Oblivious Pseudo Random Function base on Ideal Lattice, Application in PSI and PIR

Zhuang Shan(单壮)[1], Leyou Zhang(张乐友)[1,*], Qing Wu(吴青)[2], Qiqi Lai(来齐齐)[3], Fuchun Guo(郭福春)[4]

August 28, 2024

Abstract

Privacy set intersection (PSI) and private information retrieval (PIR) are important areas of research in privacy protection technology. One of the key tools for both is the oblivious pseudorandom function (OPRF). Currently, existing oblivious pseudorandom functions either focus solely on efficiency without considering quantum attacks, or are too complex, resulting in low efficiency. The aim of this paper is to achieve a balance: to ensure that the oblivious pseudorandom function can withstand quantum attacks while simplifying its structure as much as possible. This paper constructs an efficient oblivious pseudorandom function based on the ideal lattice hardness assumption and the oblivious transfer (OT) technique by Chase and Miao (CRYPTO 2020), and also constructs PSI and PIR.
Keywords: OPRF; PSI; PIR.

1 Introduction

An oblivious transfer [Rab05] 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

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China; arcsec30@163.com

[2] School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

[3] School of Computer Science, Shaanxi Normal University, Xi'an 710062, China

[4] Centre for Computer and Information Security Research, University of Wollongong, Wollongong, NSW 2522, Australia

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 [GGM86]. Two decades later, Naor and Reingold [NR04] 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 [FIPR05]. In 2009, Jarecki and Liu presented an efficient OPRF for securing intersection data [JL09].

Oblivious pseudorandom functions have been utilized in two critical applications: private set intersection (PSI) and private information retrieval (PIR) [YAVV22, DH24, GZS24]. The additional functionalities of oblivious pseudorandom functions also exhibit diversity, such as Verifiable Oblivious Pseudorandom Functions (VOPRF, [ADDS21]) and Partially Oblivious Pseudorandom Functions (POPRF, [TCR$^+$22]).

Currently, OPRFs still have a long way to go, as summarized by Casacuberta, Hesse, and Lehmann [CHL22]. 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, with applications in PSI and PIR.

2 Our works

Regarding the open problem proposed by Casacuberta, there are currently quantum-resistant OPRFs, namely Albrecht et al.'s lattice-based VOPRF [ADDS21] and Boneh et al.'s isogeny-based OPRF [BKW20]. Both constructions represent significant feasibility results but require further research to improve their efficiency [CHL22].

We adopted Chase and Miao's [CM20] 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.

As an application of this work, we constructed private set intersection (PSI) and private information retrieval (PIR) based on Chase and Miao's ideas. Since [SZWL24] analyzed that Chase and Miao's protocol 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 perturbed pseudorandom generator (PPRG) as given in [SZWL24].

3 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 $(v_1, \ldots, v_n)$ in the lattice $\mathcal{L}$, then the fundamental parallelelepiped is

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

If the lattice base $(v_1, \ldots, v_n)$ is determined, use the symbol $\mathcal{P}(\mathcal{L})$ to replace $\mathcal{P}(v_1, \ldots, v_n)$. $\forall x \in \mathbb{R}^n$, 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 $(v_1, \ldots, v_n)$. 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 $\mathcal{L}$, $(v_1, \ldots, v_n)$ and $(u_1, \ldots, u_n)$ 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 $u_i = \sum_{j=1}^{n} m'_{ij} v_j, i \in \{1, \ldots, n\}$, 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 $\det(M)\det(M') = 1$ and $\det(M) = \det(M') = \pm 1$, so

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

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, [BPR12, AKPW13])**.** *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: $(A, \lfloor As \rfloor_p) \approx_C (A, \lfloor u \rfloor_p)$.*

Definition 4 (Learning parity with noise, [YZ21, BHK+21]). *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, [CM20]). *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, [Net]). *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, [KKRT16]). *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*

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

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 $OPRF_k(x)$ is pseudorandom assuming the hash function $\mathcal{H}$ is correlation robust.

Definition 8 (PSI, [CM20]). *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 (PIR, [ACLS18]). *PIR allows a client to download an element (e.g., movie, friend record) from a database held by an untrusted server (e.g., streaming service, social network) without revealing to the server which element was downloaded.*

4 Ring-LPR based OPRF

Definition 10 (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 11 (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 1. *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:

Algorithm 1 Oblivious Pseudorandom Function (OPRF)

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

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 $s[1], \ldots, s[w]$. 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, $C_i = r_i \oplus \Delta_i$.

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

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

Here's the translation of the provided lemma and proof into English:

Lemma 2. *If 2^n-LWR is hard, then 2-LWR is also hard.*

Proof. Let $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$ and $s \in \mathbb{Z}_{\{0,1\}}^n$. Suppose there exists an efficient algorithm $\mathcal{W}$ that can recover s from $b = \lfloor As \rfloor_1$ in polynomial time. For $A' \in \mathbb{Z}_{2^2}^{m\times n}$, we have $A' = A'_1 + 2A'_2$. Thus, we get

$$b'_1 + 2b'_2 = \lfloor A'_1 s'_1 \rfloor_1 + 2\lfloor A'_2 s'_2 \rfloor_1 = \frac{A'_1 s'_1}{2} + 2 \cdot \frac{A'_2 s'_2}{2} + e \ (e \in (-1, 0]^m).$$

Hence, using $\mathcal{W}$ twice, we can solve 2^2-LWR. Repeating this process, we can solve 2^n-LWR using n applications of $\mathcal{W}$. Therefore, we have

$$nO(\mathcal{W}) \geq O(n!) \text{ or } O(e^n).$$

Thus,

$$O(\mathcal{W}) \geq \frac{O(n!)}{n} \text{ or } \frac{O(e^n)}{n}.$$

This contradicts the assumption that there exists an efficient algorithm $\mathcal{W}$ that can recover s from $u = \lfloor As \rfloor_1$ in polynomial time. Hence, the lemma is proved. $\square$

Lemma 3. *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. Similarly, from $m-1$ instances of the inner product Ring-LPR, we obtain $\alpha_2, \ldots, \alpha_m$, and let

$$\Lambda = (\alpha_1, \alpha_2, \ldots, \alpha_m), \beta = (b_1, b_2, \ldots, b_m).$$

Thus,

$$\beta = \lfloor \Lambda s \rfloor_1. \tag{4.1}$$

Assuming that the time complexity of solving s from equation (4.1) is $O(\Lambda, \beta)$, according to Lemma 2, we have

$$mO(\mathcal{W}) \geq O(\Lambda, \beta) \geq O(n!) \text{ or } O(e^n)$$

Let $m = n$, then

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

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. $\square$

5 Ring-LPN Based PRG

5.1 Proof of Quantum Resistance for LPN

Definition 12 (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*

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

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.*

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

Lemma 4. *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

$$\begin{aligned}|0\rangle|x_i\rangle + |1\rangle|(x_i + s) \bmod 4\rangle &= |0\rangle|x_i'\rangle + |1\rangle|(x_i' + s') \bmod 2\rangle \\ &\quad + 2(|0\rangle|x_i''\rangle + |1\rangle|(x_i' + s'') \bmod 2), i \leq \ell,\end{aligned}$$

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 13 (Extrapolated Dihedral Coset Problem with model 2, [BKSW18])**.** *Given a security parameter κ, an instance of $EDCP_{n,2,\rho}^\ell$ 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*

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

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 poly$(1/(n \log 2))$ in time poly$(n \log 2)$, then the $EDCP_{n,2,\rho}^\ell$ problem is considered to be broken.

Lemma 5. *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 6 ([BKSW18])**.** *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-$EDCP^{\ell}_{n,q,\rho_r}$.*

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

5.2 Ring-LPN

Definition 14 (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 3. *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 3, 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. □

5.3 Perturbed Pseudorandom Generator

Definition 15. *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*

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

Definition 16 ([SZWL24])**.** *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 $x_1 = x_2$, $\Pr(G_\gamma(x_1) = G_\gamma(x_2)) \leq \exp(-\Omega(n))$,*
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.*

Setup Let $a, x, e \in \mathcal{R}_{\{0,1\}}$. **Enc** Compute $$G_\gamma(x) = ax + e \bmod (x^n + 1) \bmod 2.$$

Figure 1: Pseudorandom generator with perturbation $G_\gamma(\cdot)$

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

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

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

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

Thus, there is a very high probability that $\|v_1 - v_2\| \geq \sqrt{n+1}$, and $N = 1$. $\square$

6 Construct PSI and PIR based on OPRF

6.1 PSI based on OPRF

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 $\psi \in \Psi$.

Figure 2: PSI based on OPRF

Lemma 7. *Assuming $f(y) \approx_C u_1$ and $g(u_1) \approx_C u_2$, then $(g \circ f)(y) \approx_C u_2$.*

Lemma 8. *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 7, 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.2 securely realizes $\mathcal{F}_{PSI}$ in the semi-honest model when parameters m, w are chosen as described in [CM20].*

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 $D_1, \ldots, D_\omega$. Then $D_1 = \ldots = D_\omega = 1^m$. 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 $\text{Hyb}_0 \approx_C \text{Hyb}_1 \approx_C \text{Hyb}_2 \approx_C \text{Hyb}_3$, $\text{Hyb}_4 \approx_C \text{Hyb}_5$ and according to Lemma 8, it be known that $\text{Hyb}_3 \approx_C \text{Hyb}_4$. Therefore, we have $\text{Hyb}_0 \approx_C \text{Hyb}_5$.

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 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 $\text{Hyb}_0 \approx_C \text{Hyb}_5$. $\square$

Definition 17 (CPA security model of the protocol in Fig.2)**.** *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.*
- **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 [JLLW23]. In [JLLW23], $Pr\mathcal{O}M$ 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.2 securely realizes $\mathcal{F}_{PSI}$ in the definition 17.*

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

$$
G_\gamma^{black-box}(x) \to (y_0, y_1) \begin{array}{l} \nearrow \quad y_0 = G_\gamma(x) \in \mathcal{R}_{\{0,1\}}, \\ \searrow \quad y_1 \in_R \mathcal{R}_{\{0,1\}}. \end{array}
$$

- **Setup** The simulator $\mathcal{S}$ generates some necessary parameters for the algorithms and selects an appropriate hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathcal{R}_{\{0,1\}}$, Hamming Correlation Robustness $\mathcal{H}_2 : \mathcal{R}_{\{0,1\}} \rightarrow [m]^\omega$, Hamming Correlation Robustness $\mathcal{H}_3 : \mathbb{Z}_{\{0,1\}}^{m\times\omega} \rightarrow \mathcal{R}_{\{0,1\}}$ and a G_γ : $\mathcal{R}_{\{0,1\}} \rightarrow \mathcal{R}_{\{0,1\}}$, a pseudorandom function $F : \mathcal{R}_{\{0,1\}} \times \mathcal{R}_{\{0,1\}} \rightarrow \mathcal{R}_{\{0,1\}}$ 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 i^{th} 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 i^{th} 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 i^{th} 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 i^{th} 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 i^{th} 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. □

6.2 PIR based on OPRF

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.

Figure 3: PIR based on OPRF

Theorem 4. *If $\mathcal{H}_1$ is a collision resistant hash function, $\mathcal{H}_2$ and $\mathcal{H}_3$ are hamming correlation robustness and $\Psi(x)$ is one way function, then the protocol in Fig.3 securely realizes $\mathcal{F}_{PIR}$ in the semi-honest model when parameters m, ω are chosen as security parameters.*

Proof. The proof process is similar to that of Theorem 2. □

Remark 1. *The PIR protocol in Fig.3 cannot withstand malicious users unless the function $\psi(v)$ has additional security definitions, at least ensuring that the output of $\psi(v)$ is pseudorandom when $y \notin \mathcal{X}$.*

References

[ACLS18] Sebastian Angel, Hao Chen, Kim Laine, and Srinath Setty. Pir with compressed queries and amortized query processing. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 962–979, 2018.

[ADDS21] Martin R. Albrecht, Alex Davidson, Amit Deo, and Nigel P. Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In Juan A. Garay, editor, *Public-Key Cryptography – PKC 2021*, pages 261–289, Cham, 2021. Springer International Publishing.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 57–74, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[BHK$^+$21] Davide Bellizia, Clément Hoffmann, Dina Kamel, Hanlin Liu, Pierrick Méaux, François-Xavier Standaert, and Yu Yu. Learning parity with physical noise: Imperfections, reductions and fpga prototype. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021:390–417, 2021.

[BKSW18] Zvika Brakerski, Elena Kirshanova, Damien Stehlé, and Weiqiang Wen. Learning with errors and extrapolated dihedral cosets. In *Public-Key Cryptography – PKC 2018*, pages 702–727. Springer International Publishing, 2018.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 520–550, Cham, 2020. Springer International Publishing.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 719–737, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[CHL22] Sílvia Casacuberta, Julia Hesse, and Anja Lehmann. Sok: Oblivious pseudorandom functions. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 625–646, 2022.

[CM20] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious prf. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 34–63, Cham, 2020. Springer International Publishing.

[DH24] Jesko Dujmovic and Mohammad Hajiabadi. Lower-bounds on public-key operations in pir. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 65–87, Cham, 2024. Springer Nature Switzerland.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian, editor, *Theory of Cryptography*, pages 303–324, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792 – 807, aug 1986.

[GZS24] Ashrujit Ghoshal, Mingxun Zhou, and Elaine Shi. Efficient pre-processing pir without public-key cryptography. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 210–240, Cham, 2024. Springer Nature Switzerland.

[JL09] Stanisław Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In Omer Reingold, editor, *Theory of Cryptography*, pages 577–594, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[JLLW23] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. The pseudorandom oracle model and ideal obfuscation. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 233–262, Cham, 2023. Springer Nature Switzerland.

[KKRT16] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni 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, page 818 - 829, New York, NY, USA, 2016. Association for Computing Machinery.

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

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudorandom functions. *Journal of the ACM*, 51(2):231 - 262, mar 2004.

[Rab05] Michael O. Rabin. How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Paper 2005/187, 2005.

[SZWL24] Zhuang Shan, Leyou Zhang, Qing Wu, and Qiqi Lai. Analysis, modify and apply in IIOT form light-weight PSI in CM20. Cryptology ePrint Archive, Paper 2024/969, 2024.

[TCR+22] Nirvan Tyagi, Sofía Celi, Thomas Ristenpart, Nick Sullivan, Stefano Tessaro, and Christopher A. Wood. A fast and simple partially oblivious prf, with applications. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 674–705, Cham, 2022. Springer International Publishing.

[YAVV22] Vijay Kumar Yadav, Nitish Andola, Shekhar Verma, and S. Venkatesan. A survey of oblivious transfer protocol. *ACM Computing Surveys*, 54(10s), sep 2022.

[YZ21] Yu Yu and Jiang Zhang. Smoothing out binary linear codes and worst-case subexponential hardness for lpn. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 473–501, Cham, 2021. Springer International Publishing.