秘匿多肢選択マッチングプロトコルの提案と評価

面 和成 †

† 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
omote@jaist.ac.jp

あらまし　近年，個人情報を秘匿しつつその情報を処理する暗号応用技術が注目されている．その中で，Paillier 暗号等の加法準同型暗号をベースにした PSI (Private Set Intersection) プロトコルが知られている．我々は，そのような PSI をベースとした秘匿多肢選択マッチングシステムの構築を目指す．しかしながら，これまでに提案された既存の PSI を秘匿多肢選択マッチングシステムに適用するにはいくつかの問題がある．本稿では，実用的な観点から，確定的な PSI をベースとした効率的な秘匿多肢選択マッチングプロトコルを提案する．本プロトコルは，honest-but-curious なサーバを導入することにより，参加者 PC での暗号処理を最小限にする．また，本プロトコルはサーバが秘密情報を持たないように設計されているため，たとえサーバが攻撃を受けたとしても秘密鍵や参加者の個人情報が漏洩しない．

Proposal and Evaluation of Private Multiple-Choice Matching Protocol

Kazumasa Omote†

†JAIST
Asahidai 1-1, Nomi-city, Ishikawa 923-1292, JAPAN
omote@jaist.ac.jp

Abstract　The applied cryptography technology which processes the personal information with keeping it secret becomes recently important. Especially, we are familiar with PSI (Private Set Intersection) protocol based on an additive homomorphic encryption such as Paillier cryptography. We aim at construction of the private multiple-choice matching system based on PSI. However, there are some problems when applying the existing PSI to a private multiple-choice matching system. In this paper, we propose an efficient private multiple-choice matching protocol based on a deterministic PSI from a viewpoint of practical use. This protocol minimizes a cryptographic processing on a participant's PC by introducing an honest-but-curious server. Furthermore, even if a server is attacked, a secret key or a participant's private information is not revealed since a server does not have any secret information.

1　はじめに

多くのサービスが電子化されている．そのため，事業者は個人情報を適切に管理し，個人情報の漏洩や改ざん等の防止を徹底し，ユーザに安心感を持ってもらうことが求められている．その一方で，サービス等の向上のため，個人情報を安全に利用することも同時に求められている．そのため，個人情報を秘匿したままで安全に利用する技術が非常に重要である．これにより，たとえサーバ上のデータがサイバー攻撃等によって漏洩したとしても，個人情報が漏れないという安心感をユーザが持てるようになる．

個人情報は，様々なデータに含まれる．例えば，個人調査書はその一例であり，氏名や住所，年齢，血液型などの情報が含まれる．本稿では，特に多肢

選択回答形式の情報を扱う．例えば，都道府県情報は 47 択情報であり，血液型情報は 4 択情報である．我々は，このような多肢選択回答形式の個人情報を秘匿したままでマッチングを行う秘匿多肢選択マッチングシステムを考える．

秘匿多肢選択マッチングシステムとは，多肢選択回答形式の個人情報を秘匿したままマッチングさせるシステムである．我々は，ビッグデータを対象とするような統計処理ではなく，確実な結果を出力させるデータ処理を扱う．そのため，このシステムを実現する際，マッチングの誤判定が発生しないことが必要である．なぜなら，マッチングに誤判定が発生すれば，事実と異なるマッチング結果が出力されるためである．

近年，個人情報を秘匿しつつその情報を処理する暗号応用技術が注目されている．その中で，Paillier 暗号等の加法準同型暗号をベースにした PSI (Private Set Intersection) プロトコルが知られている．PSI プロトコルとは，複数の参加者が大きな集合からいくつかの要素を選択し，参加者が選択した要素の重複のみを出力するプロトコルである．我々は，PSI をベースとした秘匿多肢選択マッチングシステムの構築を目指す．しかしながら，これまでに提案された既存の PSI を秘匿多肢選択マッチングシステムに適用するにはいくつかの問題がある．

本稿では，実用的な観点から，確定的な PSI をベースとした効率的な秘匿多肢選択マッチングプロトコルを提案する．本プロトコルは，honest-but-curious なサーバを導入することにより，参加者 PC での暗号処理を最小限にする．また，本プロトコルはサーバが秘密情報を持たないように設計されているため，たとえサーバが攻撃を受けたとしても参加者の秘密鍵が漏洩しない．

2 関連研究

Freedman ら [4] は，2004 年に，Oblivious Polynomial Evaluation (OPE) と加法準同型暗号を用いて PSI プロトコルを初めて提案した．Kissner ら [7] や Dachman-Soled ら [2] は，[4] をマルチパーティプロトコルに適応させた．また，Camenisch ら [1] は，OPE をベースとして保証された集合 (certified set) に対する PSI を提案した．これは，参加者が正当な集合の要素を選択したことを保証させる方式であり，malicious モデルを扱うものである．これらの既存方式はマッチングの誤判定確率が真にゼロである確定的な PSI プロトコルである．しかしながら，これらの既存方式はいづれも参加者 PC の計算量が効率的でない．具体的には，参加者が選択した要素数を k とした場合，これらの PSI プロトコルの計算量は $\mathcal{O}(k^2)$ となる．

これに対して，Dong ら [3] は，Garbled Bloom Filters をベースとした PSI を構築することにより，2^{20} 個等の膨大な量のデータに対して，参加者 PC における公開鍵ベースの計算量を $\mathcal{O}(\ell)$（ℓ: セキュリティパラメータ）に削減した．これは，ビッグデータに対して効率的に PSI 演算を行えるものである．一方，Kim ら [6] は，集合の要素として，OPE における多項式の解を使用する代わりに素数を使用することにより，参加者 PC の計算量を $\mathcal{O}(k)$ に削減した．この方式は，二者間の PSI を実現している．しかしながら，これらの PSI プロトコルは，誤判定を起こす確率を無視できる程小さくできるが，正当性が確率的である PSI である．

これらの研究とは別に，菊池ら [5] が提案した秘匿リストマッチングプロトコルは，本提案プロトコルに類似する．これは，PSI プロトコル [4] を応用したマッチングプロトコルである．しかしながら，効率的でない PSI をベースとしているため，このプロトコルも効率的であるとはいえない．

3 準備

3.1 要求事項

個人情報の安全管理： 多肢選択結果には参加者の個人情報が含まれる．そのため，サーバには多肢選択結果を秘匿したまま保存しておくことが求められる．さらに，サーバが攻撃される可能性を考慮し，サーバは参加者や自身の秘密鍵を保持しないだけでなく，これら秘密鍵をサーバ上で一切使用しないことも必要である．これにより，サーバの安全管理が容易になる．

確定的マッチングプロトコルの実現： 確かに，確率的 PSI [3, 6] の誤判定確率は，パラメータの設定によって無視できるくらい小さくできる．しかしながら，誤判定を許容できないシステムでの実現を考えると誤判定確率が真にゼロであることが望ましい．

様々な多肢選択回答形式への対応： PSI を秘匿多肢選択マッチングシステムに単純適用すると，yes/no

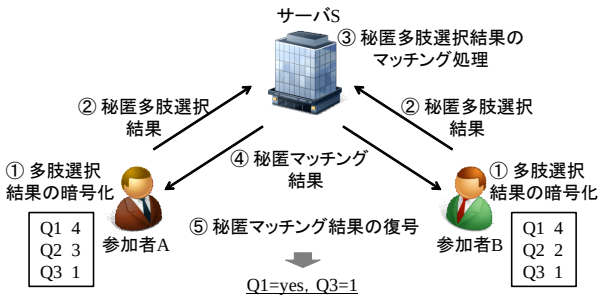

図 1: 秘匿多肢選択マッチングシステム

の二項選択になる．一方で多肢選択回答形式には，単一回答や複数回答など様々な形式が存在する．そのため，秘匿多肢選択マッチングシステムを構築する場合，PSI をこのような多様な多肢選択回答形式に対応させる必要がある．

3.2 Paillier 暗号

以下に，Paillier 暗号方式の各アルゴリズムを示す．

鍵生成アルゴリズム： $n = pq$（p, q は大きな素数）と $g = (1+\alpha n)\beta^n \bmod n^2$ （$\alpha, \beta \in \mathbb{Z}^*_{n^2}$）を計算し，公開鍵 (n, g) を出力する．また，秘密鍵として，$\lambda = lcm(p-1, q-1)$ を出力する．さらに，事前に $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$ を計算しておく．ここで，L は $L(u) = (u-1)/n$ と定義される．

暗号化アルゴリズム： メッセージ $M \in \mathbb{Z}_n$ に対して，乱数 $r \in \mathbb{Z}^*_{n^2}$ を選び，以下のようにして暗号文 $E(M)$ を計算する．

$$E(M) = g^M r^n \bmod n^2$$

復号アルゴリズム： 暗号文 $c = E(M)$ に対して，以下のようにして平文を計算する．

$$\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n = M$$

3.3 セキュリティモデル

セキュリティモデルとして，semi-honest モデルと malicious モデルの 2 種類がある．本稿では，semi-honest モデルを考える．semi-honest モデルとは，サーバ及び全ての参加者がプロトコルに従って振る舞うモデルのことである．また，サーバと参加者は結託しないものとする．このモデルの安全性では，下記のとおり正当性と秘匿性を満たす必要がある．

- 正当性 二人の参加者が各多肢選択のマッチング結果を正しく出力するなら，このプロトコルは正当性をもつ．
- 秘匿性 マッチング結果に存在しない各参加者の多肢選択項目について何も知ることが出来ないなら，このプロトコルは秘匿性をもつ．

4 秘匿多肢選択マッチングシステム

秘匿多肢選択マッチングシステムとは，二者の参加者間で各質問に対する多肢選択を行い，各参加者がマッチしたものだけを得るシステムである．サーバ S 及び参加者 A, B は，honest-but-curious なエンティティであると仮定する．つまり，S，A，B は，多肢選択結果に関心はあるが，プロトコルに違反するような不正は行わないものとする．また，S は，A, B との結託を行わないことを前提とする．

図 1 に，秘匿多肢選択マッチングシステムの概念図を示す．秘匿多肢選択マッチングシステムの基本的な手順は以下の通りである．

1. A, B は，多肢選択結果を暗号化し，それぞれが暗号化した結果を S に送信する．
2. S は，秘匿多肢選択結果のマッチング処理を暗号化したままで行い，その結果を A, B に返す．
3. A, B は，秘匿マッチング結果をそれぞれで復号し，マッチング結果を知る．

図 1 の例では，参加者 A と B のマッチング結果は Q1 の 4 と Q3 の 1 であるため，A と B はこれら 2 つのマッチング結果のみを知ることができる．しかし，マッチしなかった Q2 に関する結果を得ることができない．

多肢選択回答形式には次の 2 種類がある．

- 単一回答（二項選択，多項選択）
- 複数回答（無制限複数回答，制限付き複数回答）

単一回答は，選択肢の中から 1 つだけを選ぶ回答形式である．その中で，二項選択は 2 つの選択肢の中から 1 つを選ぶものであり，多項選択は 3 つ以上

の選択肢の中から 1 つを選ぶものである．一方，複数回答は，選択肢の中から 2 つ以上を選べる回答形式である．その中で，無制限複数回答はいくつでも選択でき，制限付き複数回答は選択する数に制限がある．

5 PSI 方式 [6] の問題点

既存の PSI 方式 [6] は，シンプルで効率的な方式である．しかしながら，[6] には以下の 3 つの問題点がある．

マッチングの誤判定： [6] が確率的 PSI であるため，マッチングしていないにも関わらずマッチングしていると誤判定する恐れがある．具体的には，参加者 A,B において，共通の素因数の有無でマッチングを判定するが，計算過程に加算があるため想定しない素因数が紛れ込む可能性がある．この問題は，[6] で言及されていない．

サーバが参加者の秘密鍵を保持： [6] は分散復号を行う．サーバは，参加者のシェアを生成する必要があるため，Paillier 暗号の秘密鍵 λ を保持する．そのため，サーバが攻撃されて λ が攻撃者に漏洩すれば，全参加者の暗号文が解読される恐れがある．

複雑な参加者 PC 処理： 暗号化の際は参加者同士で送受信，シェアを取得する際はサーバとの通信，さらに復号の際は参加者同士で送受信が必要である．このため，参加者 PC における処理フローが煩雑となる．

さらに，[6] では多肢選択回答形式に対応していない．これらの問題点により，実用的な観点から，[6] をそのまま秘匿多肢選択マッチングシステムに適用できない．

6 提案方式

本章では，semi-honest な攻撃者に対して安全な秘匿多肢選択マッチングプロトコルを提案する．提案方式は基本的には [6] をベースとするが，以下の 3 つの観点で [6] と異なる：(1) サーバがマッチングを誤判定しないようなパラメータを設定できる．(2) サーバが参加者や自身の秘密鍵を持たない．(3) 多肢選択回答形式に対応する．なお，サーバを設けることにより，参加者 PC の処理を最小限にする．

6.1 多肢選択回答形式への対応

PSI を秘匿多肢選択マッチングシステムに単純適用すると yes/no の二項選択になる．その結果，秘匿多肢選択マッチングシステムでは，二項選択結果は参加者の相手にばれてしまう．なぜなら，マッチングしない場合は相手と異なるという情報が漏れてしまうためである．そのため，本稿では PSI を拡張して多肢選択を扱えるように改良している．

6.2 プロトコル詳細

図 2 をベースに，二者間の秘匿多肢選択マッチングプロトコルの詳細を述べる．この図では，質問 j $(j = 1, \ldots, k)$ に対するプロトコルを記載している．参加者 A, B は，多肢選択結果に対応する素数 $X_A = \{a_1, \ldots, a_\gamma\} \subset \mathcal{S}_j$, $X_B = \{b_1, \ldots, b_{\gamma'}\} \subset \mathcal{S}_j$ をそれぞれ選択するとする（$\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_k = \mathcal{S}$, $\mathcal{S}_{j_1} \cap \mathcal{S}_{j_2} = \emptyset$, $1 \leq j_1, j_2 \leq k$）．$\mathcal{S}$ は選択可能な t ビット素数の公開の集合を表し，$\mathcal{S}_j (\subset \mathcal{S})$ は質問 j における選択可能な t ビット素数の公開の集合を表す．多肢選択結果は素数の積で表され，A, B の結果をそれぞれ $a = \prod_{i=1}^{\gamma} a_i$, $b = \prod_{i=1}^{\gamma'} b_i$ と表す．A と B は対称的であるため，ここでは A がマッチング結果を得る手順を主に説明する．なお，A と B は認証された通信路を利用することを仮定し，A と B が直接通信を行わない点が [6] と異なることに注意する．

1. S は，k 個の質問に対応して，集合 $\{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ を選択する．
2. A, B は，加法準同型暗号を設定し，それぞれ自身の公開鍵 pk_A, pk_B を公開する．
3. A は，自身と B の公開鍵のそれぞれを用いて，自身の多肢選択結果を暗号化することによって $E_{pk_A}(a^\phi)$ と $E_{pk_B}(a^\phi)$ を求める（$\phi \in \mathbb{Z}$）．それから，A はこれらを S に送信する．B も同様に多肢選択結果を暗号化し，それらを S に送信する．
4. S は，次の 2 つの条件を満たす乱数 $r_{aa}, r_{ab}, r_{ba}, r_{bb} \in \mathbb{Z}_\ell$ を生成する．ただし，ℓ はセキュリティパラメータである．
   - （**条件 1**）$\forall x \in \mathcal{S}_j$ に対して，$r_{aa}, r_{ab}, r_{ba}, r_{bb}$ がいづれも，x^ϕ を素因数に持たない．
   - （**条件 2**）AB 共通の素因数の積を π としたとき，$\forall x \in \mathcal{S}_j$ に対して，X_{AB}/π^ϕ,

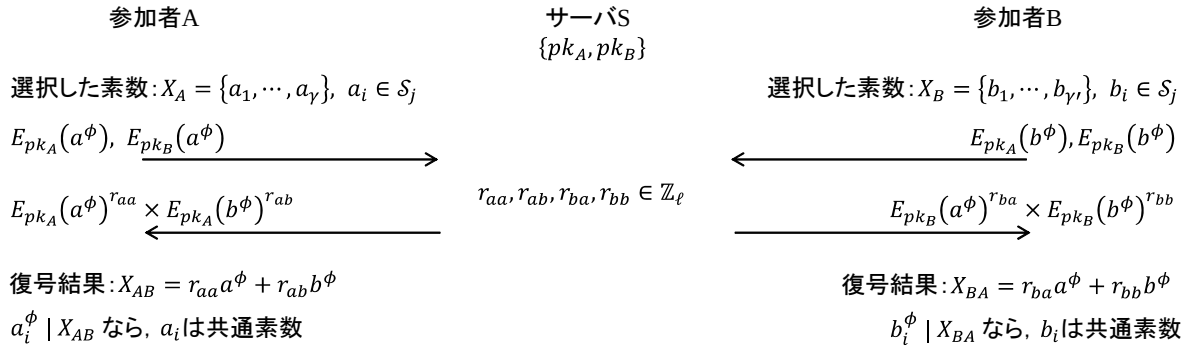

図 2: 質問 j に対する二者間の秘匿多肢選択マッチングプロトコル

X_{BA}/π^ϕ の全候補が，x^ϕ を素因数に持たない．

A に対しては，S は乱数 r_{aa}, r_{ab} を用いて，次の式 (1) を計算する．

$$E_{pk_A}(a^\phi)^{r_{aa}} \times E_{pk_A}(b^\phi)^{r_{ab}} = E_{pk_A}\left(r_{aa}a^\phi + r_{ab}b^\phi\right) = E_{pk_A}\left(X_{AB}\right) \quad (1)$$

その後 S は，式 (1) の値を A に送信する．このとき，$X_{AB} = r_{aa}a^\phi + r_{ab}b^\phi$，$X_{BA} = r_{ba}a^\phi + r_{bb}b^\phi$ と表される．

5. A は，式 (1) の値を復号して X_{AB} を得る．
6. A は，自身の多肢選択結果を示す全ての素数に対して，マッチング検証を行う，すなわち，$a_i^\phi \mid X_{AB}\ (i = 1, \ldots, \gamma)$ をチェックする．これが真であるならば，a_i は共通素数ということであり，偽であるならば a_i は共通素数ではないということになる．X_A の全ての要素でチェックすることにより，A は B とのマッチング結果を得ることができる．B も同様である．

なお，二者間のプロトコルを並列で動作させて，m 人におけるプロトコルを容易に構築できる．

6.3 制限付き複数回答の実現

制限付き複数回答を実現するには，部分的に malicious モデルに対応する必要がある．つまり，参加者が制限された選択数よりも多く選択出来ないようにする新たな仕組みが必要がある．本提案方式では，乱数を利用してメッセージ空間をコントロールする工夫を行う．

1 つの暗号文に詰め込むことのできる素数の数に制限があり，これらの素数の積のビットサイズは ℓ のサイズに依存することに注意する．具体的には，素数の積は $\frac{n-\ell-1}{\phi}$ ビット未満である必要がある．もし $\frac{n-\ell-1}{\phi}$ ビット以上の素数の積を用いる場合は，X_{AB} のサイズがメッセージ空間（$\mathbb{Z}_n$）を超えてしまうため，参加者 A は X_{AB} を復号出来なくなる．ゆえに本提案方式では，サーバ S は ℓ のサイズをコントロールすることによって，素数の積のビットサイズを制限できる．これにより，制限付き複数回答を実現でき，相手の個人情報を最大限にマッチさせて情報を引き出すという参加者の不正 [1] を防ぐ効果がある．なお，ℓ の具体的な適用については 7.2.2 節の実装評価を参照されたい．

7 評価

7.1 安全性

7.1.1 正当性

Theorem 1 (正当性)**.** 提案プロトコルはマッチング結果を正しく出力する．

Proof. まず，$x \in X_A \cap X_B$ と仮定する．このとき，x^ϕ は a^ϕ かつ b^ϕ を割り切る．ゆえに，x^ϕ は $X_{AB} = r_{aa}a^\phi + r_{ab}b^\phi$，及び $X_{BA} = r_{ba}a^\phi + r_{bb}b^\phi$ の両方を割り切る．したがって，各参加者は x が共通選択結果であることを知る．次に，$x \notin X_A \cap X_B$ と仮定する．このとき，(1) どちらか一方だけに含まれる，(2) どちらにも含まれない，の 2 つのケースを考える必要がある．(1) においては，例えば $(x \notin X_A) \cap (x \in X_B)$ としたとき，x^ϕ が b^ϕ を割り切るが a^ϕ を割り切らない．さらに，x^ϕ が r_{aa} と r_{ba} を割り切らない．よっ

て，x^ϕ は X_{AB} 及び X_{BA} を割り切らない．(2) においては，$(x \notin X_A) \cap (x \notin X_B)$ であるため，x^ϕ が a^ϕ も b^ϕ も割り切らない．また，x^ϕ が $r_{aa}, r_{ab}, r_{ba}, r_{bb}$ のいづれも割り切らない．よって，x^ϕ は X_{AB} 及び X_{BA} を割り切らない．最後に，x^ϕ が X_{AB}/π^ϕ 及び X_{BA}/π^ϕ を割り切らない．したがって，提案プロトコルはマッチング結果を正しく出力する． □

この定理は，提案方式の正当性が確定的であることを示している（表 1 参照）．6.2 節で述べたとおり，x^ϕ が X_{AB}/π^ϕ 及び X_{BA}/π^ϕ を割り切らないように，S が $r_{aa}, r_{ab}, r_{ba}, r_{bb}$ を設定することに注意する．

7.1.2 秘匿性

まずは，不定方程式に関する定理を Lemma として以下に示す．

Lemma 1. $gcd(a', b') = 1$ のとき，$a'x + b'y = d$ の解 (x, y) が必ず存在する．

これは不定方程式の有名な定理であるため，ここでは証明を省略する．

次の Theorem 2 は，提案プロトコルが秘匿性を有していることを示す．

Theorem 2 (秘匿性)**.** 攻撃者は，正直な参加者との共通の選択項目以外で正直な参加者の選択について何も有益な情報を得ることができない．

Proof. 参加者は semi-honest である．ここで，参加者 A を正直な参加者，参加者 B を攻撃者であると仮定する．すなわち，B は A の多肢選択結果を知ろうとする．このとき B は，b, $E_{pk_A}(b^\phi)$, $E_{pk_B}(b^\phi)$, 及び $X_{BA} = r_{ba}a^\phi + r_{bb}b^\phi$ を得ることができる．B は，A の多肢選択結果を知るために，$X_{BA} = r_{ba}a^\phi + r_{bb}b^\phi$ から A だけが選択した素因数を見つけなければならない．ここで，$X_{BA} = \pi^\phi(r_{ba}(a')^\phi + r_{bb}(b')^\phi)$ と書ける（$a = \pi a', b = \pi b'$）．このとき，B は X_{BA}, $(b')^\phi$, π^ϕ を知っている．ゆえに，$(b')^\phi$ と X_{BA}/π^ϕ が与えられたとき，B は，

$$r_{ba}(a')^\phi + r_{bb}(b')^\phi = X_{BA}/\pi^\phi \qquad (2)$$

の関係式から $(a')^\phi$ を求めなければならない．ここで，a', b' が共通の素因数を持たないことから $\gcd((a')^\phi, (b')^\phi) = 1$ が成り立つ．Lemma 1 より，式 (2) の $(a')^\phi$ に全ての候補を当てはめたとしても r_{ba} と r_{bb} が必ず存在する．これは，乱数空間 ($\mathbb{Z}_\ell$) の exhaustive search を必要とするため，$(a')^\phi$ を多項式時間で決めることが難しい，すなわち a' を多項式時間で決めることが難しいことを意味する． □

表 1: 正当性及び効率性の比較

方式	正当性	計算量	通信量
[4, 7]	確定的	$\mathcal{O}(k^2)$	$\mathcal{O}(k)$
[3]	確率的	$\mathcal{O}(\ell)$	$\mathcal{O}(k)$
[6]	確率的	$\mathcal{O}(k)$	$\mathcal{O}(k)$
提案方式	確定的	$\mathcal{O}(k)$	$\mathcal{O}(k)$

7.2 効率性

本節では，各参加者 PC 及びサーバ PC の計算量及び通信量について評価し，特に計算量については実装評価を行う．また，サーバ PC の処理時間で支配的である乱数チェックの処理時間を具体的に評価し，さらにサーバが生成する乱数が 6.2 節の 4 に記載の 2 つの条件を満たさない確率（乱数生成の失敗確率）が低いことを理論的に示す．

7.2.1 通信量及び計算量

二者間の秘匿多肢選択マッチングプロトコルにおいて，参加者は各質問に対して 2 つの暗号文を送受信する．提案方式では各参加者が送受信する暗号文の通信回数は $4k$ となる．ゆえに，提案方式の通信量は $\mathcal{O}(k)$ で表される．

計算量については，最も処理が重いべき乗剰余の回数で評価する．Paillier 暗号では，暗号化にべき乗剰余が 2 回必要であり，復号にべき乗剰余が 1 回必要である．提案方式では，各質問に対して 2 回の暗号化と 1 回の復号が必要となるため，各参加者のべき乗剰余回数は $5k$ 回となる．ゆえに，提案方式の計算量は $\mathcal{O}(k)$ で表される．

表 1 は，既存の PSI 方式との正当性及び効率性の比較である．これより，提案方式では，正当性が確定的であり，かつ計算量，通信量が共に低いことが明らかになった．

7.2.2 実装評価

実システムにおける演算処理時間を見積もるために提案方式の実装評価を行う．そのため，サーバ PC 及び参加者 PC における具体的な演算処理時間を測定する．具体的なパラメータに関して，まず表

表 2: 各質問に対するサーバ PC の処理時間 (CPU: Intel Xeon 2.67GHz/4-core)

$\|\mathcal{S}_j\|$	乱数生成 (4 回分)	乱数チェック			べき乗剰余 (4 回分)
		$\gamma=1$	$\gamma=2$	$\gamma=3$	
10	0.870ms	1.53ms	55.1ms	358ms	22.9ms
20		11.5ms	2.11sec	83.6sec	
30		37.9ms	17.0sec	(*)	
40		88.7ms	73.9sec	(*)	
50		173ms	229sec	(*)	

表 3: 各質問に対する参加者 PC の処理時間 (CPU: Intel Atom 2.00GHz/1-core)

暗号化 (2 回分)	復号 (1 回分)	マッチング検証		
		$\gamma=1$	$\gamma=2$	$\gamma=3$
154ms	71.3ms	0.37ms	1.51ms	2.36ms

表 4: X_{AB}/π^{ϕ} (X_{BA}/π^{ϕ}) における全候補数 (δ)

$\|\mathcal{S}_j\|$	$\gamma=1$	$\gamma=2$	$\gamma=3$
10	55	1,885	12,175
20	210	37,545	1,471,665
30	465	204,480	20,749,095
40	820	670,190	129,530,090
50	1,275	1,672,175	526,065,275

5 における乱数生成の失敗確率が 1%未満となるように $t = 16$, $\phi = 2$ を設定した（表 5 の説明は 7.3 節参照）．次にセキュリティパラメータとして，$|n| = 1024$ を設定し，ℓ は γ の値に応じて変化させる．例えば $t = 16$ の場合，サーバは $\ell = 991(\gamma = 1)$, $\ell = 959(\gamma = 2)$, $\ell = 927(\gamma = 3)$ と設定する．また，都道府県（47 択情報）の多項選択の実現を考慮し，$|\mathcal{S}_j| = 10, 20, 30, 40, 50$ を設定する．$|\mathcal{S}_j|$ と γ のパラメータは，「$|\mathcal{S}_j|$ 個の項目から γ 個を選択する」という複数回答を意味する．

実装評価では，サーバ PC と参加者 PC のスペックを分けてそれぞれ評価を行う．サーバ PC は，CPU が Intel Xenon 2.67GHz/4-core，メモリが 16GB，OS が Windows 7 (64-bit) である．参加者 PC は，CPU が Intel Atom 2.00GHz/1-core，メモリが 2GB，OS が Windows XP (32-bit) である．なお，本実装は OpenSSL を用いて C 言語で行った．

サーバは，各質問に対して，4 回の乱数生成，乱数チェック，及び 4 回のべき乗剰余演算を行う．表 2 は，各質問に対するサーバ PC の処理時間（100 回平均）である．その結果，γ が大きくなるにつれて，乱数のチェックに要する時間が支配的になることが明らかになった．なお，表 2 内の (*) の部分は，乱数チェックに 25 分以上要した箇所である．

参加者は，各質問に対して，2 回の暗号化，1 回の復号，γ 回のマッチング検証を行う．表 3 は，各質問に対する各参加者 PC の処理時間（100 回平均）である．これらの処理時間は $|\mathcal{S}_j|$ の値に依存しない．マッチング検証時間は γ の値に依存するが，暗号演算と比べてわずかな処理時間であり，暗号化・復号処理の時間が支配的であることが表 3 より明らかになった．

7.3 乱数チェックの処理時間

提案方式では，サーバ S がマッチングの誤判定を起こさないような適切な乱数を選択する．そのため，サーバは選択した乱数が 6.2 節の 4 に記載の 2 つの条件を満たしているかをチェックする．より具体的には，サーバは参加者 A に対して，r_{aa}, r_{ab} の乱数チェック，及び $|\mathcal{S}_j|$ と γ の値に応じた X_{AB}/π^{ϕ} の全候補に対する乱数チェックを行う．なお，π は AB 共通の素因数の積である．

乱数チェックの処理時間は，$|\mathcal{S}_j|$ または γ の値で大きく変わる（表 2 参照）．$|\mathcal{S}_j|$ を 10 から 50 まで変化させたとき，表 2 は乱数チェックの処理時間（100 回平均），表 4 は乱数チェックの回数を表している．この結果により，本プロトコルは γ が比較的小さい値に限り実現可能であることが明らかになった．

7.4 乱数生成の失敗確率

もし選択された乱数が 6.2 節の 4 に記載の 2 つの条件を満たさないなら，サーバ S は乱数を再選択し

表 5: t にを変化させたときの素数の個数及び乱数生成の失敗確率（$\phi = 2$）

t		12	13	14	15	16	17
素数の個数		255	464	872	1,612	3,030	5,709
P_f	$\|\mathcal{S}_j\| = 10,\ \gamma = 3$	1.44%	0.362%	0.0907%	0.0227%	0.00567%	0.00142%
	$\|\mathcal{S}_j\| = 20,\ \gamma = 3$	82.7%	35.5%	10.4%	2.70%	0.683%	0.171%
	$\|\mathcal{S}_j\| = 30,\ \gamma = 2$	21.6%	5.91%	1.51%	0.380%	0.0952%	0.0238%
	$\|\mathcal{S}_j\| = 40,\ \gamma = 2$	55.0%	18.1%	4.87%	1.24%	0.312%	0.0780%
	$\|\mathcal{S}_j\| = 50,\ \gamma = 2$	86.4%	39.2%	11.7%	3.07%	0.776%	0.194%
乱数チェック時間（$\|\mathcal{S}_j\| = 10,\ \gamma = 3$）		361ms	358ms	359ms	356ms	358ms	361ms

なければならない．そのため，これら条件に合わない確率（乱数生成の失敗確率）が低いことが望ましい．そこで本節では，この確率を理論的に見積もる．

参加者 A の質問 j を対象に考える．ある数が，$\forall a_i \in \mathcal{S}_j$ に対して a_i^ϕ を素因数に持つ確率（乱数生成の失敗確率）は以下である．

$$P_1 = 1 - \left(1 - \frac{1}{2^{\phi t}}\right)^{|\mathcal{S}_j|} \tag{3}$$

ただし，ある数が t ビット素数を約数に持つ確率は，その数のビット長に寄らず $\frac{1}{2^t}$ と一定であることに注意する．さらに，X_{AB}/π^ϕ 及び X_{BA}/π^ϕ の全候補において，a_i^ϕ を素因数に持つ確率は以下である．

$$P_2 = 1 - (1 - P_1)^{2\delta} \tag{4}$$

ただし，δ は表 4 に記載されている全候補数の値となる．δ の値が $|\mathcal{S}_j|$ と γ から一意に求められることに注意する．ゆえに，r_{aa}，r_{ab}，X_{AB}/π^ϕ，及び X_{BA}/π^ϕ の全候補の少なくとも 1 つが a_i^ϕ を素因数に持つ確率（乱数生成の失敗確率）は以下となる．

$$P_f = 1 - (1 - P_1)^4(1 - P_2) = 1 - (1 - P_1)^{2\delta+4} \tag{5}$$

乱数生成の失敗確率 (P_f) を表 5 に整理する．γ の値が大きいほど確率が高くなるため，表 2 より現実的に設定可能である最大の γ を選択している．また式 (5) より，この確率は δ の値に大きく依存する．

本稿では $\phi \geq 3$ の場合の検証は行っていないが，$\phi = 3$ にすることによって，乱数チェックの失敗確率を大幅に低減出来ることが明らかになっている．

8 まとめ

本稿では，semi-honest な攻撃者に対して安全であり，かつ確定的な PSI をベースとした効率的な秘匿多肢選択マッチングプロトコルを提案した．特に本提案方式は，単一回答だけでなく，制限付き複数回答の実現に成功している．なお，malicious モデルへの適用は，[6] と同様，ゼロ知識証明を用いて実現可能である．

参考文献

[1] J. Camenisch and G.M. Zaverucha, “Private Intersection of Certified Sets”, In *FC 2009*, pp.108-127, 2009.

[2] D. Dachman-Soled, T. Malkin, M. Raykova and M. Yung, “Efficient Robust Private Set Intersection”, In *ACNS 2009*, pp.125–142, 2009.

[3] C. Dong, L. Chen and Z. Wen, “When private set intersection meets big data: an efficient and scalable protocol”, In *CCS 2013*, pp.789–800, 2013.

[4] M.J. Freedman, K. Nissim and B Pinkas, “Efficient Private Matching and Set Intersection”, In *EUROCRYPT 2004*, pp.1–19, 2004.

[5] H. Kikuchi and D. Kagawa, “秘匿リストマッチングプロトコルとその応用”, In *ICSS 2009-7*, pp.33–36, 2009.

[6] M. Kim, H.T. Lee and J.H. Cheon, “Mutual Private Set Intersection with Linear Complexity”, In *WISA 2011*, pp. 219–231, 2011.

[7] L. Kissner and D.X. Song, “Privacy-Preserving Set Operations”, In *CRYPTO 2005*, pp. 241–257, 2005.