入力クラスが制限された能動的攻撃者に対する秘匿積集合計算

笠島 悠吾[1,a)]　杉本 航太[1]　岩本 貢[1]　渡邉 洋平[1,2]

概要：秘密計算において，能動的な攻撃者はプロトコルの仕様から任意に逸脱した振る舞いを行うことが可能であり，その結果，特定の情報の漏洩を許容せざるを得ない場合がある．例えば，互いの集合を秘匿したままその積集合を計算する秘匿積集合計算（Private Set Intersection: PSI）において，攻撃者は入力集合を極端に大きな集合に変更することで，相手の入力集合の大部分を取得でき得るが，このような振る舞いも既存の能動的安全性の定義では許容される．本稿ではこのような攻撃の現実的な脅威に着目し，「入力に対する制限」という観点から一般化した新たな能動的安全性を定義する．具体的には，各パーティに対して入力クラスを定義し，各入力クラスの範囲における積集合を出力する PSI に対する能動的安全性を定式化する．さらに，この安全性を満たす PSI プロトコルとして，デジタル署名技術を用いたプロトコルを構成し，その正当性および安全性を示す．

キーワード：秘密計算，能動的安全性，秘匿積集合計算，デジタル署名

Private Set Intersection against Malicious Adversaries with Restricted Input Classes

Yugo Kasashima[1,a)]　Kota Sugimoto[1]　Mitsugu Iwamoto[1]　Yohei Watanabe[1,2]

Abstract: In secure computation, a malicious adversary may arbitrarily deviate from the prescribed protocol, which can inevitably result in information leakage. For instance, in Private Set Intersection (PSI), an adversary can inflate its input set to an extreme size, thereby learning most of the other party's elements. Such behavior, however, is still permitted under existing definitions of malicious security. In this work, we address the practical threat posed by such attacks and introduce a generalized notion of malicious security from the perspective of input restrictions. Specifically, we define input classes for each party and formalize malicious security for PSI with respect to computing intersections within these input classes. We further present a PSI protocol that satisfies this notion by leveraging digital signature techniques, and we prove its correctness and security.

Keywords: Secure Computation, Malicious Security, Private Set Intersection, Digital Signature

1. はじめに

秘匿積集合計算（Private Set Intersection: PSI）は，二者が自身の集合を秘匿したまま積集合のみを得るための暗号プリミティブである．PSI は秘密計算（Multi-Party Computation: MPC）プロトコルの一種であり，その安全性は MPC の定義に準ずる．その安全性は攻撃者モデルによって大別され，プロトコルに従う受動的攻撃者に対する安全性（受動的安全性）とプロトコルに従うとは限らない能動的攻撃者に対する安全性（能動的安全性）が知られており，一般的に後者の方が現実的な安全性といえる．

本研究では，PSI（ひいては MPC）における「プロトコルへの “入力” が決まるタイミング」について，実運用と定式化の間に存在するギャップに着目する．実際に PSI を運

[1] 電気通信大学 / The University of Electro-Communications
[2] 国立研究開発法人 産業技術総合研究所 / AIST
[a)] kasashima@uec.ac.jp

用する際，パーティがそもそも持つ集合があり，それをプロトコルに入力して積集合を計算する，という流れが自然である．ここで，攻撃者が自身の入力を極端に大きい集合（例えば全体集合）に変更してプロトコルを実行すると積集合も極端に大きくなり，PSI で守るべき「積集合に含まれない要素」がほとんど漏洩してしまうことから，こういった攻撃をいかに防ぐかが重要である．一方，能動的安全性の定式化においてはプロトコルに入力された情報が “正当な入力” であるとみなされるため，原理上，上記の攻撃を防ぐことができない．そこで，能動的安全性と上記攻撃への対策を両立し，PSI の有用性を高めることを目指す．

1.1 本研究の貢献

本稿では，能動的安全性を「入力に対する制限」という観点から見直し，上記の原理上避けられない上記攻撃への対策が可能な定義に一般化する．その上で，その一般化した能動的安全性を満たす PSI プロトコルの構成を示す．

具体的には，各パーティ P_i が入力として取れる範囲を $\mathcal{X}_i$ に制限した $(\mathcal{X}_1, \mathcal{X}_2)$-能動的安全性を導入する[*1]．この安全性を満たす PSI プロトコルの設計フレームワークとして，デジタル署名を用いて入力クラス $\mathcal{X}_i$ に認証を与える認証フェーズ，その上で PSI を実行する計算フェーズからなる二段階の枠組みを提案する．このように定めた構成フレームワークの下で，決定的署名または Schnorr 署名 [11] に基づく認証フェーズを想定した上で，計算フェーズを実現する認証機能付き PSI（Certified-input PSI: CPSI）を具体的に構成し，諦めざるを得ない情報の少ない能動的安全性を達成可能な PSI が実現できることを示す．

1.2 関連研究

入力の真正性を前提化・検証する PSI としては，Camenisch と Zaverucha の研究が起源と言える [2]．彼らはあらかじめ認証機関によって認証された集合の要素のみが積集合計算に寄与するような，集合要素に対して外部から認証を与える方向性を早期に位置付けた．その後，[2] と同様の設定で，Schnorr 署名に基づく Authorized PSI が提案された [12]．彼らは，受動的安全性を満たすプロトコルを構成し，正当な署名を持つ要素のみが計算に寄与する構成を具体化した．さらに近年では，Merkle tree による整合性強化を組み合わせ，Authenticated PSI として入力データの完全性を実装面で高める試みも報告されている [5]．

既存研究では認証や完全性に焦点を当てるのに対し，本研究は入力クラスの制限による能動的安全性の一般化を行う点に独自性がある．つまり，本研究は既存方式を包含する形で体系化している．また，本稿で構成する署名ベースの CPSI は既存方式と比べて，単一の認証機関から与えられた署名を受理するのではなく，複数の機関から与えられる署名を受理するという新しいモデルに基づく．さらに，既存研究では Schnorr 署名に基づく方式は受動的安全性のみを満たすが，本稿では能動的安全性を満たす構成を示す．

[*1] $\mathcal{X}_1, \mathcal{X}_2$ を制限しなければ，既存の能動的安全性となる．

2. 準備

2.1 記法

本稿では一貫して κ をセキュリティパラメータとして扱う．プロトコルに参加するパーティを P_1, P_2 とし，それぞれが保持する入力集合を $X_i := \{x_{i,1}, x_{i,2}, \ldots\}$ とする．また，正の整数 a に対して，$[a] := \{1, 2, \ldots, a\}$ とする．

2.2 既存の安全性定義

PSI（MPC）では主に，受動的な攻撃者と能動的な攻撃者という二つの攻撃者モデルが考慮される．

能動的な攻撃者はプロトコルからの任意の逸脱行動が許されている．能動的な攻撃者に対する安全性は，計算したい機能が「信頼できる第三者（Trusted Third Party: TTP）によって計算される理想世界」と，「実際にプロトコルを実行することで計算される現実世界」を用いて定義される．

理想世界における実行: プロトコルに参加するパーティを P_1, P_2 とし，$c \in \{1, 2\}$ を攻撃者 $\mathcal{A}$ によって支配されるパーティのインデックスとする．機能 $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ の理想的な実行は次のように行われる:

入力: P_1, P_2 の入力をそれぞれ x_1, x_2 とする．攻撃者 $\mathcal{A}$ はそれらとは別に補助的な入力 z を持つ．

入力の送信: 正直なパーティ P_j は入力 x_j を TTP に送信する．攻撃者 $\mathcal{A}$ に操作されているパーティ P_c は中止を表すメッセージ abort，x_c，x'_c $(x'_c \neq x_c,\ |x'_c| = |x_c|)$ のいずれかを TTP に送信することができる．P_c の入力の決定は攻撃者 $\mathcal{A}$ によって行われ，P_c の入力値と補助入力 z に依存する可能性がある．TTP に送信される入力のペアを (x'_1, x'_2) とする[*2]．

早期中止: TTP がある $i \in \{1, 2\}$ に対して，abort_i という形式の入力を受け取った場合，全てのパーティに abort_i を送信し，理想的な実行は終了する．そうでない場合，実行は次のステップに進む．

攻撃者への出力: この時点で TTP は $f(x'_1, x'_2) = (f_1(x'_1, x'_2), f_2(x'_1, x'_2))$ を計算し，パーティ P_c に $f_c(x'_1, x'_2)$ を送信する．

継続または停止の選択: 攻撃者 $\mathcal{A}$ は継続を意味するメッセージ $\mathsf{continue}$ か abort_c を TTP に送信する．送信されたメッセージが $\mathsf{continue}$ である場合は，TTP は正直なパーティ P_j に $f_j(x'_1, x'_2)$ を送信する．そうでない場合は TTP は P_j に abort_c を送信する．

[*2] $c = 2$ の場合，$x_1 = x'_1$ であるが，x'_2 は必ずしも x_2 と等しくないことに注意．$i = 1$ の場合はその逆．

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i
入力: P_i は集合 $X_i = \{x_{i,1}, \ldots, x_{i,m}\}$ を入力する．
出力: パーティ P_1 に積集合 $X_1 \cap X_2$ を返す．

図 1 PSI の理想機能 $\mathcal{F}_{\mathsf{psi}}$
Fig. 1 Ideal Functionality of PSI $\mathcal{F}_{\mathsf{psi}}$

出力: 正直なパーティ P_j は TTP から得た出力値を常に出力する．攻撃者の支配下にあるパーティ P_c は何も出力しない．攻撃者 $\mathcal{A}$ は P_c の初期入力 x_c，補助入力 z，TTP から得た値 $f_c(x_1', x_2')$ を入力として，任意の確率的多項式時間計算可能な関数の値を出力する．

$f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$（$f = (f_1, f_2)$）を 2 者間関数，$\mathcal{A}$ を非一様確率多項式時間マシン，$c \in \{1,2\}$ を攻撃者 $\mathcal{A}$ の支配下にあるパーティインデックスとする．そして，入力 (x_1, x_2)，攻撃者 $\mathcal{A}$ への補助入力 z，セキュリティパラメータ κ に対する f の理想的な実行を $\mathrm{IDEAL}_{f,\mathcal{A}(z),c}(x_1, x_2, \kappa)$ とし，上記の理想的な実行からの正直なパーティと攻撃者 $\mathcal{A}$ の出力ペアとして定義する．

現実世界における実行: f を上記のようにし，Π を f を計算するための 2 者間プロトコルとする．さらに，$\mathcal{A}$ を非一様確率的多項式時間マシンとし，$c \in \{1,2\}$ を攻撃者の支配下にあるパーティのインデックスとする．そして，入力 (x_1, x_2), 攻撃者 $\mathcal{A}$ への補助入力 z, セキュリティパラメータ κ に対する Π の実世界での実行を $\mathrm{REAL}_{\Pi,\mathcal{A}(z),c}(x_1, x_2, \kappa)$ とし，Π の実行による正直なパーティと攻撃者 $\mathcal{A}$ の出力ペアとして定義する．

これらを用いて能動的安全性は以下のように定義される．

定義 1 (能動的安全性)．理想的な機能 $f = (f_1, f_2)$ および任意の非一様な確率的多項式時間アルゴリズムで動作する攻撃者 $\mathcal{A}$ に対して，次式を満たす非一様な確率的多項式時間アルゴリズム Sim が存在する場合，プロトコル Π は能動的攻撃者の存在下で機能 f を安全に計算する．

$$\{\mathrm{IDEAL}_{f,\mathrm{Sim}(z),c}(x_1,x_2,\kappa)\}_{x_1,x_2,z,\kappa} \stackrel{\mathrm{c}}{\equiv} \{(\mathrm{REAL}_{\Pi,A(z),c}(x_1,x_2,\kappa)\}_{x_1,x_2,z,\kappa}$$

2.3 Private Set Intersection (PSI)

PSI プロトコル [4], [7], [9], [10] は，各パーティ P_1, P_2 が持つ集合 X_1, X_2 を入力として，自身の入力を他者に必要以上に明かすことなく，P_1 が積集合 $X_1 \cap X_2$ を得ることができるプロトコルである．PSI プロトコルの理想的な機能 $\mathcal{F}_{\mathsf{psi}}$ を図 1 に示す．

2.4 デジタル署名

デジタル署名 [6] は，あるメッセージが特定の主体によって生成・承認されたことを第三者が検証できるようにする暗号技術である．具体的な定義を次に示す．

定義 2 (デジタル署名)．デジタル署名スキーム DS は 3 つの確率的多項式時間アルゴリズム $(\mathsf{DS.Gen}, \mathsf{DS.Sign}, \mathsf{DS.Vrfy})$ と関連するメッセージ空間 $\{M_\kappa\}$ から構成される：

- $\mathsf{DS.Gen}(1^\kappa) \to (\mathrm{sk}, \mathrm{vk})$: セキュリティパラメータ κ を入力にとり，署名鍵 sk と検証鍵 vk を出力する．
- $\mathsf{DS.Sign}(m, \mathrm{sk}) \to \sigma$: メッセージ $m \in M_\kappa$，署名鍵 sk を入力にとり，署名 σ を出力する．$m \notin M_\kappa$ の場合，アルゴリズムは $\perp$ を出力する．
- $\mathsf{DS.Vrfy}(m, \sigma, \mathrm{vk}) \to b$: メッセージ $m \in M_\kappa$，署名 σ，検証鍵 vk を入力にとり，ビット $b \in \{0,1\}$ を出力する．$b = 1$ は受理，$b = 0$ は拒否を意味する．$m \notin M_\kappa$ の場合，アルゴリズムは $b = 0$ を出力する．

デジタル署名スキームはある署名鍵で生成された署名が対応する検証鍵で検証すると必ず 1 を出力するとき，正当性を満たすという．また，署名が偽造不可能性（EUF-CMA 安全性）を満たすとは，任意のメッセージに対して他のどのような署名を見ても，これまでに署名がされていないメッセージの正当な署名は作れないことを指す．本稿で扱う署名は全て EUF-CMA 安全性を満たすものとする．デジタル署名スキームには，署名鍵とメッセージから一意に署名が定まるような方式（決定的署名）と，署名が確率的に定まるような方式（確率的署名）が存在する．

2.5 Oblivious Key-Value Stores

Oblivious Key-Values Stores (OKVS) は PSI プロトコルのプリミティブとして提案 [3] され，一連の研究 [1], [8] によって改良された．以下に OKVS の定義を示す．

定義 3 (Oblivious Key-Value Stores)．Oblivious Key-Value Stores スキームは，統計量的セキュリティパラメータ λ，計算量的セキュリティパラメータ κ，乱数空間 $\{0,1\}^\kappa$，鍵空間 $\mathcal{K}$，値空間 $\mathcal{V}$，入力長 t，出力長 m に関するランダム化アルゴリズム $\mathsf{Encode}_r, \mathsf{Decode}_r$ のペアである:

- $\mathsf{Encode}_r(\{(k_1, v_1), (k_2, v_2), \ldots, (k_t, v_t)\}) \to Q$: 異なる鍵に対する t 個の鍵と値のペアと乱数 r を持つエンコードアルゴリズムはエンコーディング $Q \in \mathcal{V}^m \cup \perp$ を出力する．
- $\mathsf{Decode}_r(Q, k) \to v$: $\mathcal{V}^m$ からのエンコーディングと鍵 $k \in \mathcal{K}$ を入力すると，値 v を出力する．

OKVS はエンコーディング Q に対して，Q が生成されるときに用いられた鍵と値のペアの集合に属す鍵 k でデコードしたときに対応する v を $1 - \mathsf{negl}(\lambda)$ で出力するとき，正当性を満たすという．さらに，エンコーディング Q を見てどの鍵と値のペアがエンコードされたものかを計算量的に識別できないとき，忘却性を満たすという．

2.6 共通参照文字列

共通参照文字列（Common Reference String, CRS）とは，各パーティが事前に共有する固定の公開文字列のことを指す．より正式には，プロトコルが開始される前にセキュリティパラメータ κ に対して確率的アルゴリズム $\mathsf{CRS.Setup}(1^\kappa) \to (\mathsf{crs}, \mathsf{td})$ が 1 度だけ実行される．ここで，crs はプロトコル内で全てのパーティが参照可能である文字列であり，一方で td はシミュレータ専用のトラップドアを表す．

3. 能動的安全性の一般化

本節では PSI における既存の能動的安全性を一般化した定義を導入する．既存の定義における理想世界では，各パーティは自身の集合を TTP に送信し，TTP がそれらの入力を用いて積集合を計算することで実行される．その際，攻撃者は自身の集合を任意の値に変更することが許されており，TTP はその変更された集合をいわゆる正しい入力値として受け入れる．我々はこの「TTP が受け入れる（正しいと見なす）入力値」の観点で既存定義を一般化する．具体的には，まず TTP が受け入れる入力値を各パーティそれぞれでクラス $\mathcal{X}_1, \mathcal{X}_2$ として定義する．TTP は攻撃者から送られる入力値が，入力クラス $\mathcal{X}_1, \mathcal{X}_2$ に属す場合のみ積集合を計算する．すなわち，理想世界での実行における入力の送信で，TTP は P_1, P_2 それぞれから入力値 x_1, x_2 を受け取った場合，$x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2$ であるかを確認し，双方の入力値がそれぞれの入力クラスに属す場合はそのまま次のステップに進み，どちらか一方でもそれぞれの入力クラスに属さない場合はその時点で全てのパーティに abort を送信する．$c \in \{1, 2\}$ を攻撃者 $\mathcal{A}$ によって支配されるパーティのインデックスとする．このように定義された理想世界において，正直なパーティの出力と攻撃者 $\mathcal{A}$ の出力ペアを $\mathrm{IDEAL}^{\mathcal{X}_1,\mathcal{X}_2}_{f,A(z),c}(X_1, X_2)$ と記述する．現実世界での実行は既存の能動的安全性と同様に，入力 (x_1, x_2)，攻撃者 $\mathcal{A}$ への補助入力 z，セキュリティパラメータ κ に対する Π の実世界での実行を $\mathrm{REAL}_{\Pi,\mathcal{A}(z),c}(x_1, x_2, \kappa)$ とし，Π の実行による正直なパーティと攻撃者 $\mathcal{A}$ の出力ペアとして定義する．

定義 4 ($(\mathcal{X}_1, \mathcal{X}_2)$-能動的安全性)**.** 入力クラス $(\mathcal{X}_1, \mathcal{X}_2)$，理想的な機能 $f = (f_1, f_2)$ および任意の非一様な確率的多項式時間アルゴリズムで動作する攻撃者 $\mathcal{A}$ に対して，次式を満たす非一様な確率的多項式時間アルゴリズム Sim が存在する場合，プロトコル Π は入力クラスが $(\mathcal{X}_1, \mathcal{X}_2)$ である能動的攻撃者の存在下で機能 f を安全に計算する．

$$\{\mathrm{IDEAL}^{\mathcal{X}_1,\mathcal{X}_2}_{f,\mathrm{Sim}(z),c}(x_1,x_2,\lambda)\}_{x_1,x_2,z,\lambda} \overset{c}{\equiv} \{(\mathrm{REAL}_{\Pi,A(z),c}(x_1,x_2,\lambda)\}_{x_1,x_2,z,\lambda}.$$

なお，定義 4 は入力クラス $\mathcal{X}_1, \mathcal{X}_2$ として任意の入力を許容するようなクラスを選ぶと既存の能動的安全性の定義に一致する，既存定義の一般化であることを強調しておく．

4. 一般化能動的安全性を満たす PSI のためのフレームワーク

本節では，前節で定義した $(\mathcal{X}_1, \mathcal{X}_2)$-能動的安全性を満たす PSI を実現するためのフレームワークを提案する．

4.1 集合に対する入力クラス

PSI では入力は集合 $X_i \subseteq \mathcal{U}$（$\mathcal{U}$ は普遍集合）である．そのため，入力クラスの定義には次の二系統が考えられる．

i. 集合レベルの入力クラス：この系統では入力集合全体に対して制約を課す．例えば「サイズが a 未満の集合」や「要素の合計値が b の集合」などがあるこの場合，入力クラスは $\mathcal{X}_i \in 2^{\mathcal{U}}$ と表現される．

ii. 要素レベルの入力クラス：この系統では $\mathcal{U}$ の各要素に対して制約を課す．例えば「偶数の値」や「a から始まる値」など，要素ごとに制約を課すような入力クラスがこれに該当する．入力クラスは述語 $\varphi_i : \mathcal{U} \to \{0, 1\}$ とそれによって定義される集合 $\mathcal{U}_i = \{x \mid x \in \mathcal{U}, \varphi_i(x) = 1\}$ を用いて，$\mathcal{X}_i \in 2^{\mathcal{U}_i}$ と表現される．

本稿では ii. の要素レベルの入力クラスを想定する．

4.2 入力クラスの具体化

前節で定義したような入力クラスという概念は非常に抽象的であり，これをそのままプロトコルで扱うことは難しい．そこでまず，各パーティ P_i の入力がそれぞれの入力クラス $\mathcal{X}_i$ に属すという事実を検証できる「証拠」を用意することで，抽象的な入力クラスという概念を具体化する．本稿ではこの具体化のためにデジタル署名技術を採用する．すなわち，各パーティの入力がそれぞれの入力クラスに属している時のみ，その入力に対して署名（証拠）を与え，入力クラスに属す/属さないを署名検証という機械的な手順で判定できるようにする．

具体的にはあらかじめ署名鍵集合 SK とそれに対応する検証鍵集合 VK を定め，$\mathrm{sk} \in SK$ で生成された（すなわち $\mathrm{vk} \in VK$ で検証したときに 1 を返す）署名を伴う入力を入力クラスに属すものと判定する．本稿ではさらに、各入力 x ごとに対応する署名鍵 $\mathrm{sk} \in SK$ が一意に定まっている（同じ署名鍵が複数の入力に対応することは許されているが，一つの入力に複数の署名鍵が対応することはない）ことを仮定する．

4.3 構成フレームワーク

提案するフレームワークはとてもシンプルであり，以下の二つのフェーズで構成される．

- **認証フェーズ**: 各パーティの入力が入力クラスに属すという証拠として署名を与え，入力クラスを具体化す

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i，入力クラス $\mathcal{X}_1, \mathcal{X}_2$，デジタル署名スキーム DS 検証鍵数 t.
入力: P_i は集合 $X_i = \{x_{i,1}, \ldots, x_{i,m_i}\}$ を入力する.
出力: $\mathsf{DS.Gen}(1^\kappa) \to (\mathrm{sk}, \mathrm{vk})$ を t 回実行し，署名鍵の集合を SK，検証鍵の集合を VK を得る．ここで SK, VK は順序対とし，$\forall j \in [t]$ に対して $\mathrm{sk}_j \in SK$ と $\mathrm{vk}_j \in VK$ は対応する鍵組とする．VK を公開し，辞書 store を初期化する．各パーティ P_i（$i \in \{1,2\}$）からの入力 X_i の各要素 $x_{i,k}$（$\forall k \in [m]$）に対して，入力クラス $\mathcal{X}_i$ に紐づく φ_i を用いて $\varphi_i(x_{i,k}) \to b$ を実行する．$b = 1$ ならば辞書 $\mathsf{store}[x_{i,k}]$ を参照し，何も登録されていなければランダムに $\mathrm{sk}_j \in SK$ を選び，$\mathsf{store}[x_{i,k}]$ に $(\mathrm{sk}_j, \mathrm{vk}_j)$ を登録する．辞書 $\mathsf{store}[x_{i,k}]$ から署名鍵 $\mathrm{sk}_{i,k}$ と検証鍵 $\mathrm{vk}_{i,k}$ を取得し，$\mathsf{DS.Sign}(x_{i,k}, \mathrm{sk}_{i,k}) \to \sigma_{i,k}$ を実行する．$\varphi_i(x_{i,k}) \to 1$ を満たさない場合は $\sigma_{i,k}, \mathrm{sk}_{i,k}, \mathrm{vk}_{i,k}$ は全て $\perp$ とする．P_i に $\{(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})\}_{k\in[m]}$ を返す.

図 2 認証の理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$
Fig. 2 Ideal Functionality of Certification $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i，デジタル署名スキーム DS，検証鍵集合 VK.
入力: P_i は集合 $X_i^{\mathsf{cert}} = \{(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})\}_{k\in[m_i]}$ を入力する.
出力: パーティ P_1 は以下の集合を得る

$$\left\{ x \;\middle|\; \begin{array}{c} \forall i \in \{1,2\}, \exists \mathrm{vk} \in VK, \\ (x, \sigma_i, \mathrm{vk}) \in X_i^{\mathsf{cert}} \wedge \mathsf{DS.Vrfy}(x, \sigma_i, \mathrm{vk}) = 1 \end{array} \right\}$$

図 3 CPSI の理想機能 $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$
Fig. 3 Ideal Functionality of CPSI $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$

るフェーズ．なお，前述したようにここでは各要素ごとに対応する署名鍵を一意に定める．このフェーズを実現する理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ を図 2 に示す.

- **計算フェーズ**: 認証フェーズで署名が与えられた入力の範囲内で積集合を計算するフェーズ．署名が与えられていない入力に対しては計算結果を出力しない．このフェーズで実現される機能を本稿では特別に認証機能付き PSI（Certified-input PSI: CPSI）と呼ぶ．これを実現する理想機能 $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ は図 3 に示す.

このフレームワークに従い，理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ および $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を用いて構成した PSI プロトコル Π_{psi} を図 4 に示す.

定理 1. 図 4 のように構成したプロトコル Π_{psi} は入力クラスが $\mathcal{X}_1, \mathcal{X}_2$ である能動的攻撃者の存在下で $\mathcal{F}_{\mathsf{psi}}$ を安全に実現する.

証明. 正当性を示すため，まず両パーティがプロトコルに従って行動することを仮定する．理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ の仕様より，集合 X_i（$i \in \{1,2\}$）の各要素 $x_{i,k} \in X_i$ に対して，

$$\mathsf{DS.Vrfy}(x_i, \sigma_i, \mathrm{vk}_i) = 1 \wedge \mathrm{vk}_i \in VK \Leftrightarrow \varphi_i(x) = 1$$

が成り立つ．よって図 4 の $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ は

$$\{x \in X_1 \cap X_2 \mid \varphi_1(x) = 1 \wedge \varphi_2(x) = 1\}$$

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i，署名スキーム DS.
入力: P_i は集合 $X_i = \{x_{i,1}, \ldots, x_{i,m}\}$ を入力する.
プロトコル:

1. $\forall i \in \{1,2\}$ に対して P_i は集合 X_i を入力として理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ のインスタンスを呼び出し，各要素 $x_{i,k}$ に対応する署名 $\sigma_{i,k}$ および検証鍵 $\mathrm{vk}_{i,k}$ の三組の集合 $X_i^{\mathsf{cert}} = \{(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})\}_{k\in[m_i]}$ を得る．さらに，両パーティは公開情報として検証鍵集合 VK を得る.
2. $\forall i \in \{1,2\}$ に対して P_i は集合 X_i^{cert} を入力として検証鍵集合 VK のもとで理想機能 $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ のインスタンスを呼び出し，P_1 は以下の集合を得る.

$$\left\{ x \;\middle|\; \begin{array}{c} \forall i \in \{1,2\}, \exists \mathrm{vk} \in VK, \\ (x, \sigma_i, \mathrm{vk}) \in X_i^{\mathsf{cert}} \wedge \mathsf{DS.Vrfy}(x, \sigma_i, \mathrm{vk}) = 1 \end{array} \right\}$$

図 4 PSI フレームワーク Π_{psi}
Fig. 4 PSI framework Π_{psi}

を返す．これは PSI の理想機能 $\mathcal{F}_{\mathsf{psi}}$ を各パーティの入力集合を入力クラスに属すように整形してから与えたものと同値であり，正当性が従う.

次に図 4 のプロトコル Π_{psi} が $(\mathcal{X}_1, \mathcal{X}_2)$-能動的安全性を満たすことを示す．攻撃者 $\mathcal{A}$ が片側のパーティ（ここでは P_1 とする）を支配していると仮定する．プロトコル Π_{psi} は $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ と $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を理想機能として呼び出す構成であるため，理想世界においてシミュレータ Sim は攻撃者 $\mathcal{A}$（P_1）の入力を観測できる．具体的には，シミュレータ Sim はステップ 1 の $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ の実行において，P_1 の入力集合 X_1 を，ステップ 2 の $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ の実行において，P_1 が入力した X_1^{cert} を観測することができる．Sim はこれらの情報から P_1 の実効入力 X_1^{in} を以下のように抽出する.

$$X_1^{\mathsf{in}} := \left\{ x \in X_1 \;\middle|\; \begin{array}{c} (x, \sigma, \mathrm{vk}) \in X_1^{\mathsf{cert}} \wedge \mathrm{vk} \in VK \\ \wedge \mathsf{DS.Vrfy}(x, \sigma, \mathrm{vk}) = 1 \end{array} \right\}$$

そして抽出した X_1^{in} を TTP に送信し，TTP は正直なパーティ P_2 の入力 X_2 と X_1^{in} の積集合 $X_2 \cap X_1^{\mathsf{in}}$ を出力する.

一方で，現実世界の実行において，P_1 は $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ で得ていない任意の値を入力することが可能であるが，署名スキーム DS の EUF-CMA 安全性より，（任意の x に対する vk の値を知っていたとしても）検証に通るような署名を生成する確率は無視可能である．よって，両世界の出力の分布は計算量的に識別できない.

P_2 が攻撃者の場合も同様に成立する．以上より，プロトコル Π_{psi} は $(\mathcal{X}_1, \mathcal{X}_2)$-能動的安全性を満たす. □

本節では，入力クラス $\mathcal{X}$ を要素レベルの述語 φ で固定し，その上で「署名が検証に通る要素だけが計算に寄与する」と言うフレームワークを理想機能を用いて実現した．特定の入力クラスに限定した場合，ここで導入した $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ の実装は既存の公開鍵基盤の範囲に収まる（ただし署名者が信頼できることを仮定）.

素だけを取り込み，残りを計算上無視するかが本質となる．そこで以降の章では，$\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ の具体的な実現には踏み込まず，本節で定めたインターフェースを前提として $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を実現するプロトコルの構成に焦点を当てる．すなわち，φ に整合する（正当な要素に対してのみ署名が得られるという）最小限の性質を満たす $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ が与えられれば，後段の CPSI がどのように署名付き要素だけを取り込み，残りを計算上無視するかが本質となる．そこで以降の章では，$\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ の具体的な実現には踏み込まず，本節で定めたインターフェースを前提として $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を実現するプロトコルの構成に焦点を当てる．

5. 決定的署名スキームに基づく CPSI

本節では決定的署名スキームに対する CPSI プロトコルを構成する．各パーティが保持する署名は前節で定義した理想機能 $\mathcal{F}_{\mathsf{cert}}^{\mathcal{X}_1,\mathcal{X}_2,\mathsf{DS}}$ に従って生成されているとする．すなわち，各要素 x に署名鍵 sk（と検証鍵 vk）が一意対応しているとする．

本節での目的は，各要素が「正当な署名」を伴うときにのみその要素が積集合の計算に寄与するような PSI プロトコル（CPSI）を，通常の PSI への簡潔な還元として実現することである．その鍵となるのは，両者が要素 x に対して「正当な署名」を保持する時のみ，共通のラベルを計算できるようにすることである．このようなラベルが計算できれば，あとはこのラベルを用いて通常の PSI を実行するだけで CPSI を実現することができる．

この方針は決定的署名と非常に相性が良い．署名 σ は正当な値を持つ時のみ得られる値であり，正当な値を持たないパーティは署名の EUF-CMA 安全性より，検証に通るような署名を作成することができない．つまり，署名は正当な要素を持つ場合にのみ扱える共有可能な秘密情報であり，決定的署名では一位に定まるため，さらに簡潔に扱える．具体的には要素 x に対するラベル τ をランダムオラクル H を用いて $\tau := H(x\|\sigma\|\mathrm{vk})$ と計算する．これにより，両者が同じ x に対して正当に認証できるなら，τ は必ず一致し，逆に署名がない（あるいは別の要素である）場合に τ が一致するにはランダムオラクルの衝突が必要となる．従って，ラベルの一致は「両者が x を正当に保持している」ことの同値条件として機能する．

最後に，本プロトコルでは鍵や署名の具体値を直接比較する必要がなく，外部に露出するのはラベル τ と PSI の結果だけである．これにより，情報漏洩を抑えつつ，計算量はランダムオラクル呼び出しと PSI 一回に集約される．この方針に基づき，構成したプロトコル $\Pi_{\mathsf{cpsi}}^{\mathsf{dDS}}$ を図 5 に示す．

定理 2. 図 5 のように構成したプロトコル $\Pi_{\mathsf{cpsi}}^{\mathsf{dDS}}$ は能動的な攻撃者の存在下で安全に $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を実現する．

証明． 各パーティ P_i は自身の集合 X_i^{cert} の各要素のうち，

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i，決定的認証スキーム dDS，検証鍵集合 VK，ランダムオラクル H

入力: P_i は $X_i^{\mathsf{cert}} = \{(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})\}_{k\in[m_i]}$ を入力する．

プロトコル:

1. P_i（$i \in \{1,2\}$）は以下のように X_i^{label} を計算する．

$$X_i^{\mathsf{label}} = \left\{ (x,\tau) \;\middle|\; \begin{array}{l} (x,\sigma,\mathrm{vk}) \in X_i^{\mathsf{cert}} \wedge \tau = H(x\|\sigma\|\mathrm{vk}) \\ \wedge \mathrm{vk} \in VK \wedge \mathsf{dDS.Vrfy}(x,\sigma,\mathrm{vk}) = 1 \end{array} \right\}$$

ここで，$T_i := \{\tau \mid (x,\tau) \in X_i^{\mathsf{label}}\}$ とする．

2. P_1, P_2 は T_1, T_2 を入力として $\mathcal{F}_{\mathsf{psi}}$ のインスタンスを呼び出し，P_1 は $T_1 \cap T_2$ を得る．
3. P_1 は $X^{\mathsf{out}} = \{x \mid (x,\tau) \in X_i^{\mathsf{label}} \wedge \tau \in T_1 \cap T_2\}$ を出力する．

図 5 決定的認証スキームに基づく CPSI プロトコル $\Pi_{\mathsf{cpsi}}^{\mathsf{dDS}}$
Fig. 5 CPSI Protocol Based on Deterministic Signature $\Pi_{\mathsf{cpsi}}^{\mathsf{dDS}}$

$\mathrm{vk}_{i,k} \in VK \wedge \mathsf{dDS.Vrfy}(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k}) = 1$ であるような $(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})$ を用いてラベル $\tau_{i,k} = H(x_{i,k}\|\sigma_{i,k}\|\mathrm{vk}_{i,k})$ を決定的に計算し，ラベル集合 $T_i = \{\tau_{i,k}\}_{k\in[m_i']}$ を PSI の理想機能 $\mathcal{F}_{\mathsf{psi}}$ に入力する．ここで仮定より，同一要素には同一の署名鍵が用いられ，署名は決定的であることから，両者が同じ x を保持していれば，両者が計算する τ は一致する．他方で，ランダムオラクルの衝突困難性により，$(x,\sigma,\mathrm{vk}) \mapsto H(x\|\sigma\|\mathrm{vk})$ は高確率で単射である．従って，確率 $1 - \mathsf{negl}(\kappa)$ で

$$T_1 \cap T_2 = \{\tau \mid (x,\sigma,\mathrm{vk}) \in X_1^{\mathsf{cert}} \cap X_2^{\mathsf{cert}} \wedge \tau = H(x,\sigma,\mathrm{vk})\}$$

が成り立つ．図 5 のステップ 3 では，P_1 が保持する X_1^{label} から，ラベルを一意に要素へ逆写像するため，

$$X^{\mathsf{out}} = \left\{ x \;\middle|\; \begin{array}{c} \forall i \in \{1,2\}, \exists \mathrm{vk} \in VK, \\ (x,\sigma,\mathrm{vk}) \in X_i^{\mathsf{cert}} \wedge \mathsf{DS.Vrfy}(x,\sigma_i,\mathrm{vk}) = 1 \end{array} \right\}$$

が結論される．ゆえに図 5 のプロトコルの出力は $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ の出力と一致する．例外はランダムオラクルの衝突のみであり，これは無視可能である．

次に，プロトコルが能動的安全性を満たすことを示す．

攻撃者 $\mathcal{A}$ が片側のパーティ（ここでは P_1）を支配していると仮定する．シミュレータは $\mathcal{A}$ のすべてのランダムオラクルクエリをリスト L に記録する．$\mathcal{A}$ が PSI の理想機能に送信するラベル集合 T_1 を取得する．リスト L およびラベル集合 T_1 から以下の実効入力 X_1^{in} を抽出する．

$$X_1^{\mathsf{in}} := \{x \mid (x\|\sigma\|\mathrm{vk},\tau) \in L \wedge \tau \in T_1 \wedge \mathsf{Vrfy}(x,\sigma,\mathrm{vk}) = 1\}$$

X_1^{in} を P_1 の入力として TTP に送信する．この X_1^{in} は現実世界で P_1 が入力したラベルの「正当な前像」に対応する．

P_1 が PSI に入力したラベルが P_2 のラベルと一致するためには，通常，事前にランダムオラクルに問い合わせを行なっていなければならず，また別要素・別署名から同一タグが生じる確率も無視可能である．従って，シミュレータ

が返す PSI 応答の分布は現実世界のそれと一致し，両世界は識別できない．

プロトコルの対称性により P_2 が $\mathcal{A}$ の支配下にある場合でも同様である．以上より，プロトコル $\Pi_{\mathsf{cpsi}}^{\mathsf{dDS}}$ は能動的な攻撃者の存在下で安全に $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を実現する． □

6. Schnorr 署名スキームに基づく CPSI

5 節では決定的署名を前提に，同一要素に対して同一値となる署名の再現性を用い，ランダムオラクルを用いてラベル $\tau = H(x\|\sigma\|\mathrm{vk})$ を計算し，それを PSI の入力とした．これに対し本節では，EUF-CMA 安全性を満たす確率的署名である Schnorr 署名を前提に，署名生成時に用いられる乱数に依存しない「共通ラベル」を暗号的に導出してから PSI に渡す．

直感的には，(i) 各要素 x に対する Schnorr 署名 $\sigma = (s, r)$ を ElGamal 暗号で秘匿しつつ OKVS に格納し，(ii) 相手が同じ (x, vk) を持つ場合にのみ OKVS から対応する暗号文を取り出せるようにキーを $H(x\|\mathrm{vk})$ で合わせ，(iii) 互いの署名暗号文を用いた代数的な乱数消去計算で署名生成および暗号文生成に用いられた乱数の影響を打ち消して，同じ x に対して両者が一致するラベル τ を再現する，という分割設計である．このとき，署名の検証条件を指数の世界に持ち込み，両者が正当署名を持つ場合だけラベルが一致するよう設計する．こうして署名が確率的でも同じラベルが得られ，PSI の入力として扱える．

同じラベルを得るために Schnorr 署名の検証式の線形性を利用する．メッセージ M に対する Schnorr 署名 (s, r) が $e = H(r\|M)$, $s = ex_{\mathsf{sch}} + k$ (k は乱数) という関係式を満たすという事実から, 同じメッセージ M, 同じ署名鍵 x_{sch} で生成された二つの署名 $(s_1, r_1), (s_2, r_2)$（および生成時に用いられる乱数 k_1, k_2）は $e_1 e_2 x_{\mathsf{sch}} = e_2(s_1 - k_1) = e_1(s_2 - k_2)$ を満たす．すなわち，

$$\frac{g^{s_1 c_2}}{{r_1}^{c_2}} = \frac{g^{s_2 c_1}}{{r_2}^{c_1}} \tag{1}$$

が成り立つ．本節ではこの性質を利用し，同じ要素 x に対して確率的に定まる異なる署名から共通のラベル τ を生成する．本構成では Schnorr 署名において s や $e = H(r\|M)$ の値は 0 でない値を仮定（実際このような確率は無視可能であるか，再署名されると考える）し，Schnorr 署名および ElGamal 暗号で用いられる群や生成元は同一とし，$\mathbb{G} = \mathbb{Z}_p^\times$, $g \in \mathbb{Z}_p^\times$ とする．

定理 3. 図 6 のように構成したプロトコル $\Pi_{\mathsf{psi}}^{n,\mathsf{SchDS}}$ は能動的な攻撃者の存在下で安全に $\mathcal{F}_{\mathsf{cpsi}}^{\mathsf{DS}}$ を実現する．

証明. まず正当性について示す．簡単化のため，集合要素のインデックスを表す k は省略する．プロトコルのステップ 3 において，同じ x, vk に対して P_i（$i \in \{1,2\}$）が P_j（$j \in \{1,2\}, j \neq i$）から得る Q_j のデコードでは，ステップ 2 で P_j がエンコードで格納した三組

$$\hat{q}_i^{(0)} = g^{d_j}, \qquad \hat{q}_i^{(1)} = s_j \cdot \mathrm{vk}_{\mathsf{E}}^{d_j}, \qquad \hat{q}_i^{(2)} = {r_j}^{\mathrm{pk}_{\mathsf{E}}^{d_j}}$$

を得る．これは OKVS の正当性から直ちに従う．ステップ 4 で P_i が計算する $w_i^{(0)}, w_i^{(1)}$ は上記の $\hat{q}_i^{(0)}, \hat{q}_i^{(1)}, \hat{q}_i^{(2)}$ から計算され，それぞれ展開・整理すると，

$$w_i^{(0)} = c_j \cdot s_j \cdot \left(\frac{\mathrm{pk}_{\mathsf{E}}}{g}\right)^{d_i + d_j}, \quad w_i^{(1)} = {r_j}^{(\mathrm{pk}_{\mathsf{E}}/g)^{d_i + d_j}}$$

となる．ここまでで，両者のローカル乱数 d_i, d_j はいずれも $(\mathrm{pk}_{\mathsf{E}}/g)^{d_i+d_j}$ という「共通の指数係数」にまとめ上げら

パラメータ: パーティ P_i（$i \in \{1,2\}$）の入力集合サイズ m_i,, Schnorr 署名スキーム SchDS，検証鍵集合 VK，ランダムオラクル H

共通参照文字列: $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{CRS.Setup}(1^\kappa)$ を実行し, $\mathsf{crs} := (\mathbb{G}, p, g, \mathrm{pk}_{\mathsf{E}})$ は公開され，$\mathsf{td} := \mathrm{sk}_{\mathsf{E}}$ はシミュレータのみが参照可能なトラップドアとして各パーティには公開されない．ここで，$\mathrm{pk}_{\mathsf{E}}, \mathrm{sk}_{\mathsf{E}}$ は ElGamal 暗号の公開鍵・秘密鍵を表しており，$\mathbb{G}$ は位数 p の巡回群，g はその生成元で，ElGamal 暗号に関連するパラメータを表す．

入力: P_i は $X_i^{\mathsf{cert}} = \{(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})\}_{k \in [m_i]}$ を入力する．ここで，$\sigma_{i,k}$ は Schnorr 署名 $(s_{i,k}, r_{i,k})$ である．

プロトコル:

1. P_i（$i \in \{1,2\}$）は集合 X_i^{cert} の各要素 $(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})$ ($k \in [m_i]$) の Schnorr 署名 $\sigma_{i,k} = (s_{i,k}, r_{i,k})$ を Elgamal 暗号の暗号化アルゴリズム Enc，乱数 $d_{i,k} \in \mathbb{Z}_{p-1}$ を用いて暗号化し，$(q_{i,k}^{(0)}, q_{i,k}^{(1)}) = \mathrm{Enc}_{\mathrm{pk}_{\mathsf{E}}}(s_{i,k}; d_{i,k})$ とする．すなわち，$q_{i,k}^{(0)} \leftarrow g^{d_{i,k}}$, $q_{i,k}^{(1)} \leftarrow s_{i,k} \cdot {\mathrm{pk}_{\mathsf{E}}}^{d_{i,k}}$ である．さらに，$e_{i,k} \leftarrow \mathrm{pk}_{\mathsf{E}}^{d_{i,k}}$, $q_{i,k}^{(2)} \leftarrow r_{i,k}^{e_{i,k}}$ を計算する．
2. P_i（$i \in \{1,2\}$）は集合 X_i^{cert} の各要素 $(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})$ ($k \in [m_i]$) に対して $H(x_{i,k} \,\|\, \mathrm{vk}_{i,k})$ を計算し，OKVS の Encode を用いて以下のように Q_i を計算する．

   $Q_i \leftarrow \mathsf{Encode}(\{(H(x_{i,k}\|\mathrm{vk}_{i,k}),\ q_{i,k}^{(0)}\|q_{i,k}^{(1)}\|q_{i,k}^{(2)})\}_{k \in [m_i]})$

   P_i は得られた Q_i を相手パーティに送信する．
3. P_i（$i \in \{1,2\}$）は受け取った Q_j（$j \in \{1,2\}, j \neq i$），集合 X_i^{cert} の各要素 $(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})$ ($k \in [m_i]$) に対して $\mathsf{Decode}(Q_j, H(x_{i,k}\|\mathrm{vk}_{i,k}))$ を計算し，$\hat{q}_{i,k}^{(0)}, \hat{q}_{i,k}^{(1)}, \hat{q}_{i,k}^{(2)}$ を得る．
4. P_i（$i \in \{1,2\}$）は集合 X_i^{cert} の各要素 $(x_{i,k}, \sigma_{i,k}, \mathrm{vk}_{i,k})$ ($k \in [m_i]$) に対して $c_{i,k} \leftarrow H(r_{i,k}\|x_{i,k})$ を計算し，$w_{i,k}^{(0)} \leftarrow \frac{\hat{q}_{i,k}^{(1)}}{\hat{q}_{i,k}^{(0)}} \cdot (\frac{\mathrm{pk}_{\mathsf{E}}}{g})^{d_{i,k}} \cdot c_{i,k}$ および $w_{i,k}^{(1)} \leftarrow (\hat{q}_{i,k}^{(2)})^{\frac{1}{\hat{q}_{i,k}^{(0)}} \cdot (\frac{\mathrm{pk}_{\mathsf{E}}}{g})^{d_{i,k}}}$ を求め，$X_i^{\mathsf{label}} = \{(x_{i,k}, \tau_{i,k})\}_{k \in [m_i]}$ を以下のように計算する．

   $$X_i^{\mathsf{label}} \leftarrow \left\{ \left((x_{i,k}, \frac{w_{i,k}^{(1)}}{g^{w_{i,k}^{(0)}}}) \right) \right\}_{k \in [m_i]}$$

   ここで，$T_i := \{\tau \mid (x, \tau) \in X_i^{\mathsf{label}}\}$ とする．
5. P_1, P_2 は T_1, T_2 を入力として $\mathcal{F}_{\mathsf{psi}}$ のインスタンスを呼び出し，P_1 は $T_1 \cap T_2$ を得る．
6. P_1 は $X^{\mathsf{out}} = \{x \mid (x, \tau) \in X_i^{\mathsf{label}} \wedge \tau \in T_1 \cap T_2\}$ を出力する．

図 6 Schnorr 署名に基づく CPSI プロトコル $\Pi_{\mathsf{cpsi}}^{\mathsf{SchDS}}$
Fig. 6 PSI Protocol Based on Schnorr Signature $\Pi_{\mathsf{cpsi}}^{\mathsf{SchDS}}$

れる．よって，$\alpha \coloneqq (\mathrm{pk}_{\mathsf{E}}/g)^{d_i+d_j}$ とする．これらの値をステップ 4 のラベル τ_i を計算する式に代入すると，

$$\tau_i = \frac{{r_j}^{(\mathrm{pk}_{\mathsf{E}}/g)^{d_i+d_j}}}{g^{c_j \cdot s_j \cdot \alpha}} = \left(\frac{{r_j}^{c_i}}{g^{s_j \cdot c_i}}\right)^{\alpha}$$

となる．ここで，式 (1) と α が両者で共通の値という事実から，$\tau_i = \tau_j$ が成り立つ．よって，両者が同じ x について正当な署名を持つときラベルが一致する．一方で，どちらか一方で検証に通らない署名の場合，式 1 の関係式が満たされないため，一致する確率は OKVS のデコードで偶然一致する確率に等しく，これは無視可能である．

次に構成したプロトコル $\Pi^{\mathsf{SchDS}}_{\mathsf{cpsi}}$ が能動的安全性を満たすことを示す．まず，P_1 が攻撃者 $\mathcal{A}$ の支配下にある場合を考える．シミュレータは内部で P_1 とプロトコルを実行し，ステップ 2 およびステップ 4 で P_1 のランダムオラクルクエリ $H(x\|\mathrm{vk}), H(r\|x)$，ステップ 2 で P_1 から送信される Q_1 およびステップ 4 で呼び出される PSI の理想機能 $\mathcal{F}_{\mathsf{psi}}$ への入力 T_1 を観測し，これらの情報から P_1 の実際の入力 X_1^{in} を抽出する．具体的には，シミュレータは，まずステップ 2 のランダムオラクルクエリの履歴を参照し，x に対応する検証鍵 vk を取り出す．次に Q_i を $H(x\|\mathrm{vk})$ でデコードし，$q_1^{(0)}, q_1^{(1)}, q_1^{(2)}$ を得る．ここで得た $q_1^{(0)}, q_1^{(1)}, q_1^{(2)}$ を用いて，ステップ 4 の計算と同様に τ を計算し，$\tau \in T_1$ を満たす場合のみ，次に進む．ElGamal 暗号文 $(q_1^{(0)}, q_1^{(1)})$ を，CRS トラップドア $\mathsf{td} = \mathrm{sk}_{\mathsf{E}}$ を用いて復号し，s' を得る．さらに，$g^{dx} = (q_1^{(0)})^{\mathrm{sk}}$ から，$r' = (q_1^{(2)})^{1/g^{dx}}$ を計算し，ステップ 4 のランダムオラクルクエリ $H(r\|x)$ と整合するかをチェックする．整合する場合のみ，ステップ 2 のランダムオラクルクエリの観測で得た対応する x, vk を用いて $\mathsf{SchDS.Vrfy}(x, (s', r'), \mathrm{vk})$ を実行する．そして，検証に通った場合のみ X_1^{in} に $(x, (s', r'), \mathrm{vk})$ を追加し，全ての $x \in X_1$ に対して処理を終えた後，X_1^{in} を TTP に送信する．

P_1 が $q_1^{(0)}, q_1^{(1)}, q_1^{(2)}$ の計算において，共通の乱数 d を用いていない場合，シミュレータは CRS トラップドア td を用いても正しい署名を抽出することはできないが，その場合，現実世界においても攻撃者は $\hat{\tau}_2$ と τ_1 の値は一様乱数と識別できない．正当な署名を伴わない要素が計算結果として出力されるのはランダムオラクルの出力が衝突する場合と OKVS でエンコードしていない点が偶然正当な署名を所持している時に計算される値と一致する場合のみであり，これらの事象が生じる確率は無視可能である．

プロトコルの対称性により P_2 が $\mathcal{A}$ の支配下にある場合でも同じ議論がそのまま適用できる．

上記の議論より，プロトコル $\Pi^{\mathsf{SchDS}}_{\mathsf{psi}}$ は能動的な攻撃者の存在下で安全に $\mathcal{F}^{\mathsf{DS}}_{\mathsf{cpsi}}$ を実現する． □

7. まとめ

本研究は，入力集合の変更による不正な情報収集という現実的な脅威に着目し，「入力クラスの制限」という新しい観点から能動的安全性を一般化した．さらに，この一般化に対応するフレームワークとして，入力クラスに属す入力に対してのみ署名を与えることで入力クラスという抽象的な概念を具体化し，署名が与えられた要素だけを計算に寄与させる認証機能付き PSI（CPSI）を構成した．

CPSI の具体的な構成として，二種類の署名方式に基づく構成を示した．一つは決定的署名に基づく方式であり，署名の一意性を利用して効率的かつシンプルにラベルを導出できる利点を持つ．もう一つは確率的署名である Schnorr 署名に基づく方式であり，署名生成時の乱数依存性を代数的に打ち消し，共通ラベルを導出することで，より広く利用される署名方式にも対応できる柔軟性を備える．

謝辞 本研究は JSPS 科研費 JP23H00468, JP23H00479, JP23K17455, JP23K21644, JP23K24846 の助成，および JST CREST JPMJCR23M2 における AIP チャレンジプログラムの支援を受けたものです．

参考文献

[1] Bienstock, A., Patel, S., Seo, J. Y. and Yeo, K.: Near-Optimal Oblivious Key-Value Stores for Efficient PSI, PSU and Volume-Hiding Multi-Maps, *USENIX Security '23*, USENIX Association (2023).

[2] Camenisch, J. and Zaverucha, G. M.: Private Intersection of Certified Sets, *FC 2009*, Springer (2009).

[3] Garimella, G., Pinkas, B., Rosulek, M., Trieu, N. and Yanai, A.: Oblivious Key-Value Stores and Amplification for Private Set Intersection, *Advances in Cryptology - CRYPTO 2021*, Springer, pp. 395–425 (2021).

[4] Ghosh, S. and Nilges, T.: An Algebraic Approach to Maliciously Secure Private Set Intersection, *Advances in Cryptology - EUROCRYPT 2019*, Springer (2019).

[5] Gong, Z., Zheng, Z., Hu, Z., Tian, K., Zhang, Y., Zhedanov, O. and Liu, F.: Authenticated Private Set Intersection: A Merkle Tree-Based Approach for Enhancing Data Integrity, *CoRR* (2025).

[6] Katz, J.: *Digital Signatures*, Springer (2010).

[7] Pinkas, B., Rosulek, M., Trieu, N. and Yanai, A.: PSI from PaXoS: Fast, Malicious Private Set Intersection, *Advances in Cryptology - EUROCRYPT 2020*, Springer, pp. 739–767 (2020).

[8] Raghuraman, S. and Rindal, P.: Blazing Fast PSI from Improved OKVS and Subfield VOLE, *ACM CCS 2022*, ACM, pp. 2505–2517 (2022).

[9] Rindal, P. and Rosulek, M.: Improved Private Set Intersection Against Malicious Adversaries, *Advances in Cryptology - EUROCRYPT 2017*, pp. 235–259 (2017).

[10] Rosulek, M. and Trieu, N.: Compact and Malicious Private Set Intersection for Small Sets, *ACM CCS 2021*, ACM, pp. 1166–1181 (2021).

[11] Schnorr, C.: Efficient Identification and Signatures for Smart Cards, *Advances in Cryptology - CRYPTO '89*, Vol. 435, Springer, pp. 239–252 (1989).

[12] Wen, Y., Gong, Z., Huang, Z. and Qiu, W.: A new efficient authorized private set intersection protocol from Schnorr signature and its applications, *Clust. Comput.*, pp. 287–297 (2018).