Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Fast post-quantum private set intersection from oblivious pseudorandom function for mobile social networks[☆]

Zhuang Shan[a], Leyou Zhang[a,*], Qing Wu[b], Qiqi Lai[c], Fuchun Guo[d]

[a] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
[b] School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[c] School of Computer Science, Shaanxi Normal University, Xi'an 710121, China
[d] Centre for Computer and Information Security Research, University of Wollongong, Wollongong, NSW 2522, Australia

ARTICLE INFO

Keywords:
Mobile social networks
Private set intersection
Oblivious pseudorandom function
Private information retrieval

ABSTRACT

Mobile social networks have become integral to our daily lives, transforming communication methods and facilitating social interactions. With technological advancements, users generate vast amounts of valuable and sensitive personal data, which is stored on servers to enable instant information sharing. To protect the sharing data, each platform has implemented many techniques such as end-to-end encryption mechanisms, fully homomorphic encryption, etc. However, these approaches face several security and privacy challenges, including potential leaks of user data, vulnerabilities in encryption that expose privacy ciphertexts to probabilistic attacks, and threats posed by future quantum computers.

Aimed at the above, we introduce a private set intersection (PSI) protocol based on oblivious pseudorandom functions (OPRF) under ring LPR problem from lattice. The proposed perturbed pseudorandom generator not only enhances the PSI's resistance to probabilistic attacks, but also leads to generate a more efficient OPRF and a PSI. It boasts a time complexity of $O(n \log n)$ and is superior to existing well-known fast post-quantum PSI protocol operating at $O(mn \log(mn))$, where m is the bit length of the cryptographic modulus and n represents the dimension of the security parameter. Simulation experiments and security analyses demonstrate that our proposal effectively preserves user privacy, ensures collusion resilience, verifies computation results, and maintains low computational costs. Finally, as an expansion of our OPRF, we also give a fast private information retrieval (PIR) protocol.

1. Introduction

Mobile social networks have greatly enriched the ways people communicate and enhanced the convenience of social interactions. With the development of technology, users generate a large amount of useful and sensitive personal data within mobile social networks. This data often needs to be stored and processed to provide more personalized services and experiences [1,2]. However, due to the limited storage capacity of mobile social network devices, it is impossible to store all the data generated at any given moment, which presents challenges for data storage and privacy protection.

To address this issue while ensuring data confidentiality and security, many mobile social network platforms have started adopting advanced privacy-preserving technologies, such as private set intersection (PSI). The technology allows two or more parties to securely compute the intersection of their datasets without disclosing their respective data sets. This way, even if data is stored in distributed systems, it can effectively prevent data breaches and violations of user privacy, such as those caused by data leaks or unauthorized access.

The application of PSI in mobile social networks not only enhances data security but also strengthens user trust in the platform, which is crucial for protecting user privacy and improving the platform's competitiveness. In this way, mobile social networks can continue to provide a rich and vibrant social experience and efficient information services while safeguarding personal privacy. Furthermore, as an important application in the field of privacy computing, PSI has recently garnered widespread attention due to its efficiency and practicality, jointly promoting the rapid implementation of privacy computing technology and ensuring the secure flow and value extraction of data elements.

[☆] This document is the results of the research project funded by the National Science Foundation.

[*] Corresponding author.

E-mail addresses: arcsec30@stu.xidian.edu.cn (Z. Shan), lyzhang@mail.xidian.edu.cn (L. Zhang), xiyouwuq@126.com (Q. Wu), laiqq@snnu.edu.cn (Q. Lai), fuchun@uow.edu.au (F. Guo).

https://doi.org/10.1016/j.sysarc.2025.103346
Received 3 November 2024; Received in revised form 24 December 2024; Accepted 16 January 2025
Available online 25 January 2025

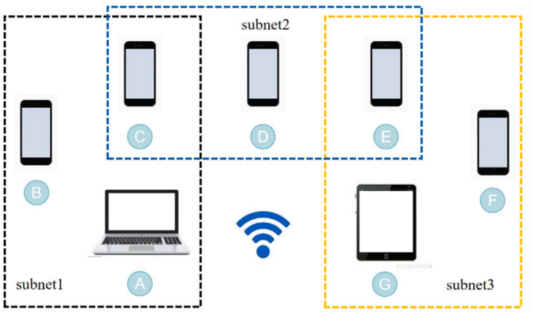

Fig. 1. Mobile social networks.

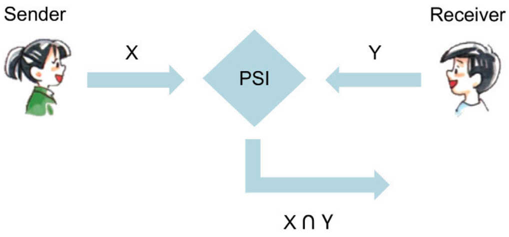

Fig. 2. Private set intersection.

There are many common construction tools for PSI [3], and oblivious transfer (OT) is one of them. An OT [4] is a crucial tool used for secure multiparty computation. In this tool, the sender transmits data from a set of messages to the receiver but remains oblivious to which specific message was sent, while the receiver is unaware of the other messages they did not receive. This protocol is also known as the oblivious transfer protocol. The essence of an oblivious pseudorandom function is a pseudorandom function (PRF) enhanced with oblivious transfer capabilities.

In 1986, Goldreich, Goldwasser, and Micali introduced a new cryptographic primitive known as the pseudorandom function, whose output appears to be randomly chosen [5]. Two decades later, Naor and Reingold [6] noticed that their number-theoretic PRF allows for an interactive and oblivious evaluation, where a “client” with input x obtains $F_k(x)$ for a function $F_k(x)$ that is contributed by a “server”. Neither does the client learn the function (i.e., its key k), nor does the server learn x or $F_k(x)$. Freedman et al. later called such two-party protocol an OPRF and gave first formal definitions and two OPRFs based on the Naor-Reingold PRF [7]. In 2009, Jarecki and Liu presented an efficient OPRF for securing intersection data [8].

Oblivious pseudorandom functions have been utilized in PSI [9]. The additional functionalities of oblivious pseudorandom functions also exhibit diversity, such as verifiable oblivious pseudorandom functions (VOPRF, [10]) and partially oblivious pseudorandom functions (POPRF, [11]).

Currently, OPRFs still faces challenges, as summarized by Casacuberta, Hesse, and Lehmann [12]. Efficient OPRF constructions often rely on discrete-log or factoring-type hardness assumptions, which are vulnerable to quantum computers. This paper aims to address this by constructing OPRFs based on lattice-hardness assumptions and improving their efficiency (see Figs. 1 and 2).

1.1. Contributions

Regarding the open problem proposed by Casacuberta, there are currently quantum-resistant OPRFs, namely Albrecht et al.’s lattice-based VOPRF [10] and Boneh et al.’s isogeny-based OPRF [13]. Both constructions represent significant feasibility results but require further research to improve their efficiency [12]. So, fast post-quantum private set intersection from oblivious pseudorandom function is proposed in this paper, and it has the following advantages:

- *Symmetric encryption is adopted, which is efficient and reduces the risk of privacy leakage*. The PSI in this paper is constructed based on OPRF, which belongs to asymmetric encryption, thus reducing the number of interactions between users and lowering the risk of user privacy leakage. Compared to symmetric encryption, the operational cost of asymmetric encryption is lower, reducing reliance on authoritative institutions.
- *The structure of OPRF is simple, and it is relatively efficient in post-quantum OPRF*. The OPRF used to construct PSI in this paper is based on a new lattice problem, namely the learning parity with rounding over ring problem(Ring-LPR). The Ring-LPR problem not only has a simple structure but also possesses the capability to resist quantum attacks.
- *A perturbed pseudorandom generator (PPRG) can withstand probabilistic attacks*. In addition to OPRF, the PSI in this paper also includes a structure with a perturbed pseudorandom generator, which can overcome the weakness of weak encryption in symmetric encryption, thereby preventing adversaries from guessing the corresponding plaintext using statistical methods on the ciphertext ratios.

1.2. Technical overview

We adopted oblivious transfer technique and hamming correlation robustness, both of which are used in the OPRF construction presented in this paper. For the incidental pseudorandom function subject, we initially aimed to use learning parity with noise (LPN) over rings. However, this approach results in varying encryption outcomes for the same private data, preventing the recipient from matching the private data. Thus, we sought to make LPN over rings behave consistently like learning with rounding (LWR), leading to the introduction of the concept of learning parity with rounding over rings (LPR over rings) in this paper.

To prove that LPR over rings is quantum-resistant, we established a reduction bridge between LPR over rings and LWR. Yes, LPR over rings is reduced to LWR, not LPN over rings. For $(q = 2^n, p)$-LWR instances, we demonstrated the hardness of $(q = 2, p = 1)$-LWR instances and $(q = 2, p = 1)$-LWR over rings, where $(q = 2, p = 1)$-LWR over rings corresponds to LPR over rings. To verify that the computational efficiency of the post-quantum OPRF in this paper is quite fast, we compared the OPRF with the LWE-instantiated OPRF from [14]. The results showed that, as theoretical analysis suggested, the computation efficiency improves with the increase of security parameters.

Based on OPRF, we constructed private set intersection (PSI) based on OPRF. Since the paper [15] analyzed that PSI based on symmetric encryption does not resist probabilistic attacks and proposed the concept of perturbed pseudorandom generator, we used LPN over rings to construct a pseudorandom generator and proved that it satisfies the definition of PPRG as given in [15].

1.3. Organizations

The structure of this paper is as follows. Section 3 provides the necessary definitions and lemmas as a foundation for the readers’ knowledge. Section 4 presents the construction and efficiency analysis of OPRF, along with the definition and reduction of Ring-LPR. Section 5 details the construction of the PSI in this paper, security proofs, and LWE-based efficiency analysis, as well as the construction of the PPRG and the proof of its pseudorandomness. Finally, Section 6 summarizes the advantages and limitations of the PSI presented in this paper, as well as the extension of OPRF to PIR

2. Preliminary

Each element of a lattice in $\mathbb{R}^n$ can be expressed linearly by n linearly independent vector integer coefficients. This set of linearly independent vectors is called a lattice basis, and we know that the lattice basis is not unique. Given a set of lattice bases $(v_1, \dots, v_n)$ in the lattice $\mathcal{L}$, then the fundamental parallelelepiped is

$$\mathcal{P}(v_1, \dots, v_n) = \left\{ \sum_{i=1}^{n} k_i v_i \middle| k_i \in [0,1) \right\}.$$

If the lattice base $(v_1, \dots, v_n)$ is determined, use the symbol $\mathcal{P}(\mathcal{L})$ to replace $\mathcal{P}(v_1, \dots, v_n)$. $\forall x \in \mathbb{R}^n$, project it onto $\mathcal{P}(\mathcal{L})$. According to the properties of projection, there is a unique $y \in \mathcal{P}(\mathcal{L})$ makes $y - x \in \mathcal{L}$. Use the symbol $\det(\mathcal{L})$ to represent the volume of the fundamental parallelelepiped of the lattice $\mathcal{L}$. In other words, the symbol $\det(\mathcal{L})$ represents the determinant of a matrix composed of a set of lattice bases $(v_1, \dots, v_n)$. For a given n dimensional lattice, the $\det(\mathcal{L})$ size of any set of lattice bases of the lattice is constant.

Given n lattice $\mathcal{L}$, $(v_1, \dots, v_n)$ and $(u_1, \dots, u_n)$ are two arbitrary groups of lattice $\mathcal{L}$ respectively lattice bases. Therefore, there is $v_i = \sum_{j=1}^{n} m_{ij} u_j$ and $u_i = \sum_{j=1}^{n} m'_{ij} v_j, i \in \{1, \dots, n\}$, there are two integer matrices M and M' such that

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = M \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \text{ and } \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = M' \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

It is easy to prove that M and M' are inverse to each other, and M and M' are both integer matrices, there are $\det(M) \cdot \det(M') = 1$ and $\det(M) = \det(M') = \pm 1$, so

$$\det(v_1, \dots, v_n) = \pm \det(u_1, \dots, u_n).$$

Definition 1. An ideal lattice is a subset of rings or domains that satisfies the following two properties:

1. Additive closure: If any two elements in the ideal are added, the result is still in the ideal. In other words, for any elements a and b in the ideal, $a + b$ also belongs to that ideal.
2. Multiplicative absorptivity: If an element in the ideal is multiplied by any element in the ring (or field), the result is still in the ideal. In other words, for any element a in the ideal and any element r in the ring (or field), ar and ra belong to that ideal.

For a commutative ring, further require that the ideal be closed for both addition and multiplication. Such an ideal is called a true ideal.

Definition 2. Referring to the definition of ideal, the ideal lattice $\mathcal{I}$ is a subset of the lattice $\mathcal{L}$ that satisfies the following two properties:

1. Additive closure: If any two elements in an ideal lattice are added, the result is still in the ideal lattice. In other words, for any elements a and b in an ideal lattice, $a+b$ also belongs to that ideal lattice.
2. Multiplicative absorptivity: If an element in an ideal lattice is multiplied by an element in any other ideal lattice, the result remains in the ideal lattice. In other words, for any element a in the ideal and any element r in another ideal lattice, both ar and ra belong to that ideal lattice.

Corollary 1. *The ideal lattice $\mathcal{I}$ is a true idea of the lattice $\mathcal{L}$.*

For $f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$ is mapped to

$$Rot(f) = a_0 I + a_1 X + \cdots + a_{n-1} X^{n-1} \in \tilde{\mathcal{R}}.$$

Among them, $\tilde{\mathcal{R}}$ is the mapping of all $\mathbb{Z}[x]/<x^n + 1>$ to the elements in the ideal lattice $\mathcal{I}$ collection, and

$$X = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

So there is

$$Rot(f) = \begin{pmatrix} a_0 & -a_{n-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix},$$

it is easy to prove that this mapping relationship is isomorphic.

Definition 3 (*Learning with Rounding, [16,17]*)**.** Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $p = p(\lambda)$ be integers. The LWR problem states that for $A \in \mathbb{Z}_q^{m \times n}$, $s \in \mathbb{Z}_q^n$, $u \in \mathbb{Z}_q^m$ the following distributions are computationally indistinguishable: $(A, \lfloor As \rfloor_p) \approx_C (A, \lfloor u \rfloor_p)$. Here $\lfloor x \rfloor_p = \lfloor \frac{q}{p} x \rfloor$, $\lfloor x \rfloor$ represents the floor function, which rounds down to the nearest integer. For example, $\lfloor 3.14 \rfloor = 3$ and $\lfloor 3 \rfloor = 3$.

Definition 4 (*Learning Parity with Noise, [18,19]*)**.** Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$ be integers. The LPN problem states that for $A \in \mathbb{Z}_2^{m \times n}$, $s \in \mathbb{Z}_2^n$, $u, e \in \mathbb{Z}_2^m$ the following distributions are computationally indistinguishable: $(A, As + e) \approx_C (A, u)$.

Definition 5 (*Hamming Correlation Robustness, [14]*)**.** For a hash function $\mathcal{H}(\cdot)$ and a pseudorandom function $F_k(\cdot)$ with key k, $\mathcal{H}(\cdot)$ is Hamming correlation robust if $\mathcal{H}(x) \approx_C F_k(x)$.

Definition 6 (*OT*[1])**.** The message sender sends data to the receiver from a set of pending messages but remains oblivious to which specific message was sent. Meanwhile, the receiver is unaware of the additional data they want to receive. This protocol is also known as oblivious transfer.

Definition 7 (*OPRF, [20]*)**.** Let the PRF key k consist of two bit-strings $q, s \in \{0,1\}^\lambda$. Let $F(\cdot)$be a pseudorandom code that produces a pseudorandom string and let $\mathcal{H}$ be a hash function. The pseudorandom function is computed as

$$\mathrm{OPRF}_k(x) = \mathcal{H}(q \oplus [F(x) \cdot s]),$$

where $\cdot$ denotes bitwise-AND and $\oplus$ denotes bitwise-XOR. For a randomly generated s, if $F(x)$ has enough Hamming weight then the function $\mathrm{OPRF}_k(x)$ is pseudorandom assuming the hash function $\mathcal{H}$ is correlation robust.

Definition 8 (*PSI, [14]*)**.** PSI enables two parties, each holding a private set of elements, to compute the intersection of the two sets while revealing nothing more than the intersection itself.

Definition 9 (*Dihedral Coset Problem*)**.** Given a security parameter κ, for an instance of the DCP_q^ℓ problem, where N denotes the modulus and ℓ represents the number of states. Each state is expressed as

$$|0\rangle|x_i\rangle + |1\rangle|(x_i + s) \bmod q\rangle, \quad i \le \ell,$$

and it stores $1 + \lceil \log_2 q \rceil$ bits, where $x \in_R \mathbb{Z}_q^n$ and $s \in \mathbb{Z}_q^n$. If s can be computed with probability $\mathrm{poly}(1/\log q)$ in time $\mathrm{poly}(\log q)$, then the DCP_q^ℓ problem is considered to be broken.

[1] https://blog.csdn.net/m0_61869253/article/details/139362753

Note 1. *The Dihedral Coset Problem is a difficult problem in quantum computing, and solving it has a time complexity of $O(e^n)$ or $O(n!)$.*

Lemma 1. *If an efficient algorithm $\mathcal{W}$ can solve DCP_2^{ℓ} in polynomial time, then there exists an efficient algorithm $\mathcal{W}'$ that can solve DCP_q^{ℓ} in polynomial time.*

Proof. We use a proof by contradiction. Suppose $q = 2^n$ and there exists an efficient algorithm $\mathcal{W}$ that can solve DCP_2^{ℓ} in polynomial time. For instances of DCP_4^{ℓ}, we have

$$|0\rangle|x_i\rangle+|1\rangle|(x_i+s) \bmod 4\rangle = |0\rangle|x_i'\rangle + |1\rangle|(x_i'+s') \bmod 2\rangle \\ + 2(|0\rangle|x_i''\rangle + |1\rangle|(x_i'+s'') \bmod 2), i \leq \ell,$$

so running the algorithm $\mathcal{W}$ twice will solve $\mathrm{DCP}_{4=2^2}^{\ell}$. Similarly, running $\mathcal{W}$ four times will solve $\mathrm{DCP}_{16=2^4}^{\ell}$, and continuing in this manner, running the algorithm $\mathcal{W}$ n times will solve DCP_q^{ℓ}. Let $O(\mathcal{W})$ represent the time complexity of the algorithm $\mathcal{W}$. Thus, we have $\mathcal{W}' \leq nO(\mathcal{W})$ and algorithm $\mathcal{W}'$ is an efficient algorithm. □

Definition 10 (*Extrapolated Dihedral Coset Problem with model* 2, *[21]*)**.** Given a security parameter κ, an instance of $\mathrm{EDCP}_{n,2,\rho}^{\ell}$ is provided, where 2 denotes the modulus, ρ represents the probability density function, and ℓ denotes the number of states. Each state is expressed as

$$\sum_{j \in \mathrm{supp}(\rho)} \rho(j)|j\rangle|(x_i + js) \bmod 2\rangle, i \leq \ell,$$

and stores 2 bits, where $x_i \in_R \mathbb{Z}_2^n$ and $s \in \mathbb{Z}_2^n$. If s can be determined with probability $\mathrm{poly}(1/(n \log 2))$ in time $\mathrm{poly}(n \log 2)$, then the $\mathrm{EDCP}_{n,2,\rho}^{\ell}$ problem is considered to be broken.

Lemma 2. *If there exists an algorithm for solving $EDCP_{n,4,\rho}^{\ell}$, then this algorithm can also solve DCP_4^{ℓ}.*

Proof. Let

$$|b\rangle = \frac{1}{\sqrt{2}}|0\rangle|x_i\rangle + \frac{1}{\sqrt{2}}|1\rangle|(x_i+s) \bmod 4\rangle.$$

Thus, $\rho(0)|0\rangle = \frac{1}{\sqrt{2}}|0\rangle$ and $\rho(1)|1\rangle = \frac{1}{\sqrt{2}}|1\rangle$. Hence, DCP_2^{ℓ} is a special case of $\mathrm{EDCP}_{n,2,\rho}^{\ell}$. Therefore, if there exists an algorithm for solving $\mathrm{EDCP}_{n,2,\rho}^{\ell}$, this algorithm can also solve DCP_2^{ℓ}. □

Lemma 3 (*[21]*)**.** *Let $(n, q, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP and (n, q, α) be an instance of LWE. If there exists an algorithm for solving $LWE_{n,q,\alpha}$, then there exists an algorithm for solving G-$EDCP_{n,q,\rho_r}^{\ell}$.*

Corollary 2. *Let $(n, 2, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP and $(n, 2, \alpha)$ be an instance of LPN. If there exists an algorithm for solving $LPN_{n,\alpha}$, then there exists an algorithm for solving G-$EDCP_{n,2,\rho_r}^{\ell}$.*

3. Ring-LPR based OPRF

3.1. Constructing OPRF

Fig. 3 presents the ring LPR-based oblivious pseudorandom function. In the next section, we will prove the security of the oblivious pseudorandom function.

3.2. Security proof of OPRF

In this subsection, we will provide the definition of the underlying lattice problem for OPRF, learning parity with rounding, and its reduction proof.

Definition 11 (*Learning Parity with Rounding*)**.** Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$ be integers. The LPR problem states that for $A \in \mathbb{Z}_2^{m \times n}$, $s \in \mathbb{Z}_2^n$, $u \in \mathbb{Z}_2^m$ the following distributions are computationally indistinguishable: $(A, \lfloor As \bmod 4 \rfloor_1) \approx_C (A, \lfloor u \rfloor_1)$.

Definition 12 (*Learning Parity with Rounding Over Ring*)**.** The Ring LPR problem states that for $a, s, u \in \mathcal{R}_2$ the following distributions are computationally indistinguishable: $(a, \lfloor as \bmod 4 \rfloor_1) \approx_C (a, \lfloor u \rfloor_1)$.

Lemma 4. *For an LWR problem instance $\lfloor As \rfloor_p$, if there exists an algorithm $\mathcal{W}$ for solving s from $\lfloor As \rfloor_1$, then there also exists an algorithm $\mathcal{W}'$ for solving the LWR problem.*

Proof. Given that there exists an algorithm $\mathcal{W}$ that can solve $\lfloor As \rfloor_1 = \lfloor \frac{As}{q} \rfloor$, for an LWR problem instance $\lfloor As \rfloor_p$, we have:

$$\begin{aligned} \frac{1}{p}\lfloor As \rfloor_p &= \frac{1}{p}\left\lfloor \frac{pAs}{q} \right\rfloor \\ &= \frac{1}{p}\left(\frac{pAs}{q} + e\right) \quad (e \in (-1, 0]^m) \\ &= \frac{1}{q}As + e' \quad \left(e' \in \left(-\frac{1}{p}, 0\right]^m\right) \\ &\approx \lfloor As \rfloor_1. \end{aligned}$$

Thus, the algorithm $\mathcal{W}$ can be used to solve the LWR problem. □

We get next corollary by Lemma 3.

Corollary 3. *Let $(n, 2, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP and $(n, 2, \alpha)$ be an instance of 2-LWR. If there exists an algorithm for solving 2-LWR, then there exists an algorithm for solving G-$EDCP_{n,2,\rho_r}^{\ell}$.*

Corollary 4. *Let $(n, 2, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP and $(n, 2, \alpha)$ be an instance of LPR. If there exists an algorithm for solving LPR, then there exists an algorithm for solving G-$EDCP_{n,2,\rho_r}^{\ell}$.*

Lemma 5. *If there exists an algorithm $\mathcal{W}$ for solving the Ring-LPR problem, then there also exists an algorithm $\mathcal{W}'$ for solving the LPR problem.*

Proof. For an instance of the inner product Ring-LPR

$$b = \lfloor a \cdot s \rfloor_1$$

where $a = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$, we can represent a as a circulant matrix, specifically

$$A_1 := \begin{pmatrix} a_0 & -a_{n-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}.$$

Thus,

$$b = \lfloor a \cdot s \rfloor_1 \Rightarrow b = A_1 s.$$

where $a = (a_0, a_1, \ldots, a_{n-1}) \leftarrow a = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$. We use a proof by contradiction. Suppose there exists an efficient algorithm $\mathcal{W}$ that can solve Ring-LPR in polynomial time. We take the first row from A_1, denote it as α_1, and have $\lfloor \alpha_1 s \rfloor_1 = b_1$, where b_1 is the first component of b. For the LWR problem instance, $\vec{\beta} = \lfloor A\vec{s} \rfloor_1$, assume

PRF.Setup The users P_1 and P_2 agree on λ, δ, protocol parameters m, w, and two hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathcal{R}_{\{0,1\}}$ and $\mathcal{H}_2 : \mathcal{R}_{\{0,1\}} \rightarrow [m]^w$.

PRF.Enc P_2 selects a pseudorandom function key $k \in \mathcal{R}_{\{0,1\}}$. For input private data $x \in \mathcal{X} \subset \{0,1\}^*$, compute

$$v := \mathcal{H}_2(F_k(\mathcal{H}_1(x))) = \mathcal{H}_2(\lfloor k\mathcal{H}_1(x) \rfloor_1).$$

P_2 initializes a matrix $D \in 1^{m \times w}$ and sets $D_i[v[i]] = 0$.

PRF.OT

- P_1 and P_2 execute oblivious transfer, where P_1 sends $s[1], \dots, s[w]$. P_2 receives random messages $\{r_i^{(0)}, r_i^{(1)}\}_{i \in [w]}$ and P_1 receives $\{r_i\}_{i \in [w]}$, where $r_i = r_i^{s[i]}$.
- P_2 performs
  - Let $\{r_i^{(0)}\}_{i \in [w]}$ be the column vectors of A and compute $B = A \oplus D$.
  - Compute $\Delta_i = B_i \oplus r_i^{(1)}$, $i \in [w]$ and send the results to P_1.
- P_1 computes C, where: if $s[i] = 0$ then $C_i = r_i$; otherwise, $C_i = r_i \oplus \Delta_i$.

Fig. 3. Oblivious Pseudorandom Function (OPRF).

$\Lambda^T = (\alpha_1, \alpha_2, \dots, \alpha_m)$.

Thus, we use the algorithm $\mathcal{W}$ m times to find β_i such that $\lfloor \gamma_i \rfloor_1 = \beta_i = \lfloor \alpha_1 s_1 \rfloor_1$, and thus we can solve the equation

$$\gamma = \Lambda\vec{s}, \gamma^T = (\gamma_1, \dots, \gamma_m).$$

Assuming that the time complexity of solving s from LWR problem instance is $O(\Lambda, \beta)$, according to Corollary 3, let $O(\gamma = \Lambda\vec{s})$ be the computational complexity of solving the equation $\gamma = \Lambda\vec{s}$, we have

$$mO(\mathcal{W}) + O(\gamma = \Lambda\vec{s}) \geq O(\Lambda, \beta) \geq O(n!) \text{ or } O(e^n).$$

Let $m = n$, then

$$O(\mathcal{W}) \geq \frac{O(\Lambda, \beta) - O(\gamma = \Lambda\vec{s})}{n}$$
$$\geq \frac{O(n!) - O(\gamma = \Lambda\vec{s})}{n} \text{ or } \frac{O(e^n) - O(\gamma = \Lambda\vec{s})}{n}.$$

This contradicts the assumption that there is an efficient algorithm $\mathcal{W}$ that can solve the inner product Ring-LPR in polynomial time, thus the theorem holds. □

3.3. Efficiency analysis

This section simulates the OPRF computation efficiency of this paper and OPRF in [14] on MAC, Pad and Phone. The PRF of [14] is instantiated based on LWE.

3.3.1. Efficiency analysis on MAC

The tools used in the subsection are Python 3.12, the programs are performed on MacBook Air MAC Desktop Apple M1, RAM 8.00 GB (see Fig. 4).

3.3.2. Efficiency analysis on mobile pad

The tools used in the subsection are Pydriod 3, the programs are performed on Xiaomi Pad 6 Pro File Explorer 1th Qualcomm(R)AI Engine(TM) Xiaolong 8+ mobile platform@3.2 GHz, RAM 8.00+3.00 GB (see Fig. 5).

3.3.3. Summary of data comparison

From the simulation results, it can be seen that for $n \leq 250$, the LWE-based OPRF in [14] is slightly faster, while for $n > 250$, the ring LPR-based OPRF in this paper is faster. Furthermore, as n increases, the advantages of ring LPR become more pronounced. Based on the simulation results for Pad, the OPRF in this paper is more stable; although there are fluctuations, they are less significant compared to the LWE-based OPRF in [14].

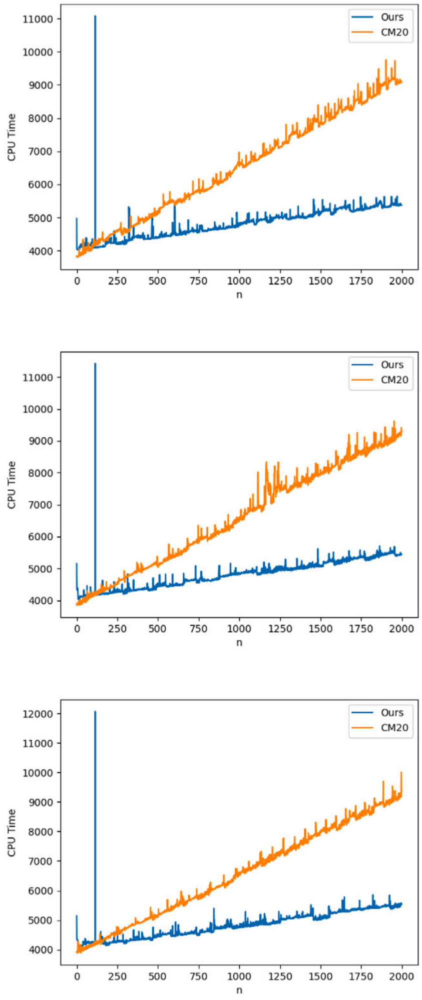

Fig. 4. Parallel comparison of OPRF on MAC, where n represents the security parameter, unit is microseconds.

4. PSI based on OPRF

In this paper, apart from OPRF, another tool used in the construction of PSI is a perturbed pseudorandom generator [15]. The perturbed pseudorandom generator in this paper is constructed from Ring-LPN.

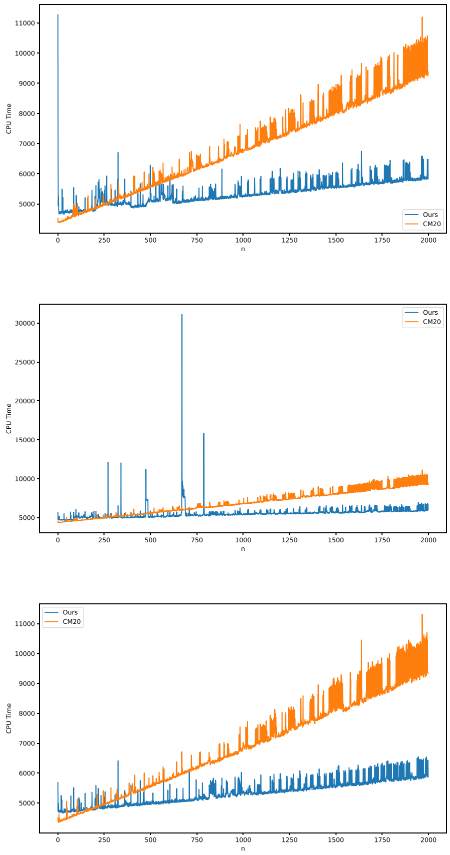

Fig. 5. Parallel comparison of OPRF on mobile pads, where n represents the security parameter, unit is microseconds.

Next, we will present the reduction process for Ring-LPN.

4.1. Reduction of ring-LPN

Definition 13 (*Learning Parity with Noise Over Ring*)**.** The learning parity with noise over ring problem states that for $a, s, e, u \in \mathcal{R}_{\{0,1\}}$ the following distributions are computationally indistinguishable: $(a, as + e) \approx_C (a, u)$.

Corollary 5. *If there exists an efficient algorithm $\mathcal{W}$ that can solve the Ring-LPN problem in polynomial time, then there also exists an algorithm $\mathcal{W}'$ that can solve the LPN problem.*

Proof. The proof method is similar to that of Lemma 5, but this way the computational complexity of $\mathcal{W}$ will decrease. If we want the Ring-LPN problem to be 'approximately' as hard as the LPN problem, then for the security parameters κ_1 of the Ring-LPN problem and κ_2 of the LPN problem, we have

$$\frac{e^{\kappa_1}}{\kappa_1^2} \geq e^{\kappa_2}, \text{ or } \frac{(\kappa_1)!}{\kappa_1^2} \geq (\kappa_2)!.$$

Thus, we can roughly obtain $\kappa_1 \geq 1.5\kappa_2$ and $\kappa_2 \geq 12$. Note that $O(n)$ is an asymptotically large quantity with respect to n. We use the most extreme case to determine the relationship between κ_1 and κ_2. □

4.2. Perturbed pseudorandom generator

Definition 14. Let $a = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathcal{R}_{\{0,1\}}$. Define the norm of a as $\|a\|$, and

$$\|a\| = \sqrt{\sum_{i=0}^{n-1} |a_i|^2}.$$

Setup Let $(a, x, e)^T \in \mathcal{R}^3_{\{0,1\}}$.

Enc Compute

$$G_\gamma(x) = ax + e \bmod (x^n + 1) \bmod 2.$$

Fig. 6. Pseudorandom generator with perturbation $G_\gamma(\cdot)$.

Definition 15 ([15])**.** A pseudorandom generator with perturbation, denoted as $G_\gamma(\cdot)$, is defined such that for $x_1, x_2 \in \mathcal{X}$, there exists γ satisfying the following conditions:

1. When $x_1 = x_2$, $\Pr(G_\gamma(x_1) = G_\gamma(x_2)) \leq O(\exp(-n))$,
2. When $x_1 = x_2$, such that $\|G_\gamma(x_1) - G_\gamma(x_2)\| < \gamma$, there exists N such that $\|G_\gamma(x_1) - G_\gamma(x_2)\| \geq \gamma \cdot N$, where clearly $N = 1$ is optimal.

Theorem 1. *The Ring-LPN problem itself can be viewed as a pseudorandom function with perturbations.*

Proof. We prove each statement separately. First, when $x_1 = x_2$, we have

$$\Pr\left(G_\gamma(x_1) = G_\gamma(x_2)\right) = \Pr(e_1 = e_2) = \frac{1}{2^n}.$$

Additionally, set $\gamma = \sqrt{n+1}$, so

$$\|(Ax_1 + e_1) - (Ax_2 + e_2)\| = \|e_1 - e_2\| < \gamma.$$

When $x_1 \neq x_2$, set $v_1 = G_\gamma(x_1)$, $v_2 = G_\gamma(x_2)$, and know that

$$\Pr(\|v_1 - v_2\| \leq \sqrt{n}) = \sum_{k=0}^{n} C_n^k \left(\frac{1}{3}\right)^k \left(\frac{1}{2}\right)^{n-k} + \sum_{k=0}^{n/2} C_n^k \left(\frac{1}{3}\right)^k \left(\frac{1}{6}\right)^k \left(\frac{1}{2}\right)^{n-2k}.$$

Because

$$\sum_{k=0}^{n} C_n^k \left(\frac{1}{3}\right)^k \left(\frac{1}{2}\right)^{n-k} = \frac{1}{2^n}\left(\frac{2}{3} + \left(\frac{2}{3}\right)^2 + \cdots + \left(\frac{2}{3}\right)^n\right) = \frac{3}{2^n}\left(1 - \left(\frac{2}{3}\right)^n\right),$$

and

$$\sum_{k=0}^{n/2} C_n^k \left(\frac{1}{3}\right)^k \left(\frac{1}{6}\right)^k \left(\frac{1}{2}\right)^{n-2k} \leq \frac{3 \cdot 6}{17} \frac{1}{2^{n-\frac{n}{2}}}\left(1 - \left(\frac{1}{3 \cdot 6}\right)^{\frac{n}{2}}\right).$$

Therefore

$$\Pr\left(\|v_1 - v_2\| \leq \sqrt{n} < \sqrt{n+1}\right) \leq \frac{1}{2^n}.$$

Thus, there is a very high probability that $\|v_1 - v_2\| \geq \sqrt{n+1}$, and $N = 1$ (see Fig. 6). □

4.3. PSI based on OPRF

Lemma 6. *Assuming $f(y) \approx_C u_1$ and $g(u_1) \approx_C u_2$, then $(g \circ f)(y) \approx_C u_2$.*

1. **Setup** P_1 and P_2 agree on security parameters λ, σ, protocol parameters m, ω, hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$, hamming correlation robustness $\mathcal{H}_2 : \mathcal{R}_{\{0,1\}} \to [m]^{\omega}$, hamming correlation robustness $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \to \mathcal{R}_{\{0,1\}}$ and a $G_\gamma : \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$, a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$.
2. **OPRF Evaluation**
   (a) P_2 sends the PRF key k to P_1.
   (b) $\forall x \in \mathcal{X}$, P_1 computes $v = \mathcal{H}_2(F_k(\mathcal{H}_1(x)))$ and its OPRF value $\psi = G_\gamma(\mathcal{H}_3(C_1[v[1]] \| \cdots \| C_\omega[v[\omega]]))$ and sends ψ to P_2.
   (c) Let Ψ be the set of OPRF values received from P_1. $\forall y \in \mathcal{Y}$, P_2 computes $v = F_k(\mathcal{H}_1(y))$ and its OPRF value $\|\psi - G_\gamma(\mathcal{H}_3(A_1[v[1]] \| \cdots \| A_\omega[v[\omega]]))\| < \sqrt{\omega}\gamma$ and outputs y iff $\psi \in \Psi$.

Fig. 7. PSI based on OPRF.

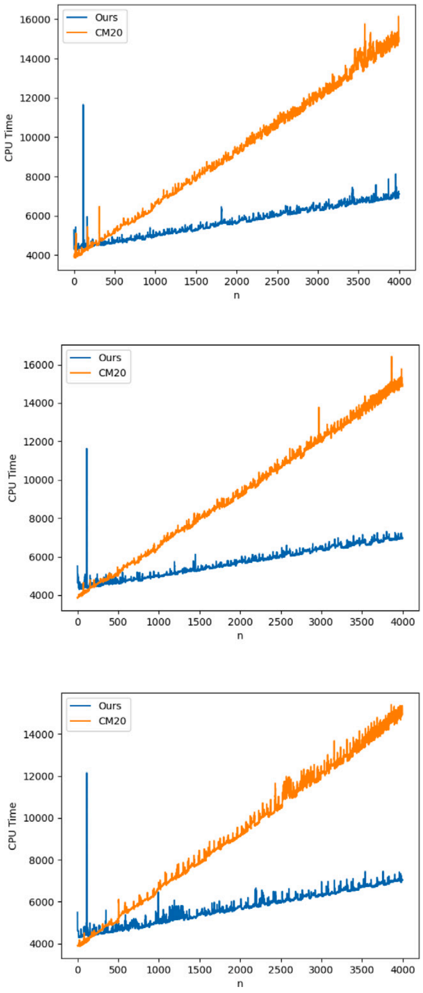

Fig. 8. Parallel comparison of PSI on MAC, where n represents the security parameter, unit is microseconds.

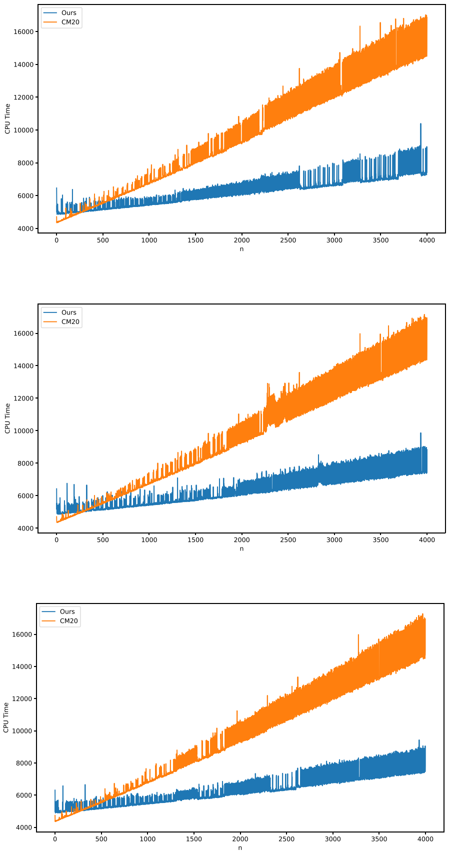

Fig. 9. Parallel comparison of PSI on mobile pads, where n represents the security parameter, unit is microseconds.

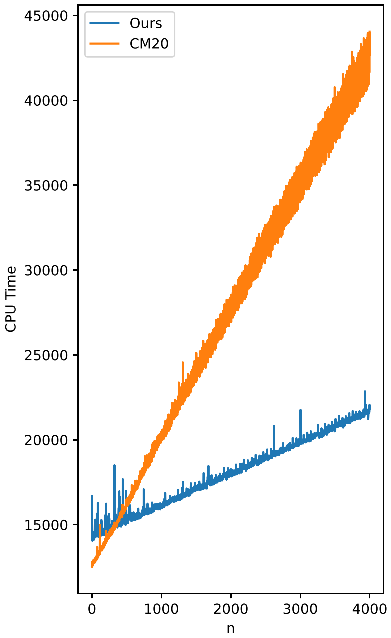

Fig. 10. Comparison of PSI on mobile phones, where n represents the security parameter, unit is microseconds.

1. **Setup** P_s and P_u is server and user whose agree on security parameters λ, σ, protocol parameters m, ω, hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$ and a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$.
2. **OPRF Evaluation**
   (a) P_u sends the PRF key k to P_s.
   (b) $\forall (x,m) \in \mathcal{X} \times \mathcal{M}$, P_s computes $v = F_k(\mathcal{H}_1(x))$ and its OPRF function
   $$\psi(v) = \Psi(x) = \psi_1(v) + \psi_2(v)$$
   and sends $\psi(v)$ to P_u, here $\psi_1(v) = 0, \psi_2(v) = m$. It is needed that $\Psi(x)$ is one way function.
   (c) Let $\psi(\cdot)$ be the set of OPRF function received from P_s. $\forall y \in \mathcal{Y}$, P_u computes $v = F_k(\mathcal{H}_1(y))$ and its OPRF function value $\Psi(y)$ and outputs y iff $\Psi(y)$ is meaningful. "$\psi(y)$ is meaningful" means that when the user successfully retrieves information, they obtain comprehensible text, which is what "meaningful" refers to. When the user fails to retrieve information, they receive unintelligible ciphertext, which is considered meaningless in that case.

Fig. 11. PIR based on OPRF.

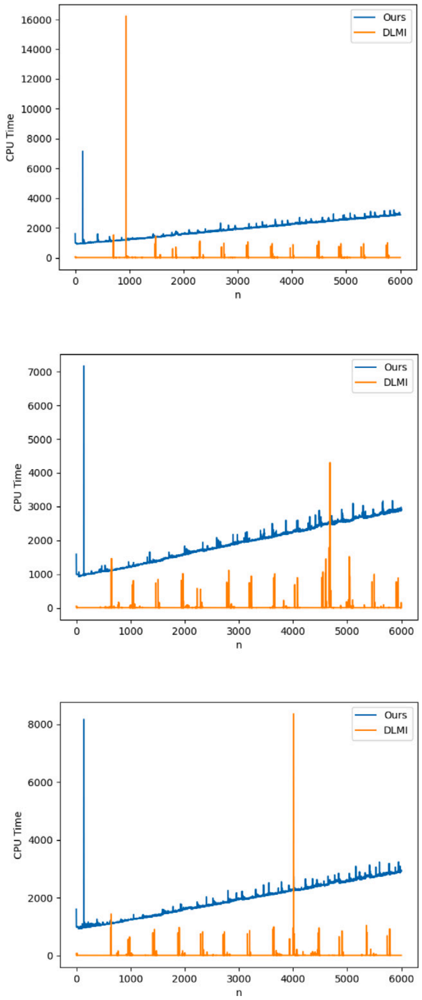

Fig. 12. Parallel comparison of PIR on MAC, where n represents the security parameter, unit is microseconds.

Lemma 7. *Find a suitable pseudorandom function $\widetilde{F}_k : \mathcal{R}_{\{0,1\}} \times \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$. Assuming that the pseudo-random function $F_k : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$ and the hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$ are indistinguishable, we have*

$$\widetilde{F}_k(y) \approx_C F_k(\mathcal{H}_1(y)).$$

Proof. On one hand, because the pseudorandom $\widetilde{F}_k : \mathcal{R}_{\{0,1\}} \times \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$, for any $k \in \mathcal{R}_{\{0,1\}}$, $y \in \mathcal{Y} \subset \{0,1\}^*$, we have $\widetilde{F}_k(y) \approx_C u_\omega \in \mathcal{R}_{\{0,1\}}$.

On the other hand, due to the pseudorandom function $F_k : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$, for $u_{\ell_1} \in \mathcal{R}_{\{0,1\}}$, we have $F_k(u_{\ell_1}) \approx_C u_\omega$. According to the property of the hash function, have $\mathcal{H}_1(y) \approx_C u_{\ell_1}$. Combining with Lemma 6, one can obtain that $F_k(\mathcal{H}_1(y)) \approx_C u_\omega$. Consequently, $\widetilde{F}_k(y) \approx_C F_k(\mathcal{H}_1(y))$. □

Theorem 2. *If $\mathcal{H}_1$ is a collision resistant hash function, $\mathcal{H}_2$ and $\mathcal{H}_3$ are hamming correlation robustness, then the protocol in Fig. 7 securely realizes $\mathcal{F}_{PSI}$ in the semi-honest model when parameters m, w are chosen as described in [14].*

Proof. Perspective from P_1.

Hyb$_0$ P_1's view and P_2's output in the real protocol.

Hyb$_1$ Same as Hyb$_0$ except that on P_2's side, for each $i \in [\omega]$, if $s[i] = 0$, then sample $A_i \leftarrow \{0,1\}^m$ and compute $B_i = A_i \oplus D_i$; otherwise sample $B_i \leftarrow \{0,1\}^m$ and compute $A_i = B_i \oplus D_i$. This hybrid is identical to Hyb$_0$.

Hyb$_2$ Initialize an $m \times w$ binary matrix D to all 1's. Denote its column vectors by $D_1, \dots, D_\omega$. Then $D_1 = \cdots = D_\omega = 1^m$. For $y \in \mathcal{Y}$, randomly select $v \leftarrow [m]^\omega$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$.

Hyb$_3$ Find a suitable pseudorandom function $\widetilde{F}_k : \mathcal{R}_{\{0,1\}} \times \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $\widetilde{v} = \widetilde{F}_k(y)$, randomly select $v \leftarrow [m]^\omega$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$.

Hyb$_4$ Let there be a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$ and a hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $v' = F_k(\mathcal{H}_1(y))$, randomly select $v \leftarrow [m]^\omega$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$.

Hyb$_5$ Let there be a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$, Hamming Correlation Robustness $\mathcal{H}_2 : \mathbb{Z}_{\{0,1\}}^{m \times \omega} \to \mathcal{R}_{\{0,1\}}$ and a hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $v' = F_k(\mathcal{H}_1(y))$, $v = \mathcal{H}_2(v')$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$.

Given that Hyb$_0 \approx_C$ Hyb$_1 \approx_C$ Hyb$_2 \approx_C$ Hyb$_3$, Hyb$_4 \approx_C$ Hyb$_5$ and according to Lemma 7, it be known that Hyb$_3 \approx_C$ Hyb$_4$. Therefore, we have Hyb$_0 \approx_C$ Hyb$_5$.

Perspective from P_2.

Hyb$_0$ P_2's view in the real protocol.

Hyb$_1$ $\psi \leftarrow \mathcal{R}_{\{0,1\}}$, all other aspects are consistent with the real protocol.

Hyb$_2$ Introduce $G_\gamma : \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$ and Hamming Correlation Robustness $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m \times \omega} \to \mathcal{R}_{\{0,1\}}$, let the initial matrices be $C_1 = \cdots = C_\omega = 1^m$, randomly select $v \in [m]^\omega$, set $C_i[v[i]] = 0$ for all $i \in [\omega]$. Compute $G_\gamma(C_1[v[1]] \| \cdots \| C_\omega[v[\omega]])$.

Hyb$_3$ Let the initial matrices be $C_1 = \cdots = C_\omega = 1^m$, find an appropriate pseudorandom function $\widetilde{F}_k : \mathcal{R}_{\{0,1\}} \times \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $\widetilde{v} = \widetilde{F}_k(y)$, randomly select $v \leftarrow [m]^\omega$, set $C_i[v[i]] = 0$ for all $i \in [\omega]$. Compute $G_\gamma(C_1[v[1]] \| \cdots \| C_\omega[v[\omega]])$.

Hyb$_4$ Let the initial matrices be $C_1 = \cdots = C_\omega = 1^m$, set a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$, a hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$ and Hamming Correlation Robustness $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $v' = F_k(\mathcal{H}_1(y))$, randomly select $v \leftarrow [m]^\omega$. Set $C_i[v[i]] = 0$ for all $i \in [\omega]$. Compute $G_\gamma(\mathcal{H}_3(C_1[v[1]] \| \cdots \| C_\omega[v[\omega]]))$.

Hyb$_5$ Let the initial matrices be $C_1 = \cdots = C_\omega = 1^m$, set a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$ and a hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$, Hamming Correlation Robustness $\mathcal{H}_2 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \to \mathcal{R}_{\{0,1\}}$ and $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \to \mathcal{R}_{\{0,1\}}$. For $y \in \mathcal{Y}$, compute $v' = F_k(\mathcal{H}_1(y))$, compute $v' = F_k(\mathcal{H}_1(y))$. Set $C_i[v[i]] = 0$ for all $i \in [\omega]$. Compute $G_\gamma(\mathcal{H}_3(C_1[v[1]] \| \cdots \| C_\omega[v[\omega]]))$.

Similarly, it can be proven that $\text{Hyb}_0 \approx_C \text{Hyb}_5$. □

Definition 16 (*CPA Security Model of the Protocol in Fig. 7*)**.** Assume there exists a perturbed pseudorandom oracle machine $Pr\mathcal{O}M_\gamma$ (where γ is the upper bound on the norm of the perturbation in $Pr\mathcal{O}M_\gamma$), such that for an input x, it outputs two values: one is a random value y_0, and the other is a pseudorandom value y_1 with x as its input.

- **Setup** The simulator $\mathcal{B}$ generates the necessary parameters for the algorithms. The adversary $\mathcal{A}$ chooses s and sends it to the simulator $\mathcal{S}$ using OT.
- **Hash Queries, PRF Queries and PRG Queries** The adversary $\mathcal{A}$ sequentially performs hash function queries, pseudorandom function queries, and pseudorandom synthesizer queries. Here, the adversary cannot know the key in pseudorandom function queries.
- **Challenge** The adversary $\mathcal{A}$ selects a private message m and sends it to the simulator $\mathcal{B}$. The simulator queries the hash function, pseudorandom function, and oblivious transfer values of the real scheme, inputs these results into the pseudorandom oracle machine $Pr\mathcal{O}M_\gamma$, obtains two ciphertexts c_0 and c_1, and sends them to the adversary $\mathcal{A}$.
- **Guessing** After receiving the two ciphertexts c_0 and c_1, $\mathcal{A}$ guesses which ciphertext corresponds to the encryption of m and sends the guess back to the simulator $\mathcal{B}$.

The advantage of the adversary $\mathcal{A}$ is defined as the advantage of the simulator $\mathcal{B}$ in distinguishing the outputs of $Pr\mathcal{O}M_\gamma$.

Note 2. *The $Pr\mathcal{O}M$ mentioned in this paper differs from [22]. In [22], $Pr\mathcal{O}M$ refers to a pseudorandom oracle machine that outputs random values when the adversary does not know the pseudorandom function key, and outputs pseudorandom function values based on the key known to the adversary when the key is known. This is a single-value output. However, the $Pr\mathcal{O}M$ required in this paper outputs both of these values simultaneously, making it a multi-value output.*

Theorem 3. *If $\mathcal{H}_1$ is a collision resistant hash function, $\mathcal{H}_2$ and $\mathcal{H}_3$ are hamming correlation robustness, then the protocol in Fig. 7 securely realizes $\mathcal{F}_{PSI}$ in Definition 16.*

Proof. Suppose the adversary $\mathcal{A}_{P_1}$ can break the scheme with non-negligible advantage. Now, the simulator $\mathcal{S}$ simulates the scheme. Suppose there exists a black-box $G_\gamma^{black-box}$ such that

$$\begin{tikzcd}
 & y_0 = G_\gamma(x) \in \mathcal{R}_{\{0,1\}}, \\
G_\gamma^{black-box}(x) \to (y_0, y_1) \arrow[ur] \arrow[dr] & \\
 & y_1 \in_R \mathcal{R}_{\{0,1\}}.
\end{tikzcd}$$

- **Setup** The simulator $\mathcal{S}$ generates some necessary parameters for the algorithms and selects an appropriate hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathcal{R}_{\{0,1\}}$, Hamming Correlation Robustness $\mathcal{H}_2 : \mathcal{R}_{\{0,1\}} \to [m]^\omega$, Hamming Correlation Robustness $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \to \mathcal{R}_{\{0,1\}}$ and a $G_\gamma : \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$, a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \to \mathcal{R}_{\{0,1\}}$ with key $k \in \mathcal{R}_{\{0,1\}}$. The adversary $\mathcal{A}_{P_1}$ selects s and transmits s to the simulator $\mathcal{S}$ using OT.
- **H-Query, PRF-Query and PRG-Query** The adversary $\mathcal{A}_{P_1}$ makes queries about the hash function, pseudorandom function, oblivious transfer values, and pseudorandom generator. The simulator $\mathcal{S}$ pre-establishes lists for handling H-Query, PRF-Query, and PRG-Query respectively.
  - $\mathcal{H}_1$-Query For the ith query $x_i \in \{0,1\}^*$ corresponding to the value of $\mathcal{H}_1$, the simulator $\mathcal{S}$ selects from the hash value list if available, otherwise selects a random $X_i \in \mathcal{R}_{\{0,1\}}$. Set $X_i = \mathcal{H}_1(x_i)$ and update the list accordingly.
  - $\mathcal{H}_2$-Query For the ith query $y_i \in \mathcal{R}_{\{0,1\}}$ corresponding to the value of $\mathcal{H}_2$, the simulator $\mathcal{S}$ selects from the hash value list if available, otherwise selects a random $Y_i \in [m]^\omega$. Set $Y_i = \mathcal{H}_2(y_i)$ and update the list accordingly.
  - $\mathcal{H}_3$-Query For the ith query $z_i \in \mathbb{Z}_{\{0,1\}}^{m\times\omega}$ corresponding to the value of $\mathcal{H}_3$, the simulator $\mathcal{S}$ selects from the hash value list if available, otherwise selects a random $Z_i \in \mathcal{R}_{\{0,1\}}$. Set $Z_i = \mathcal{H}_3(z_i)$ and update the list accordingly.
  - F-Query For the ith query $u_i \in \mathcal{R}_{\{0,1\}}$ corresponding to the value of F, the simulator $\mathcal{S}$ selects from the pseudorandom function value list if available, otherwise selects a random $U_i \in \mathcal{R}_{\{0,1\}}$. Set $U_i = F(u_i, k)$ and update the list accordingly.
  - G_γ-Query For the ith query $w_i \in \mathcal{R}_{\{0,1\}}$ corresponding to the value of G'_γ, the simulator $\mathcal{S}$ selects from the pseudorandom generator value list if available, otherwise selects a random $W_i \in \mathcal{R}_{\{0,1\}}$. Set $W_i = G'_\gamma(w_i)$ and update the list accordingly. Note that G'_γ is not $G_\gamma^{\text{black-box}}$.
- **Challenge** $\mathcal{A}_{P_1}$ selects $m \in \mathcal{X}/\mathcal{Y}$ and sends it to $\mathcal{S}$. $\mathcal{S}$ using the corresponding hash function queries and pseudorandom function queries, inputs the queried values into the black-box G'_γ, obtaining ψ_0 and ψ_1, and then sends ψ_0, ψ_1 to $\mathcal{A}_{P_1}$.
- **Guess** Based on the received ψ_0 and ψ_1, $\mathcal{A}_{P_1}$ guesses whether ψ_0 or ψ_1 is the ciphertext of the encrypted message m.

According to the assumption, if the adversary $\mathcal{A}_{P_1}$ can break the scheme with a non-negligible advantage, then the simulator $\mathcal{S}$ can also break the black-box G'_γ with a non-negligible advantage. This contradicts the assumption that G'_γ is secure. □

4.4. Efficiency analysis PSI

This section simulates the PSI computation efficiency of this paper and PSI in [14] on MAC, Pad, and Phone. The PRF of [14] is instantiated based on LWE.

4.4.1. Efficiency analysis on MAC

The tools used in the subsection are Python 3.12, the programs are performed on MacBook Air MAC Desktop Apple M1, RAM 8.00 GB (see Fig. 8).

4.4.2. Efficiency analysis on mobile pad

The tools used in the subsection are Pydriod 3, the programs are performed on Xiaomi Pad 6 Pro File Explorer 1th Qualcomm(R)AI Engine(TM) Xiaolong 8+ mobile platform@3.2 GHz, RAM 8.00+3.00 GB (see Fig. 9).

4.5. Analysis of efficiency on mobile phones

The tools used in the subsection are Pydriod 3, the programs are performed on Redmi K30 File Explorer 4th Qualcomm(R)AI Engine(TM) Qualcomm Xiaolong 730G 8+ mobile platform@2.2 GHz, RAM 6.00 GB (see Fig. 10).

4.5.1. Summary of data comparison

From the simulation results, it can be seen that for $n \leq 400$, the LWE-based OPRF in [14] is slightly faster, while for $n > 400$, the ring LPR-based OPRF in this paper is faster. Furthermore, as n increases, the advantages of ring LPR become more pronounced. Based on the simulation results for Pad, the OPRF in this paper is more stable; although there are fluctuations, they are less significant compared to the LWE-based OPRF in [14].

5. Expansion of this work

Private Information Retrieval (PIR) [23–29] is a technique that enables a client to securely download a specific element, such as a movie or a friend's record, from a database managed by an untrusted server, such as a streaming service or a social network, without disclosing to the server which particular element has been retrieved. Given the functional similarities between PIR and PSI, this paper extends its exploration into the construction of PIR using OPRF (see Fig. 11).

5.1. Efficiency analysis PIR

This section simulates the PSI computation efficiency of this paper and machine learning-based PIR in [30](DLMI for short) on MAC. The tools used in the subsection are Python 3.12, the programs are performed on MacBook Air MAC Desktop Apple M1, RAM 8.00 GB.

The OPRF-based PIR proposed in this paper has a runtime that differs from the machine learning-based PIR by no more than approximately 5×10^{-3} seconds. Additionally, the security of our PIR scheme is theoretically supported in comparison to [30] (see Fig. 12).

6. Conclusion

This paper presents a PSI based on efficient post-quantum OPRF and proves its security under the semi-honest model, demonstrating security even in the CPA model in Definition 16. The addition of PPRG enables the PSI to effectively resist probabilistic attacks. In the simulation experiments, the proposed PSI shows greater efficiency compared to post-quantum PSIs represented by LWE.

Although the PIR in this study is not as efficient as the machine learning-based PIR, the gap between the two is already quite small. However, there are also notable shortcomings; the efficiency of the proposed PSI still lags behind that of non-post-quantum PSIs, which will be addressed in future work.

CRediT authorship contribution statement

Zhuang Shan: Writing – original draft, Conceptualization. **Leyou Zhang:** Writing – review & editing, Writing – original draft. **Qing Wu:** Conceptualization. **Qiqi Lai:** Writing – review & editing. **Fuchun Guo:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant 61872087 and Grant 51875457; in part by the Key Foundation of National Natural Science Foundation of China under Grant U19B2021; and in part by the Key Research and Development Program of Shaanxi under Program 2022GY-028 and Program 2022GY-050.

Data availability

No data was used for the research described in the article.

References

[1] R. Lei, X. Chen, D. Liu, C. Song, Y. Tan, A. Ren, CEIU: Consistent and efficient incremental update mechanism for mobile systems on flash storage, J. Syst. Archit. 152 (2024) 103151, http://dx.doi.org/10.1016/j.sysarc.2024.103151, URL: https://www.sciencedirect.com/science/article/pii/S1383762124000882.

[2] J. Sun, L. Yin, M. Zou, Y. Zhang, T. Zhang, J. Zhou, Makespan-minimization workflow scheduling for complex networks with social groups in edge computing, J. Syst. Archit. 108 (2020) 101799, http://dx.doi.org/10.1016/j.sysarc.2020.101799, URL: https://www.sciencedirect.com/science/article/pii/S1383762120300928.

[3] Y. Gao, Y. Luo, L. Wang, X. Liu, L. Qi, W. Wang, M. Zhou, Efficient scalable multi-party private set intersection(-variants) from bicentric zero-sharing, in: Proceedings of the Conference on Computer and Communications Security, CCS, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[4] M.O. Rabin, How to exchange secrets with oblivious transfer, 2005, URL: https://eprint.iacr.org/2005/187.

[5] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions, J. ACM 33 (4) (1986) 792–807, http://dx.doi.org/10.1145/6490.6503.

[6] M. Naor, O. Reingold, Number-theoretic constructions of efficient pseudo-random functions, J. ACM 51 (2) (2004) 231–262, http://dx.doi.org/10.1145/972639.972643.

[7] M.J. Freedman, Y. Ishai, B. Pinkas, O. Reingold, Keyword search and oblivious pseudorandom functions, in: J. Kilian (Ed.), Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 303–324.

[8] S. Jarecki, X. Liu, Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection, in: O. Reingold (Ed.), Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 577–594.

[9] V.K. Yadav, N. Andola, S. Verma, S. Venkatesan, A survey of oblivious transfer protocol, ACM Comput. Surv. 54 (10s) (2022) http://dx.doi.org/10.1145/3503045.

[10] M.R. Albrecht, A. Davidson, A. Deo, N.P. Smart, Round-optimal verifiable oblivious pseudorandom functions from ideal lattices, in: J.A. Garay (Ed.), Public-Key Cryptography – PKC 2021, Springer International Publishing, Cham, 2021, pp. 261–289.

[11] N. Tyagi, S. Celi, T. Ristenpart, N. Sullivan, S. Tessaro, C.A. Wood, A fast and simple partially oblivious PRF, with applications, in: O. Dunkelman, S. Dziembowski (Eds.), Advances in Cryptology – EUROCRYPT 2022, Springer International Publishing, Cham, 2022, pp. 674–705.

[12] S. Casacuberta, J. Hesse, A. Lehmann, Sok: Oblivious pseudorandom functions, in: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), 2022, pp. 625–646, http://dx.doi.org/10.1109/EuroSP53844.2022.00045.

[13] D. Boneh, D. Kogan, K. Woo, Oblivious pseudorandom functions from isogenies, in: S. Moriai, H. Wang (Eds.), Advances in Cryptology – ASIACRYPT 2020, Springer International Publishing, Cham, 2020, pp. 520–550.

[14] M. Chase, P. Miao, Private set intersection in the internet setting from lightweight oblivious PRF, in: D. Micciancio, T. Ristenpart (Eds.), Advances in Cryptology – CRYPTO 2020, Springer International Publishing, Cham, 2020, pp. 34–63.

[15] Z. Shan, L. Zhang, Q. Wu, Q. Lai, Analysis, modify and apply in IIOT form light-weight PSI in CM20, 2024, URL: https://eprint.iacr.org/2024/969.

[16] J. Alwen, S. Krenn, K. Pietrzak, D. Wichs, Learning with rounding, revisited, in: R. Canetti, J.A. Garay (Eds.), Advances in Cryptology – CRYPTO 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 57–74.

[17] A. Banerjee, C. Peikert, A. Rosen, Pseudorandom functions and lattices, in: D. Pointcheval, T. Johansson (Eds.), Advances in Cryptology – EUROCRYPT 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 719–737.

[18] D. Bellizia, C. Hoffmann, D. Kamel, H. Liu, P. Méaux, F.-X. Standaert, Y. Yu, Learning parity with physical noise: Imperfections, reductions and FPGA prototype, IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021 (2021) 390–417, URL: https://api.semanticscholar.org/CorpusID:235814670.

[19] Y. Yu, J. Zhang, Smoothing out binary linear codes and worst-case sub-exponential hardness for LPN, in: T. Malkin, C. Peikert (Eds.), Advances in Cryptology – CRYPTO 2021, Springer International Publishing, Cham, 2021, pp. 473–501.

[20] V. Kolesnikov, R. Kumaresan, M. Rosulek, N. Trieu, Efficient batched oblivious PRF with applications to private set intersection, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 818–829, http://dx.doi.org/10.1145/2976749.2978381.

[21] Z. Brakerski, E. Kirshanova, D. Stehlé, W. Wen, Learning with errors and extrapolated dihedral cosets, in: Public-Key Cryptography – PKC 2018, Springer International Publishing, 2018, pp. 702–727.

[22] A. Jain, H. Lin, J. Luo, D. Wichs, The pseudorandom oracle model and ideal obfuscation, in: H. Handschuh, A. Lysyanskaya (Eds.), Advances in Cryptology – CRYPTO 2023, Springer Nature Switzerland, Cham, 2023, pp. 233–262.

[23] S. Angel, H. Chen, K. Laine, S. Setty, PIR with compressed queries and amortized query processing, in: 2018 IEEE Symposium on Security and Privacy, SP, 2018, pp. 962–979, http://dx.doi.org/10.1109/SP.2018.00062.

[24] A. Burton, S.J. Menon, D.J. Wu, Respire: High-rate PIR for databases with small records, in: Proceedings of the Conference on Computer and Communications Security, CCS, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[25] J. Dujmovic, M. Hajiabadi, Lower-bounds on public-key operations in PIR, in: M. Joye, G. Leander (Eds.), Advances in Cryptology – EUROCRYPT 2024, Springer Nature Switzerland, Cham, 2024, pp. 65–87.

[26] B. Fisch, A. Lazzaretti, Z. Liu, C. Papamanthou, Thorpir: Single server PIR via homomorphic thorp shuffles, in: Proceedings of the Conference on Computer and Communications Security, CCS, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[27] A. Gascon, Y. Ishai, M. Kelkar, B. Li, Y. Ma, M. Raykova, Computationally secure private information retrieval and aggregation in the shuffle model, in: Proceedings of the Conference on Computer and Communications Security, CCS, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[28] A. Ghoshal, M. Zhou, E. Shi, Efficient pre-processing PIR without public-key cryptography, in: M. Joye, G. Leander (Eds.), Advances in Cryptology – EUROCRYPT 2024, Springer Nature Switzerland, Cham, 2024, pp. 210–240.

[29] M. Luo, F.-H. Liu, H. Wang, Faster FHE-based single-server private information retrieval, in: Proceedings of the Conference on Computer and Communications Security, CCS, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[30] M. Lam, J. Johnson, W. Xiong, K. Maeng, U. Gupta, Y. Li, L. Lai, I. Leontiadis, M. Rhu, H.-H.S. Lee, V.J. Reddi, G.-Y. Wei, D. Brooks, E. Suh, GPU-based private information retrieval for on-device machine learning inference, in: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1, ASPLOS '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 197–214, http://dx.doi.org/10.1145/3617232.3624855.

Zhuang Shan received the B.S. from Liaoning Institute of Science and Technology, benxi, China, in 2019. And he received the M.S. from North Minzu University, yinchuan, China, in 2022.

He is currently pursuing the Ph,D. degree in mathematics with Xidian University, Xi'an, China. His current interests include cryptography, reduction of hard problems in lattice, and network security.

Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2002 and 2009, respectively. From 2013 to 2014, he served as a visiting scholar at the University of Wollongong, Australia. He currently worked in Xidian University as a professor.

His current research interests include public key cryptography, network security and computer security. He has over 120 scientific publications in many highly ranked cybersecurity journals and conferences.

Qing Wu received the M.S. and Ph.D. degrees from the Xidian University, Xi'an, China, in 2006 and 2009, respectively.

She currently works with Xi'an University of Posts and Communications, Xi'an, as a Professor. Her current research interests include artificial intelligence security and cloud security.

Qiqi Lai received the B.S. from PLA University of Information Engineering, henan, China, in 2008. And he received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2011 and 2015.

His currently works with Shaanxi Normal University, Xi'an, as a Professor. His current research interests include the theory of lattice-based public key cryptography and its provable security, as well as the construction and analysis of homomorphic encryption schemes.

Funcun Guo received the B.S. and M.S. degrees from Fujian Normal University, China, in 2005 and 2008, respectively, and the Ph.D. degree from the University of Wollongong, Australia, in 2013. He is currently an Associate Research Fellow with the School of Computing and Information Technology, University of Wollongong.

His primary research interests include the public key cryptography, in particular protocols, encryption and signature schemes, and security proof.