プライバシ保護型データ解析手法の
計算高速化に関する研究

代表研究者　西出 隆志　筑波大学　システム情報系　准教授

1　はじめに

本研究課題では、複数の計算参加者がそれぞれ独自の入力を持つ状況において、互いに自身の入力を他の参加者に明かすことなく計算を実行し、計算結果を得る手法の改良に取り組んでいる (Fig. 1)。このような手法は、各計算参加者の入力データに対するプライバシを厳密に保護することができるため、従来は機密性の高さから共有・解析が困難であった有用データの安全かつ積極的な利活用を促進するものとして、注目を集めている。

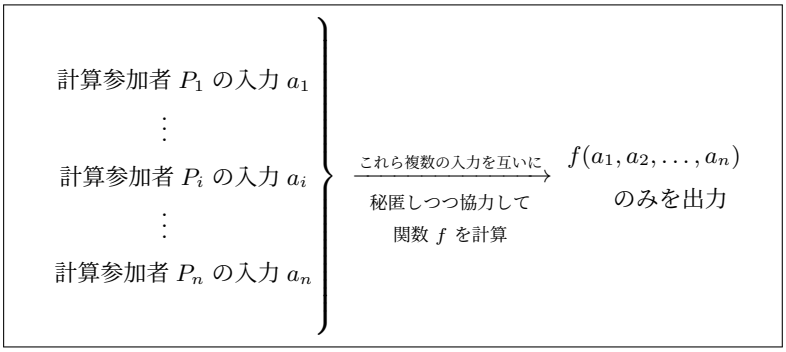


Fig. 1. プライバシ保護型データ解析の概念

さらに、一般的にデータを保護したまま処理を行うことが可能となれば、機密データとその上で行う計算処理とを、信頼できない外部環境に安全にアウトソースすることが可能となる。このような観点からも、プライバシ保護型データ解析技術は、クラウ

公益財団法人電気通信普及財団
研究調査助成報告書　第40号　2025年度

ドコンピューティングやデータ共有基盤において重要な要素技術の一つとして位置づけられている。プライバシ保護型データ解析を実現するための暗号技術的枠組みは、これまでの理論暗号研究において十分に検討され、その可能性が示されてきた。しかしながら、現実のアプリケーションにおける大規模データの処理やリアルタイム応答性といった要請に応えるには、依然として処理性能に関する課題が残されている。本研究では、これらの課題に対処するため、特に計算の高速化に焦点を当て取り組んだ。

2　プライバシ保護型データ解析手法の基盤となる技術

代表的なプライバシ保護型データ解析手法として、完全準同型暗号 (Fully Homomorphic Encryption, FHE) に基づく方法と、秘密分散 (Secret Sharing) に基づく方法がある。また完全準同型暗号と秘密分散の中間的技法として Homomorphic Secret Sharing (HSS) と呼ばれる方法も存在する。

2.1　完全準同型暗号に基づく方法

完全準同型暗号は、暗号化されたデータを復号せずに計算を実行できる技術である。完全準同型暗号は 2009 年に Craig Gentry [Gen09] によってはじめて提案された。提案当初の方式は理論的実現可能性を示すだけであったが、その後、様々な実用化に向けた効率化が今日でも活発に進められている。現状、代表的な完全準同型暗号方式として、[BGV12, FV12, GSW13, CKKS17, CGGI20] があり、これらの標準化についても作業が進められている状況である。

今、平文 m を完全準同型暗号で暗号化した暗号文を $\mathsf{Enc}(m)$ と表すと、完全準同型暗号では次のような暗号文に対する処理が可能である。

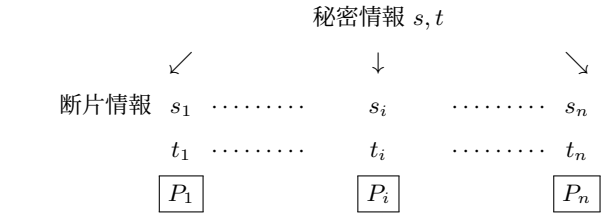
$$\mathsf{Enc}(m_1) +_{\mathsf{FHE}} \mathsf{Enc}(m_2) = \mathsf{Enc}m_1 + m_2)$$
$$\mathsf{Enc}(m_1) +_{\mathsf{FHE}} \quad m_2 \quad = \mathsf{Enc}(m_1 + m_2)\ (m_2 \text{ は平文})$$
$$\mathsf{Enc}(m_1) \times_{\mathsf{FHE}} \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 \times m_2)$$
$$\mathsf{Enc}(m_1) \times_{\mathsf{FHE}} \quad m_2 \quad = \mathsf{Enc}(m_1 \times m_2)\ (m_2 \text{ は平文})$$

このような暗号文同士の演算が可能な性質は「準同型性（homomorphism）」と呼ばれる。ここで $+_{\mathsf{FHE}}$ は暗号文に対する加算処理、$\times_{\mathsf{FHE}}$ は暗号文に対する乗算処理を表す。こ

れらの暗号文に対する処理は公開鍵に秘密鍵や元の平文を知らずに実行が可能である。この性質により、入力を暗号文のまま計算を行い、最終出力を復号することで正しい計算結果が得られる。そのため計算に対する入力の暗号文が集まれば、その入力に対応する出力の暗号文を、計算参加者間の通信無しで計算できる利点がある。しかし暗号文同士の演算の計算量は重く改良の余地がある。

2.2 秘密分散に基づく方法

秘密分散はある秘密にしたい値を複数の断片情報に分割し、断片情報からは元の値を推測することは困難だが、十分な数の断片情報を集めることで元の値を復元可能とする技術である (Fig. 2)。



1. $s = s_1 + s_2 + \cdots + s_i + \cdots + s_n$
$t = t_1 + t_2 + \cdots + t_i + \cdots + t_n$ となっている

2. 各参加者 P_i は秘密情報 s, t の断片情報 s_i, t_i のみを受け取る

Fig. 2. 秘密分散の概略

秘密分散に基づく方法では、全ての入力は複数の断片情報 (シェアと呼ばれる) に分割され、計算参加者らは各入力の断片情報のみを得て、計算を進めることで、入力を秘匿する手法である (Fig. 3)。

秘密分散に基づく方法では、完全準同型暗号に基づく方法と比べて、計算量が小さいという利点を持つが、一方で、入力を持ち寄った計算参加者らの間で通信を頻繁に行う必要があり、この通信量の多さが計算全体のボトルネックとなっている。

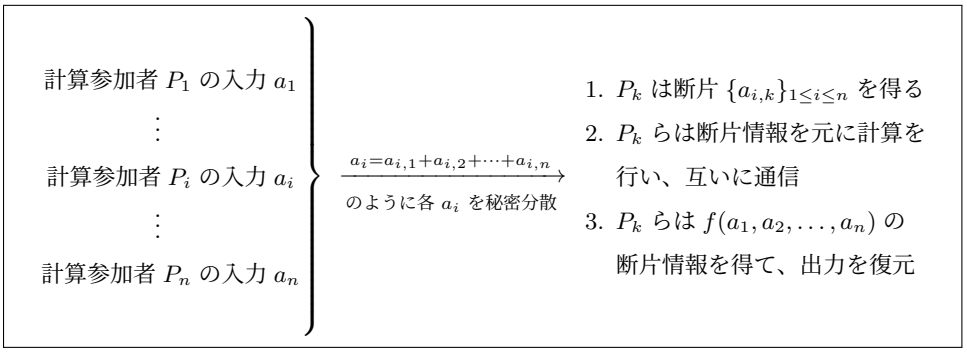


Fig. 3. 秘密分散に基づく方法の概略

2.3 Homomorphic Secret Sharing に基づく方法

Homomorphic Secret Sharing [BCG+17, BKS19, DIJL23] は、完全準同型暗号と秘密分散法の中間に位置する技術であり、近年注目されているプライバシ保護型計算の枠組みである。Homomorphic Secret Sharing では、初期入力は各計算サーバに暗号文で与えるが、途中計算結果は秘密分散のように、複数の断片に分割されており、それぞれのサーバが局所的な処理を行った後に結果の断片を結合することで、全体の計算結果を得る。Homomorphic Secret Sharing の大きな特徴は、サーバ間の通信を行うことなく、個々のサーバがシェアに対して独立に計算を実行できる点にある (Fig. 4)。これにより、秘密分散方式における通信コストの高さという課題を克服しつつ、完全準同型暗号と比べて計算負荷の軽減を実現できている。

Homomorphic Secret Sharing と秘密分散に基づく手法の違いは主に以下の点にある。

- Homomorphic Secret Sharing での初期入力は完全準同型暗号に類似した特殊な暗号文の形で計算を実行するサーバらに与えられる。
  - この暗号文は Nearly Linear Decryption と呼ばれる内積計算に近い形で復号が可能となっている。
  - またクライアントは事前にサーバらに復号用の鍵を秘密分散した形で配布しておく。これによりサーバ間の通信が不要な形で、乗算が可能となる。
- 計算の途中結果や最終結果は (暗号文ではなく) 秘密分散に類似した形でサーバ間に分散して保持される。

- 乗算は、初期入力と途中計算結果の間にのみ適用可能であり (このタイプの計算は Restricted Multiplications Straight-line (RMS) program と呼ばれる)、途中計算結果同士では加算のみが許されている (乗算は不可)。

このように秘密分散に基づく手法とは異なり上記の RMS という計算制約はあるものの、計算参加者間での通信を伴わずに計算を実行可能であるという特徴を有している。

本研究では、SIMD (Single Instruction, Multiple Data) 型の並列計算に対応した HSS 構成に着目し、複数の入力を同時に処理することで、HSS に基づくプライバシ保護型計算の検証可能性と呼ばれる性質を効率的に実現することに取り組んでいる。

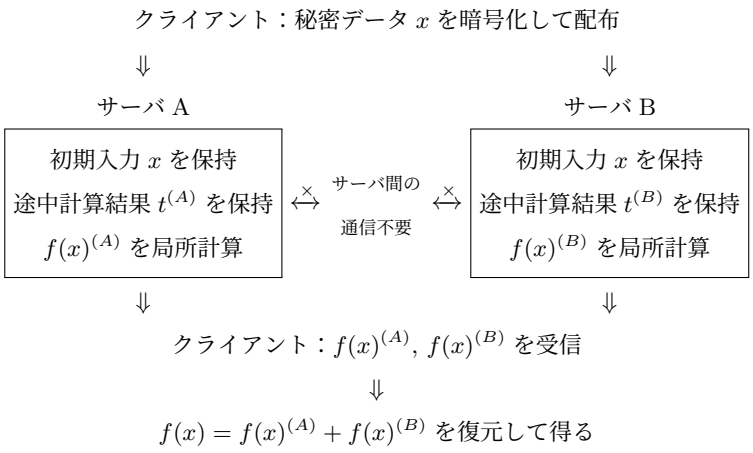


Fig. 4. Homomorphic Secret Sharing (HSS) の概念図

3　研究成果の要約

プライバシ保護型データ解析手法の高速化実現に向けて、本研究プロジェクトで取り組んで得られた成果は、基盤技術に基づき分類すると以下となる。

完全準同型暗号に関連する成果:

[MMN+24, NOFN24, NHT+24, 中林土 +24, NOF+23]

秘密分散に基づく手法に関連する成果:

[TN24, 土西 25]

HSS に基づく手法に関連する成果:

[XN24, XN25]

これらのいくつかの成果について以降説明する。

3.1 研究成果 [MMN+24]

従来の完全準同型暗号には、例えば整数データを暗号化する際に扱える平文のサイズが 16 ビット程度に制限されるという課題があった [MMN22]。本研究での成果 [MMN+24] において、2 つの 16 ビット整数平文を持つ暗号文を用いて 1 つの 32 ビットの平文を扱えるよう拡張し、この拡張した暗号文に対する大小比較やキャリー計算などの複雑な演算の計算高速化を実現した。この高速化の中では通常であれば 2 変数関数として処理しなければならない計算を、より計算量の少ない 1 変数関数の複数の組み合わせで表現できることを明らかにすることで、全体の計算の高速化を実現した。

ここでは論文で用いた 1 つの手法の概略について説明する。2 つの 16 ビット整数平文を持つ暗号文を用いて 1 つの 32 ビットの平文を表現する状況においては、例えば暗号文同士の乗算を計算する際に、乗算における桁上がり (以降、乗算キャリーと呼ぶ) の計算が必要となる。

乗算キャリーとは例えば平文の各桁 (16 ビット整数平文) の法が 11 として、入力の $8, 5$ から

$$8 \times 5 = 40 = \boxed{3} \times 11 + \underbrace{7}_{40 \bmod 11}$$

の $\boxed{3} \in \mathbb{Z}_{11}$ を求める計算に相当する。通常であれば、この場合、$8, 5$ のような 2 つの入力を取る 2 変数関数 $\mathsf{mcarry}(a, b)$ を使って計算する必要がある。しかしこれは次の手法を用いることで、1 変数関数評価の組み合わせでより効率的に計算可能なことが示せる。

[KMNN15, Theorem 2] において、$\mathsf{mcarry}(a, b)$ を計算する多項式は各桁の平文の法を t とすると、以下の形で表されることが示されている。

$$\mathsf{mcarry}(a, b) \equiv ab \cdot (\Psi(ab) - \Psi(a) - \Psi(b) + \Psi(1)) \bmod t$$

ここで $\Psi(\cdot)$ は平文の法である t のみから決まる $t-2$ 次の 1 変数多項式であり、この多項式 $\Psi(\cdot)$ の全ての係数は Bernoulli 数を使って具体的に書き下すことができる。

よって $a, b \in \mathbb{Z}_t$ の暗号文から $\mathsf{mcarry}(a,\, b) \in \mathbb{Z}_t$ の暗号文を計算する際には、1 変数多項式 $\Psi(\cdot)$ の評価のみを用いて次のように効率的に計算ができる。

1. a, b の暗号文から乗算を計算し、ab の暗号文を計算する。
2. ab, a, b のそれぞれの暗号文に $\Psi(\cdot)$ に対応する 1 変数関数評価 $\mathbb{Z}_t \to \mathbb{Z}_t$ を実行し、$\Psi(ab), \Psi(a), \Psi(b)$ の暗号文を得る。
3. $\Psi(ab), \Psi(a), \Psi(b)$ の暗号文から準同型性を用いて $\Psi(ab) - \Psi(a) - \Psi(b)$ の暗号文を得る。
4. $\Psi(ab) - \Psi(a) - \Psi(b)$ の暗号文と、ab の暗号文から準同型性を用いて乗算を計算することで、$ab \cdot (\Psi(ab) - \Psi(a) - \Psi(b) + \Psi(1))$ の暗号文を得る。

[MMN22] で示されている OneHotSlot と呼ばれる効率的な 1 変数関数評価を用いることで、$\Psi(\cdot)$ の多項式評価は、より少ないべき乗計算のみを用いて効率的に計算ができる。そのため結果として、2 変数関数 $\mathsf{mcarry}(a,\, b)$ を直接評価することを避け、全ての途中計算を 1 変数関数評価のみを通じて行うことで効率化が達成される。

3.2 研究成果 [NOFN24]

完全準同型暗号では単純な加算、乗算は比較的効率的に計算が可能である。一方、加算、乗算以外の複雑な計算は非線形演算と呼ばれ、その実現手法は必ずしも自明ではない。非線形演算の計算のために用いられる一手法として、ルックアップテーブル評価手法と呼ばれる計算アプローチがある。ルックアップテーブル評価手法とは計算の真理値表に相当するものを事前に作成しておき、入力に相当する出力ビットを、ルックアップテーブルから入力を暗号化したままで適切に選択する手法である。

このような手法では計算高速化は可能であるが、高速化すればするほど必要なメモリの量が増加するというトレードオフも存在する。本研究の成果 [NOFN24] では、まず入力を暗号化したまま必要な出力を選択する処理において、真理値表の性質によっては不要に繰り返し計算される部分があることに着目した。そしてこの不要な繰り返し計算部分を検出し、一度計算した結果を再利用することで繰り返し計算を回避し、全体として計算速度を向上できることを示した。また現実的に許容可能な範囲で必要メモリ量を

増加させる最適なパラメータを特定することで、計算処理の高速化を最大限に実現するルックアップテーブル評価手法を提案した。

3.3 研究成果 [TN24]

秘密分散に基づく手法では、データを複数の断片に分割し、断片情報を計算参加者らに分配することで元の値を秘匿したまま計算を行う。秘密分散手法として広く知られたものとして、シャミア秘密分散や複製型秘密分散がある。特に複製型秘密分散は計算参加者数がそれほど多くない場合に有効と考えられる秘密分散である。複製型秘密分散を用いた効率的な既存手法は存在するが、これらには、不正な参加者が他の参加者が正しい計算結果を得ることを妨害しうるという課題があった。本研究の成果 [TN24] では、こうした不正者の妨害に対して耐性を持ち、全ての計算参加者が正しい計算結果を復元できるようにするための改良を行った。また、この安全性の向上によって新たに発生しうる計算オーバーヘッドを抑制するための手法も併せて提案し、実行性能の低下を小さく抑えることに成功した。

より具体的にはこの方式では (2, 5) 複製型秘密分散を用いている。これは全体で 5 人の計算参加者がおり、2 人の断片情報を組み合わせることで元の情報を復元できる形で、全ての入力が計算参加者間で秘密分散される。既存の方式 [DEK21] では 4 人の計算参加者を仮定しており、計算実行中にある計算参加者による不正な挙動が観測されたとき、その不正な計算参加者を特定する処理が必要になっていた。本提案 [TN24] では計算参加者を 1 人増やし 5 人とすることで、不正な計算参加者を特定するための処理を大きく削減でき、より通信量の少ない方式を構成できることを示した。

3.4 研究成果 [XN24, XN25]

従来の Homomorphic Secret Sharing (HSS) 方式においては、計算サーバが誤動作した場合や、意図的に不正な動作を行った場合に、その出力の正当性を検証する手段が存在しない、あるいは限定的であるという問題があった。すなわち、計算結果が改竄されていたとしても、クライアントがその正否を確認することが困難であり、信頼性の高い応用には適用が制限されていた。

本研究では、こうした問題に対処するため、Verifiable Homomorphic Secret Sharing (VeHSS) と呼ばれる新たな構成を提案し、計算サーバが出力する結果の正当性をクライアント側で検証可能とする枠組みを構築した。本構成は、Boyle らによる HSS 構成 [BKS19] が SIMD (Single Instruction, Multiple Data) 型の並列ベクトル演算に対応している点に着目している。

具体的には、入力ベクトルの各スロット (要素) に対し、計算対象の本データに混在させる形で、あらかじめクライアントが生成したランダムなダミー平文値 (検証用乱数) を一部のスロットに埋め込む。この検証用スロットに対応する出力については、クライアントが関数 f の適用結果を事前に計算可能であるため、サーバから返された計算結果と一致するかを確認することで、サーバからの出力の正当性が検証可能となる。

提案手法の基本的な考え方は次の通りである：クライアントは、ベクトル内の特定スロットに埋め込んだダミー入力 r に対する期待される関数出力 $f(r)$ を計算しておき、サーバから返された出力ベクトルの該当スロットと照合する。この照合がすべての検証用スロットで成功した場合には、ベクトル全体に対するサーバの計算が正しく実行されたと判定する。これにより、低オーバーヘッドでありながら検証可能性を実現する VeHSS の構成を得ることができた。

また完全準同型暗号へのある種の攻撃手法として [CSBB24, CCP+24] が報告されている。この攻撃では不正な暗号文の復号を、秘密鍵の保有者に依頼し、その復号結果を攻撃者が得ることで、秘密鍵が漏洩する可能性を指摘している。暗号文が攻撃者によって不正に用意されたものか、正しく計算されたものかを復号者が正しく判断することは必ずしも容易ではないため、この攻撃への対処も容易ではないという状況にある。VeHSS で用いた本手法を利用することで、計算結果の暗号文の復号を依頼された者は、計算で用いられた関数の詳細を知っている必要があるものの、[CSBB24, CCP+24] の攻撃の効果を抑制できうることに関しても [XN25] では論じている。

【参考文献】

[BCG+17] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, and Michele Orrù. Homomorphic secret sharing: optimizations and applications. In *Conference on Computer and Communications Security (CCS)*, pages 2105–2122, 2017.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science (ITCS)*, pages 309–325. ACM, 2012.

[BKS19] Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques, Eurocrypt*, pages 3–33. Springer, 2019.

[CCP+24] Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto. Attacks against the IND-CPA$^{\mathsf{D}}$ security of exact FHE schemes. In *Conference on Computer and Communications Security (CCS)*, pages 2505 – 2519. ACM, 2024.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*, pages 409–437. Springer, 2017.

[CSBB24] Marina Checri, Renaud Sirdey, Aymen Boudguiga, and Jean-Paul Bultel. On the practical CPA^D security of "exact" and threshold FHE schemes and libraries. In *CRYPTO*, pages 3–33. Springer, 2024.

[DEK21] Anders Dalskov, Daniel Escudero, and Marcel Keller. Fantastic four:honest-majority four-party secure computation with malicious security. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2183–2200, 2021.

[DIJL23] Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear MPC from sparse LPN. In *CRYPTO*, pages 315–348. Springer, 2023.

[FV12] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144, 2012.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Annual ACM Symposium on Theory of Computing, STOC*, pages 169–178. ACM, 2009.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, pages 75–92. Springer, 2013.

[KMNN15] Shizuo Kaji, Toshiaki Maeno, Koji Nuida, and Yasuhide Numata. Polynomial expressions of carries in p-ary arithmetics. *arXiv preprint arXiv:1506.02742*, 2015.

[MMN22] Daisuke Maeda, Koki Morimura, and Takashi Nishide. Efficient homomorphic evaluation of arbitrary bivariate integer functions. In *Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pages 13–22. ACM, 2022.

[MMN+24] Daisuke MAEDA, Koki MORIMURA, Shintaro NARISADA, Kazuhide FUKUSHIMA, and Takashi NISHIDE. Efficient homomorphic evaluation of arbitrary uni/bivariate integer functions and their applications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E107.A(3):234–247, 2024.

[NHT+24] Akira Nakashima, Takuya Hayashi, Hikaru Tsuchida, Yukimasa Sugizaki, Kengo Mori, and Takashi Nishide. Faster homomorphic evaluation of arbitrary bivariate integer functions via homomorphic linear transformation. In *Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pages 76–86. ACM, 2024.

[NOF+23] Shintaro Narisada, Hiroki Okada, Kazuhide Fukushima, Shinsaku Kiyomoto, and Takashi Nishide. GPU acceleration of high-precision homomorphic computation utilizing redundant representation. In *Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pages 1–9. ACM, 2023.

[NOFN24] Shintaro Narisada, Hiroki Okada, Kazuhide Fukushima, and Takashi Nishide. Time-memory trade-off algorithms for homomorphically evaluating look-up table in TFHE. In *Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pages 1–10. ACM, 2024.

[TN24] Hikaru Tsuchida and Takashi Nishide. Secure five-party computation with private robustness and minimal online communication. In *International Conference on Provable Security (ProvSec)*, pages 43–62. Springer, 2024.

[XN24] Ye Xu and Takashi Nishide. Verifiable homomorphic secret sharing for SIMD operations. In *International Symposium on Computing and Networking Workshops (CANDARW)*, pages 314–320. IEEE, 2024.

[XN25] Ye Xu and Takashi Nishide. VeHSS: Two-server verifiable homomorphic secret sharing leveraging SIMD operations. *Journal of Information Processing*, 2025. (to appear).

[中林土 +24] 中島明, 林卓也, 土田光, 杉崎行優, 森健吾, and 西出隆志. 整数型準同型暗号における homomorphic linear transformation を用いた任意二変数関数計算. 暗号と情報セキュリティシンポジウム (SCIS, Symposium on Cryptography and Information Security), 2024.

[土西 25] 土田光 and 西出隆志. Faf-安全な複製型秘密分散ベースマルチパーティ計算. 暗号と情報セキュリティシンポジウム (SCIS, Symposium on Cryptography and Information Security), 2025.

＜発表資料＞

題名	掲載誌・学会名等	発表年月
VeHSS: Two-server Verifiable Homomorphic Secret Sharing Leveraging SIMD Operations	Journal of Information Processing, Information Processing Society of Japan	2025 年掲載予定
FaF-安全な複製型秘密分散ベースマルチパーティ計算	暗号と情報セキュリティシンポジウム (SCIS)	2025 年 1 月
Verifiable Homomorphic Secret Sharing for SIMD Operations	International Symposium on Computing and Networking Workshops, IEEE	2024 年 11 月
Faster Homomorphic Evaluation of Arbitrary Bivariate Integer Functions via Homomorphic Linear Transformation	Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC), ACM	2024 年 10 月
Time-Memory Trade-off Algorithms for Homomorphically Evaluating Look-up Table in TFHE	Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC), ACM	2024 年 10 月
Secure Five-party Computation with Private Robustness and Minimal Online Communication	International Conference on Provable Security (ProvSec), Springer	2024 年 9 月
Efficient Homomorphic Evaluation of Arbitrary Uni/Bivariate Integer Functions and Their Applications	IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences	2024 年 3 月
整数型準同型暗号における homomorphic linear transformation を用いた任意二変数関数計算	暗号と情報セキュリティシンポジウム (SCIS)	2024 年 1 月
GPU Acceleration of High-Precision Homomorphic Computation Utilizing Redundant Representation	Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC), ACM	2023 年 11 月