてのみタプルデータを選別する[*1]．ステップ 18 は，選別したタプルデータを $d_{i,j}$ として組織 B に送る．$d_{i,j}$ の個数は $\mathcal{O}(N_B AT_A)$ となるが，5.2.1 項で前述したように，AT_A は定数と見なせるので $\mathcal{O}(N_B)$ となる．$d_{i,j}$ の個々のデータ量は暗号文のサイズであり，定数である[*2]．そこで，ステップ 7 において，$\mathcal{O}(N_B)$ 個の完全準同型暗号文，ステップ 18 においても $\mathcal{O}(N_B)$ 個の加法準同型暗号文[*3]を送信することになるので，通信量は $\mathcal{O}(N_B)$ を含み，$\mathcal{O}(N_A)$ は含まない（後段のステップ 32 により $\mathcal{O}(N_B + AT_B)$ となる）．

表 2 提案方式および比較対象方式の計算量，通信量のオーダ
Table 2 Computational and communication complexity order.

	計算量	通信量
提案方式	$\mathcal{O}(N_A + N_B + AT_B)$	$\mathcal{O}(N_B + AT_B)$
比較対象方式	$\mathcal{O}(N_A + N_B + AT_B)$	$\mathcal{O}(N_A + N_B + AT_B)$

比較対象方式（アルゴリズム 6）では，組織 A と B の等しい ID の組を見つけるのに，冪剰余の可換性を用いる．組織 A は，各 $id_A \in ID_A$ のハッシュ値を鍵 ka を用いて冪剰余した値 $f_{ka}(h(id_A))$ および，タプルデータ（$tup(id_A)$）の加法準同型暗号文 $AHE(tup(id_A))$ を組織 B に送信する（ステップ 3）．このとき，冪剰余の値およびタプルデータの加法準同型暗号文について，各々 $\mathcal{O}(N_A)$ 個のデータを通信する．組織 B は組織 A から受け取った $f_{ka}(h(id_A))$ を，自らの鍵 kb を用いてさらに冪剰余する（ステップ 7）．組織 B は各 $id_B \in ID_B$ のハッシュ値を鍵 kb を用いて冪剰余した値 $f_{kb}(h((id_B))$ と乱数 i を組織 A に送信する（ステップ 5）．このとき，N_B 個の冪剰余の値と乱数の通信が発生する．組織 A は組織 B から受け取った $f_{kb}(h(id_B))$ を自らの鍵 ka を用いてさらに冪剰余し，その値と乱数 i を組織 B に送り返す（ステップ 9）．このとき，N_B 個の冪剰余の値と乱数の通信が発生する．組織 B 側で $f_{kb}(f_{ka}(h(id_A)))$ と $f_{ka}(f_{kb}(h(id_b)))$ の一致判定により，$id_A = id_B$ となる ID の対を求め，それらの対について，クロス集計表 $(c_{i,j})$ に id_A のタプルデータを格納する（ステップ 11）．以上から，比較対象方式の通信量は $\mathcal{O}(N_A + N_B)$ を含む（後段のステップ 22 により $\mathcal{O}(N_A + N_B + AT_B)$ となる）．

以上をまとめると，提案方式では，完全準同型性を利用して id_A と id_B の等しい対を秘匿状態で求めるので，組織

[*1] Cuckoo hashing やパッキングなどの効率化のために分かりにくくなっているが，アルゴリズム 2 のステップ 3 および 7 の変形である．
[*2] タプルデータのサイズは，AT_A を通じて暗号文の個数に反映されるが，上述したように AT_A は定数と見なせる．
[*3] 厳密には加法準同型暗号文をさらに完全準同型暗号化した暗号文

© 2020 Information Processing Society of Japan

表 3 実装実験結果（単位：秒）
Table 3 Result of execution time in seconds.

すべての条件で 3 回実行し，その平均をとった

	LAN 環境			WAN 環境		
	計算時間	通信時間	総実行時間	計算時間	通信時間	総実行時間
提案方式	5,170	53	5,223	5,160	473	5,633
比較対象方式	5,989	75	6,064	5,909	1,338	7,247
提案/比較対象	86.3%	70.3%	86.1%	87.3%	35.3%	77.7%

B から組織 A に $\mathcal{O}(N_B)$ 個の完全準同型暗号化された id_B を送れば，組織 A 側で等号判定を行うことができる．また，完全準同型性を利用して，id_B と等しい id_A について $tup(id_A)$ を秘匿状態で選別するので，組織 A から $\mathcal{O}(N_B)$ 個の加法準同型暗号化された $tup(id_A)$ を送ればよい．そのため，通信量は $\mathcal{O}(N_B)$ となる．一方，比較対象方式では，$f_{kb}(f_{ka}(h(id_A)))$ と $f_{ka}(f_{kb}(h(id_b)))$ の比較により id_A と id_B の等号判定を秘匿状態で行う．f_{ka} は組織 A，f_{kb} は組織 B でしか計算できないので，id_A および id_B は組織 A と B の両方を通る必要がある．そのため，通信量は $\mathcal{O}(N_A + N_B)$ となる．

4.3 節で述べたように組織 A，B の役割は，匿名加工後のクロス集計表の開示先には依存しない．そのため，たとえば顧客情報管理組織と小売店が提案方式を実行する場合に計算量や通信量が小さくなるようにどちらが組織 A，B の役割をそれぞれ担うのかを設定すればよい．組織 B の役割をレコード数が小さい組織が担うことで，通信量はレコード数が小さい組織のレコード数にのみ依存する．よって，最も支配的な変数をオーダから除去した．このため，2 章の表 1 の顧客情報管理組織と小売店のように ID の数が大きく異なる場合でも，通信量が ID の数が小さい小売店の ID の数にのみ依存する．

5.3 実装比較

5.3.1 実装評価

提案方式は C++を用いて実装した．ソケット通信には Boost C++ libraries [29]，並列計算には intelTBB [30] を用いた．これらは広く用いられている．また，加法準同型暗号には暗号研究者である John Betherncourt が開発したライブラリ [31] を使用し，完全準同型暗号には文献 [8] と同様に SEAL 3.0 [14] を用いた．

比較対象方式は提案方式と同様に C++を用いて実装し，ソケット通信，並列計算，加法準同型暗号は提案方式と同一のライブラリを用いた．比較対象方式固有の処理である冪剰余については，広く用いられている GMP ライブラリ [32] を用いた．

比較対象方式の実装の妥当性について説明する．アルゴリズムについては，提案方式との公平な比較のため，5.1 節で述べた (1)〜(3) を適用した．5.1 節で述べたように提案方式における効率化手法のうち，比較対象方式に適用可能なパッキングは比較対象方式にも適用し，効率化した (1)．匿名加工は提案方式と同じく差分プライバシ基準に基づくラプラスノイズ重畳とした (2)．提案方式とセキュリティ条件を同じくし，セキュリティ条件を緩和したことを受けて，比較対象方式を効率化した (3)．

比較対象方式の実装については，比較対象方式の基となる文献 [3], [19] では，実装について報告されていないため，筆者らが独自に実装を行った．提案方式の実装との公平性については，同一の言語を使用し，冪剰余以外は提案方式と同じライブラリを用い，冪剰余には広く用いられているライブラリを用いた．

組織 A，B にはそれぞれ 1 台ずつ計算機を割り当てた．組織 A 側は CPU が Ryzen Threadripper 2950X，メモリ 128 GB のサーバであり，組織 B 側は CPU が Core i7-8550U，メモリ 16 GB のノート PC である．実験は LAN と WAN を想定した環境でそれぞれ行った．LAN 環境は帯域が 790 Mbps（ネットワーク性能測定ツール iperf による計測）である．WAN 環境は文献 [33] を参考に，tc コマンドを用いて帯域を 40 Mbps，通信遅延を 40 ms に設定した．

組織 A，B のデータのサイズや形式は 2 章で説明したとおり，文献 [1] を参考に $N_A = 10^7$，$N_B = 10^4$，$AT_A = 57$，$AT_B = 10^4$ とした．すべてのタプルデータは二値化されている．また，提案方式のその他の定数および変数については以下のように設定した．$m = 16{,}384$，$slot = 2{,}048$，$\log(p) = 1{,}024$，$p' = 65{,}537$，$\log(p'') = 1{,}024$，$l = 32$．これらのパラメータは組織 A，B の計算機のメモリサイズを加味し，プログラムが使用するメモリサイズが計算機のメモリサイズを超過しないよう設定した．また，ID_A，ID_B の要素のビット数は ID_A，ID_B のすべての ID を表現可能な $\lceil \log(N_A) \rceil + 1$ ビットであるとした．

l に関しては，0.1–差分プライバシを満たすラプラスノイズを重畳することを考え，設定した（$\epsilon = 0.1$）．本稿で想定するクロス集計表の形式では 1 ユーザの性別，年齢，都道府県に変化があったとき，それぞれ最大 2 万セルが最大 1 だけ変化する．よってクロス集計表の感度 ΔS を 6×10^4 に設定した．このとき提案方式の破綻確率を $1/10^6$ とし，4.4 節で述べたように l を算出すると，$\pm 27\sigma$ までのラプラスノイズが加わることを想定すればよいことが分かる．こ

© 2020 Information Processing Society of Japan

こで $\pm 27\sigma = \pm 1.62 \times 10^7 \approx \pm 2^{24}$ である．このため l は 25 bit 程度に設定すればよいが，2 のべき乗のビット数の方が扱いやすいため，今回は 32 に設定した．

比較対象方式と提案方式の実行結果を**表 3** に示す．実行結果はそれぞれの方式と環境の計算時間と通信時間，総実行時間である．LAN 環境における実験では提案方式は比較対象方式の 86.1%の総実行時間となった．WAN 環境では提案方式は比較対象方式の 77.7%の総実行時間であった．

5.3.2 実装評価の考察

理論比較の結果通り，実装比較でも通信がクリティカルとなる WAN 環境で提案方式は優位性が顕著であった．理論比較におけるオーダでは差異がなかった計算量に関しては，実装比較において，提案方式が比較対象方式に比べて小さかった．一般に完全準同型暗号の乗算などの処理は加法準同型暗号の暗号化処理や冪剰余の処理に比べ，計算量が大きい．しかし，提案方式における計算量削減により，顕在化しなかったとみられる．現在も完全準同型暗号の高速化 [34] は進められており，今後の進展により，さらに高速化されると考える．

また，計算量のほとんどは組織 A が担うものであり，組織 A がより高性能な計算機を用いることで計算量の増加による実行時間の短縮が期待できる．特に，並列化可能な処理が大半なため，複数台の計算機を用いた分散処理が効果を発揮することが期待できる．このような特徴はクラウド環境での実行に向いており，近年広がっているクラウド型のサービスとの親和性が高い．

6. 提案方式の安全性

6.1 安全性の要件

6.1.1 攻撃者モデル

提案方式の攻撃者は，プロトコルの参加者すなわち組織 A と B，それ以外の外部の攻撃者が考えられる．そのうち外部の攻撃者については，ファイアウォール，侵入検知，認証，アクセル制御などの従来技術によって対策することとする．そこで，以下では，組織 B に対する攻撃者として組織 A，組織 A に対する攻撃者として組織 B を想定する．

提案方式の元になった方式 [8] は，semi-honest モデルにおいて安全性を証明している．提案方式は安全性の面において方式 [8] を改良していない．また，比較対象方式 [3] も semi-honest モデルにおいて安全性を証明している．そこで，提案方式の攻撃者モデルは比較対象方式と同様に semi-honest モデルとする．

6.1.2 保護の対象となる情報

保護の種類は秘匿であり，semi-honest モデルにおいて，提案方式（アルゴリズム 5）が以下の 5 つの要件を満たすことを目標とする．

条件 1 組織 B のタプルデータ（$tup_B(id_B)$，$id_B \in ID_B$）の属性の値（1 または 0）に関する情報が組織 A に対して漏洩しない．

条件 2 組織 B の ID 集合（ID_B）の情報が組織 A に漏洩しない．

条件 3 組織 A と組織 B に共通する ID の集合（$ID_A \cap ID_B$）の情報が組織 A に漏洩しない．

条件 4 組織 A のタプルデータ（$tup_A(id_A)$，$id_A \in ID_A$）の属性の値に関する情報が，組織 B に対して，差分プライバシの定めるレベルを超えて漏洩しない．

条件 5 組織 A の ID 集合（ID_A）のうち共通集合（$ID_A \cap ID_B$）以外の ID の情報が，組織 B に対して，差分プライバシの定めるレベルを超えて漏洩しない．

条件 4 および条件 5 について補足する．差分プライバシは，3.1.3 項で述べたように，データに対する統計量にノイズを加えることで，統計量からの元のデータについての推定を制限する [22]．このとき，式 (1) のように，パラメータ ϵ を用いて，ϵ–差分プライバシとして安全性のレベルが指定される．式 (1) を満たすラプラスノイズを，式 (2) により求め，統計量すなわちクロス集計表の各セルの値に重畳する．提案方式では，4.4 節において，そのようなノイズを重畳している．

なお，ID の数（N_A，N_B），タプルデータの属性数（AT_A，AT_B），属性の値域（0 または 1）[*4] は秘匿の対象でなく，相手組織に漏洩してもいい．この点は，比較対象方式と同様である．

1 章および 2 章で述べたように，提案方式は，組織 B の ID 集合（ID_B）が組織 A の ID 集合（ID_A）の部分集合であることを前提としている．そのため，$ID_B = ID_A \cap ID_B$ であり，組織 A と B の ID の共通集合（$ID_A \cap ID_B$）は組織 B に漏洩する．比較対象方式の文献 [3] では，$ID_A \cap ID_B$ はいずれの組織にも漏洩しないので，この点は先行研究よりも条件を緩めている．そのため，5.1 節で修正した比較対象方式（アルゴリズム 6）では，公平なセキュリティ条件とするため，$ID_A \cap ID_B$ は組織 B に漏洩してもよいとした．

6.2 安全性の証明

条件 1 組織 B のタプルデータ（$tup_B(id_B)$，$id_B \in ID_B$）の属性の値に関する情報が組織 A に対して漏洩しない．

（証明）アルゴリズム 5 において，組織 B から組織 A に情報が送られるのは，ステップ 7 とステップ 32 の 2 カ所のみであるため，これらの送信情報に着目する．

ステップ 7 で送信される情報について：ステップ 7 の送信情報 T_B は，ステップ 5 のように，ID 集合 ID_B の情報であり，属性値の情報は含まれない．

ステップ 32 で送信される情報について：組織 B は，ステップ 25 で完全準同型暗号の秘密鍵を用いて，組織 A と

[*4] 属性の値域（とりうる値の集合）は秘匿の対象ではなく，提案方式の実行時点における属性の値が秘匿の対象である．

© 2020 Information Processing Society of Japan

B の ID 集合およびタプルデータの属性値を反映した $d_{i,j}$ を復号し，クロス集計表（CT）を生成する．組織 B はステップ 32 において，クロス集計表（CT）の値に，加法準同型暗号の加法性を利用して乱数を加算し，組織 A に送る．組織 A は，ステップ 34 において，加法準同型暗号の秘密鍵を用い，クロス集計表（CT）の値を復号する．しかし，CT の値は，ステップの 32 の乱数によって秘匿されているので，組織 A に漏洩しない．

以上から，条件 1 に関する安全性は乱数加算の安全性に帰着するが．4.2 節で述べたように，乱数は $[0, p-1]$ の一様乱数であり，乱数加算後の CT の各セルの値には，p による剰余（$\bmod p$）が適用されるので，乱数加算後の値から加算前の値は漏洩しない．なお，p は加法準同型暗号の平文の法である．

条件 2 組織 B の ID 集合（ID_B）の情報が組織 A に漏洩しない．

（証明）条件 1 の証明中に述べたように，組織 B から組織 A に情報が送られるのは，ステップ 7 とステップ 32 の 2 カ所のみである．

ステップ 7 で送信される情報について：組織 B の ID 集合 ID_B は，ステップ 5 において，組織 B 側で T_B に格納される．その直後のステップ 7 において，T_B は完全準同型暗号を用いて組織 B 側で暗号化され，組織 A に送られる．組織 B は，ステップ 25 までは，完全準同型暗号を復号しない．完全準同型暗号の秘密鍵は組織 B が管理しているので，完全準同型暗号が安全で鍵管理が適切であれば，組織 A は秘密鍵を知ることはない．そのため，組織 A は，ステップ 8〜19 までの間に T_B の値に関して情報を得ることはない．

ステップ 32 で送信される情報については，条件 1 の証明中で述べたように，乱数加算により組織 A に対して秘匿される．以上から，条件 2 に関する安全性は，提案方式の部品となる完全準同型暗号の安全性に帰着する．

条件 3 組織 A と組織 B に共通する ID の集合（$ID_A \cap ID_B$）の情報が組織 A に漏洩しない．

（証明）提案方式は，組織 B の ID 集合（ID_B）が組織 A の ID 集合（ID_A）の部分集合であることを前提としているので，組織 A は $ID_A \cap ID_B = ID_B$ であることを知っている．そのため，条件 3（$ID_A \cap ID_B$ の秘匿性）は，条件 2（ID_B の秘匿性）と等価である．よって，条件 3 に関する安全性は，提案方式の部品となる完全準同型暗号の安全性に帰着する．

条件 4 組織 A のタプルデータ（$tup_A(id_A)$，$id_A \in ID_A$）の属性の値に関する情報が，組織 B に対して，差分プライバシの定めるレベルを超えて漏洩しない．

条件 4 の証明は以下の手順で行う．まず，アルゴリズム 5 のステップ 34 を以下のステップ 34' に変更したアルゴリズムをアルゴリズム 5' とする．

ステップ 34'：CT を復号し，B へ送信

アルゴリズム 5' では，差分プライバシによる保護を行わないので，クロス集計表 CT が組織 B に直接開示される．そのため，組織 B は，クロス集計表に基づき，組織 A のタプルデータの属性の値について情報を得ることが可能である．このとき，以下の条件 4' を証明する．

条件 4' アルゴリズム 5' において，組織 A のタプルデータ（$tup_A(id_A)$，$id_A \in ID_A$）の属性値に関し，組織 B は，出力（クロス集計表）から得られる以外の情報を得られない．

次に，以下の補題 4 を証明する．

補題 4 条件 4' が真のときに，条件 4 が真である．

（条件 4' の証明）組織 A から B に情報が送信されるのは，アルゴリズム 5' のステップ 18 とステップ 34' のみであるため，これらの送信情報に着目する．

ステップ 18 の送信情報について：ステップ 18 で送信される $d_{i,j}$ [*5]は，ステップ 16 において，組織 A 側で加法準同型暗号により暗号化される．この加法準同型暗号は，ステップ 34 において組織 A 側の秘密鍵を用いて復号される．そのため，加法準同型暗号が安全で鍵管理が適切であれば，ステップ 20〜32 における組織 B 側の処理の間，$d_{i,j}$ は漏洩しない．

ステップ 34' の送信情報について：送信情報はクロス集計表 CT であり，アルゴリズム 5' において，秘匿しない情報である．以上から，条件 4' に関わる安全性は，提案方式の部品である加法準同型暗号の安全性に帰着する．

（補題 4 の証明）アルゴリズム 5' とアルゴリズム 5 の相違は，ステップ 34 において，差分プライバシのためのラプラスノイズを重畳する点のみである．3.1.3 項で述べたように，このラプラスノイズは式 (2) によってランダムに生成される．尺度パラメータ σ は，統計量の定義，タプルデータの形式と値域，および ϵ のみに依存し，属性値とは独立であるため，ラプラスノイズには属性値の情報は反映されない．そこで，アルゴリズム 5 において，組織 B は属性値に関して，アルゴリズム 5' を超える情報を知ることはできない．以上から，ステップ 34 で出力にラプラスノイズが重畳されていても，組織 B は，出力（ラプラスノイズを重畳したクロス集計表）から得られる以外の情報を得られない．ラプラスノイズは，ϵ–差分プライバシを満たすように生成され，クロス集計表に重畳されている．以上から，条件 4' が真ならば，条件 4 も真である．

条件 4' と補題 4 の両者が証明されたことにより，条件 4 が証明された．すなわち，条件 4 に関する安全性は，提案方式の部品である加法準同型暗号の安全性に帰着する．

条件 5 組織 A の ID 集合（ID_A）のうち共通集合（$ID_A \cap ID_B$）以外の ID の情報が，組織 B に対して，差分プ

[*5] 厳密には，$d_{i,j}$ のうち組織 A のタプルデータの属性値 $tup_A(T_A[i][k])$

© 2020 Information Processing Society of Japan

ライバシの定めるレベルを超えて漏洩しない．

条件 5 は，条件 4 と同様の手順で証明する．すなわち，条件 4 の証明に用いたアルゴリズム 5’ を想定し，以下の条件 5’ を証明する．

条件 5’ アルゴリズム 5’ において，組織 A の ID 集合（ID_A）のうち共通集合（$ID_A \cap ID_B$）以外の ID の情報に関し，組織 B は，出力（クロス集計表）から得られる以外の情報を得られない．

その後，以下の補題を証明する．

補題 5 条件 5’ が真のときに，条件 5 が真である．

以下では，条件 4 の証明と同様であるため，簡単に説明する．

（証明 5’ の証明）アルゴリズム 5’ において，組織 A から B に情報が送信されるのは，ステップ 18 とステップ 34’ のみであるため，これらの送信情報に着目する．

ステップ 18 の送信情報について：加法準同型暗号が安全で鍵管理が適切であれば，ステップ 20〜32 における組織 B 側の処理の間，組織 A の ID 集合のうち共通集合以外の ID の情報は漏洩しない．

ステップ 34’ の送信情報について：送信情報はクロス集計表 CT であり，アルゴリズム 5’ において，秘匿しない情報である．以上から，条件 5’ に関わる安全性は，提案方式の部品である加法準同型暗号の安全性に帰着する．

（補題 5 の証明）補題 4 と同様の手順により証明される．

条件 5’ と補題 5 の両者が証明されたことにより，条件 5 が証明された．すなわち，条件 5 に関する安全性は，提案方式の部品である加法準同型暗号の安全性に帰着する．

7. 結論と今後の課題

組織間データ利用は，単一組織では実現しえないサービスの実現が期待できる．その中でも組織どうしが互いのデータをクロス集計することを考え，匿名加工をともなう 2 パーティ秘匿クロス集計について検討した．

本稿では，完全準同型暗号を用いる方式を提案した．加法準同型暗号文の完全準同型暗号文への格納やフェルマーの小定理を応用した等号判定を行うことで，完全準同型暗号を用いた匿名加工をともなう 2 パーティ秘匿クロス集計を実現した．さらに，完全準同型暗号を用いることによる計算量の増加抑制のために Chen らの PSI プロトコル [8] での高速化技術を応用し，計算量と通信量を削減した．加えて，ラプラスノイズ重畳による破綻の確率を考慮した加法準同型暗号へのパッキングの工夫により，計算量と通信量を削減した．

千田らの方式 [3] との理論比較および実装比較を行った．理論比較では，計算量のオーダを変えずに通信量のオーダから最も支配的な変数 N_A を除去した．この要因は，名寄せ処理の多くを組織 A 側で完結させ，N_A に大きさが依存するデータの通信処理を除去したことである．実装比較では，提案方式は千田らの方式 [3] に比べ，LAN 環境で 86.1%，WAN 環境で 77.7%の総実行時間となった．この要因は，通信量の削減による通信時間の短縮と計算処理の多くを高性能なサーバを持つ組織 A 側で実行することによる計算時間の短縮のためである．

今後の課題としては，本稿では山口ら [1] のユースケースを取り上げたが，データのサイズや形式を変化させた場合，具体的にどのようなケースでは提案方式が優れているのかを明らかにする．また，一方の組織への共通集合の開示を許容しない方式の検討に取り組む．

参考文献

[1] 山口高康，寺田雅之：セキュアスムージング手法による組織間プライバシ保護リコメンドシステム，情報処理学会論文誌，Vol.56, No.9, pp.1754–1769 (2015).

[2] Vaidya, J. and Clifton, C.: Privacy Preserving Naïve Bayes Classifier for Vertically Partitioned Data, *Proc. SIAM 2004*, pp.522–526 (2004).

[3] 千田浩司, 寺田雅之, 山口高康ほか：統計的開示制御を考慮したセキュアマッチングプロトコル，Vol.2011-CSEC-52, No.12 (2011).

[4] Gentry, C.: Fully Homomorphic Encryption using Ideal Lattices, *Proc. STOC 2009*, pp.169–178 (2009).

[5] Yao, A.: Protocols for secure computations, *Proc. SFCS 1982*, pp.160–164 (1982).

[6] Chida, K., Genkin, D., Hamada, K., et al.: Fast Large-Scale Honest-Majority MPC for Malicious Adversaries, *Proc. CRYPTO 2018*, pp.34–64 (2018).

[7] 佐久間淳，陸 文傑：準同型暗号を用いた秘密計算技術と実用化に向けた活動，情報処理学会論文誌特集，Vol.59, No.10, pp.898–903 (2018).

[8] Chen, H., Laine, K. and Rindal, P.: Fast Private Set Intersection from Homomorphic Encryption, *Proc. CCS 2017* (2017).

[9] Lu, W., Kawasaki, S. and Sakuma, J.: Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data, *Proc. NDSS 2017* (2017).

[10] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography, *Proc. STOC 2005*, pp.84–93 (2005).

[11] Lyubashevsky, V., Peikert, C. and Regev, O.: On Ideal Lattices and Learning with Errors over Rings, *Proc. EUROCRYPT 2010*, pp.1–23 (2010).

[12] Smart, N. and Vercauteren, F.: Fully homomorphic SIMD operations, *Designs, Codes and Cryptography*, Vol.71, No.1, pp.57–81 (2012).

[13] Brakerski, Z., Gentry, C. and Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping, *Proc. ITCS 2012*, pp.309–325 (2012).

[14] Microsoft Research: Simple Encrypted Arithmetic Library (release 3.0.0), available from ⟨https://github.com/Microsoft/SEAL⟩

[15] Halevi, S. and Laine, K.: HElib, available from ⟨http://shaih.github.io/HElib/⟩

[16] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *Proc. EUROCRYPTO 1999*, LNCS 1592, pp.223–238 (1999).

[17] Kolesnikov, V., Matania, N., Pinkas, B., et al.: Practical Multi-party Private Set Intersection from Symmetric-

© 2020 Information Processing Society of Japan

Key Techniques, *Proc. CCS 2017*, pp.1257–1272 (2017).
[18] Agrawal, R., Evfimievsski, A. and Srikant, R.: Information sharing across private databases, *ACM SIGMOD 2003*, pp.86–97 (2003).
[19] 千田浩司，五十嵐大，濱田浩気ほか：匿名等結合プロトコルとその応用，*Proc. SCIS 2011* (2011).
[20] Sweeney, L.: K-anonymity: A model for protecting privacy, *IJUFKS*, Vol.10, No.5, pp.557–570 (2002).
[21] Machanavajjhala, A., Kifer, D., Gehrke, J., et al: l-diversity: Privacy beyond k-anonymity, *TKDD*, Vol.1, No.1 (2007).
[22] Dwork, C.: Differential privacy, *Proc. ICALP*, Vol.4052, pp.1–12 (2006).
[23] Pagh, R. and Rodler, F.: Cuckoo hashing, *Proc. ESA 2001*, pp.121–133 (2001).
[24] Arbitman, Y., Naor, M. and Segev, G.: Backyard cuckoo hashing: Constant worst-case operations with a succinct representation, *Proc. FOCS 2010*, pp.787–796 (2010).
[25] Halevi, S. and Shoup, V.: Algorithms in HElib, ICC, *Proc. CRYPTO 2014*, LNCS, Vol.8616, pp.554–571, Springer (2014).
[26] 清　雄一，大須賀昭彦：差分プライバシに基づく大規模クロス集計値匿名化手法の提案，*Proc. CSS 2015* (2015).
[27] 寺田雅之，山口高康，本郷節之：匿名化個票開示への差分プライバシの適用，情報処理学会論文誌，Vol.58, No.9, pp.1483–1500 (2017).
[28] 寺田雅之，鈴木亮平，山口高康ほか：大規模集計データへの差分プライバシの適用，情報処理学会論文誌，Vol.56, No.9, pp.1801–1816 (2015).
[29] Boost project: Boost C++ libraries, available from ⟨https://www.boost.org/⟩
[30] Intel: Intel TBB, available from ⟨https://software.intel.com/en-us/intel-tbb⟩
[31] John Bethencourt: Paillier Library, available from ⟨http://hms.isi.jhu.edu/acsc/libpaillier/⟩
[32] GNU project: GNU Multi-Precision Library, available from ⟨http://gmplib.org/⟩
[33] Mohassel, P. and Rindal, P.: ABY3: A Mixed Protocol Framework for Machine Learning, *Proc. CCS 2018* (2018).
[34] Halevi, S. and Shoup, V.: Faster Homomorphic Linear Transformations in HElib, *Proc. CRYPTO 2018*, pp.93–120 (2018).

推薦文

匿名加工をともなう 2 パーティ秘匿クロス集計を取り上げ，完全準同型暗号を用いた方式を提案しており，従来方式と比べて，総実行時間を大幅に短縮している点が評価できる．また，提案方式は，組織の片方が大量のレコードを有しており，かつ高性能なサーバを有している 2 パーティの関係が非対称な状況に適した方式になっているため，クラウド型サービスとの親和性が高いと思われる点も評価できる．以上のことから，推薦論文に値するため，推薦いたします．

（コンピュータセキュリティ研究会主査　山内 利宏）



片山 源太郎

2018 年電気通信大学情報理工学部総合情報学科卒業．同大学大学院情報理工学研究科情報学専攻博士前期課程在学中．



吉浦 裕　（正会員）

1981 年東京大学理学部情報科学科卒業．日立製作所を経て，2003 年電気通信大学勤務．現在，同大学大学院情報理工学研究科教授．情報セキュリティ，プライバシ保護の研究に従事．博士（理学）．日立製作所社長技術賞（2000 年），情報処理学会論文賞（2005 年，2011 年），システム制御情報学会産業技術賞（2005 年），IEEE IIH-MSPbest paper award（2006 年），日本セキュリティ・マネジメント学会論文賞（2010 年，2016 年），IFIP I3E best paperaward（2016 年）等受賞．電子情報通信学会，日本セキュリティ・マネジメント学会，人工知能学会，システム制御情報学会，IEEE 各会員．本会フェロー．

© 2020 Information Processing Society of Japan