Probabilistic Attacks and Enhanced Security for "Private Set Intersection in the Internet Setting from Lightweight Oblivious PRF"

Zhuang Shan
arcsec30@163.com
Xidian University
Xi'an, Shaanxi, China

Leyou Zhang*
lyzhang@mail.xidian.edu.cn
Xidian University
Xi'an, Shaanxi, China

Qiqi Lai
lyzhang@mail.xidian.edu.cn
Shaanxi Normal University
Xi'an, Shaanxi, China

Abstract

Privacy Set Intersection (PSI) has been an important research topic within privacy computation. Its main function is to allow two parties to compute the intersection of their private sets without revealing any other private information. Therefore, PSI can be applied to various real-world scenarios.

Chase and Miao presented an impressive construction "Private set intersection in the Internet setting from lightweight oblivious prf" (CM20 for short) at Crypto 2020, highlighting its convenient structure and optimal communication cost. However, it does have some security vulnerabilities. Let X be the privacy set of user P_1, Y be the privacy set of user P_2. The CM20 protocol uses a pseudorandom function (PRF) to encrypt the privacy $x \in X$ of P_1 into D_1 and the privacy $y \in Y$ of P_2 into D_2, $D_1 = D_2$ as $x = y$. It then sends random data F_1 to user P_1 and random data F_2 to user P_2 using a random oblivious transfer technique. User P_2 computes $\delta = D_2 \oplus F_2$ and sends δ to user P_1, and user P_1 uses δ and F_1 to re-encrypt D_1. Repeat this until P_1 re-encrypts all the privacy in all the privacy sets X, packages them up and sends them to P_2, who completes the privacy set intersection. However, if an adversary obtains δ and F_2 by any means, they can recover the PRF's encryption of the user's privacy, and the recovery process is non-trivial. This significantly weakens the security of the CM20 protocol.

In this paper, we make three main contributions. First, based on the above analysis, we present a method for attacking CM20, called *probabilistic attacks*. This attack is based on estimating and analysing the frequency distribution of the encrypted data from the PRF and the probability distribution of the original private data, and determining the relationship between the two. Although not 100% effective, this method of attack poses a significant threat to the security of user data.

Secondly, we introduce a new tool called the *perturbed pseudorandom generator* (PPRG). We show that the PPRG can overcome probabilistic attacks by replacing the random oblivious transfer and one of the hash functions (originally there were two) in CM20.

Finally, we provide a dedicated indistinguishability against chosen-plaintext attack (IND-CPA) security model for this PSI protocol. The efficiency analysis shows that the proposed PSI is comparable to CM20's PSI, whether on a PC, MAC, pad or mobile phone.

CCS Concepts

• **Security and privacy** → **Security protocols**.

Keywords

PSI, Probabilistic attacks, Pseudorandom generator

ACM Reference Format:
Zhuang Shan, Leyou Zhang, and Qiqi Lai. 2024. Probabilistic Attacks and Enhanced Security for "Private Set Intersection in the Internet Setting from Lightweight Oblivious PRF". In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Private Set Intersection (PSI) allows two parties, Alice (with a private input set X) and Bob (with a private input set Y), to learn the intersection $X \cap Y$ of their sets, while ensuring that neither party learns anything beyond the intersection. All standard PSI protocols have communication and computation complexity proportional to the cardinality of the input sets. [15, 16, 20, 25, 35, 38, 45].

At Crypto 2020, a lightweight PSI was presented by Chase and Miao, entitled as "Private Set Intersection in the Internet Setting of Lightweight Oblivious PRF", referred to as CM20 in this paper [20]. CM20 is a very classic PSI article, the main structure is the use of hash functions and pseudorandom functions to encrypt all the private data of Alice (the user), and then send the encrypted data to Bob (another user) through the oblivious transfer (OT [7, 40, 41]), Bob will also encrypt his own private data using hash functions and pseudorandom functions, and then with the encrypt data that Alice sends. Bob also encrypts his private data with a hash function and a pseudorandom function, and then compares it with the encrypted data sent by Alice, and the successful match (the encryption result is the same) is the intersection of the two parties' privacy sets. CM20 achieves a better balance between computation and communication compared to the previous results.

However, CM20 pursues the balance between computation and communication too much and neglects the security issue. When an adversary intercepts the data during the oblivious transfer phase, it is possible to recover the pseudo-random encrypted value of the protocol for the user's privacy. The pseudorandom function is a weak encryption [20, 23], and gets the same output (i.e., pseudorandom function value) for the same input. Based on this property, we give an attack with the name *probabilistic attacks*. The meaning of the probabilistic attacks is that, if the adversary has access to the private portion of the data in some way, then the adversary can statistically intercept the pseudorandom portion of the function values. If the difference between the two proportions is not large, the adversary can think that these pseudorandom function values are the result of encrypting the private data by pseudorandom function in CM20 protocol, although this attack means is not 100% effective, but this has threatened the user's data security.

There are three main contributions in this paper. First, we present the attack method for CM20, namely the *probabilistic attacks*, and simulate the attack on CM20. This attack is not only applicable to CM20, but can also be used against other protocols with weak encryption.

Second, we introduce the definition of a *perturbed pseudorandom generator* (PPRG). We replace the random oblivious and one of the hash functions (originally there were two) in the original CM20 scheme with a perturbed pseudorandom generator. The PPRG strengthens the security of the CM20 protocol, making it strongly secure, i.e. achieving randomised encryption. Randomised encryption means that for the same private data, the encryption results are different, effectively resisting probabilistic attacks.

Finally, we define a new security model, the indistinguishability against chosen-plaintext attack (IND-CPA) security model for PSI, and prove that our PSI protocol is IND-CPA security model. This model gives the adversary the ability to intercept the encrypted data of others, which allows him to perform probabilistic attacks.

1.1 Our Contributions

1.1.1 Probabilistic Attacks. The core idea of probabilistic attacks is to guess and attack the data by analysing the frequency distribution of the pseudorandomly encrypted data in CM20 (encrypted data for short) and the probability distribution of the original private data. First, we assume that there is an encrypted data set Y, where each data item $y \in Y$ is obtained by encrypting a private data item x using the CM20 protocol. We then count the frequency of these encrypted items within the dataset, thus obtaining the probability of occurrence for each encrypted item.

For each piece of encrypted data y, we need to guess the corresponding original private data x, and this guess depends on the probability distribution $\Pr(x)$ of the original private data and the probability of occurrence of the encrypted data.

The specific guessing method is to compute the difference in probability distributions between the encrypted data y and each possible original private data item x. Common ways to compute this difference include Euclidean distance or KL divergence. The x with the smallest difference is chosen as the corresponding original private data, assuming that it is the most likely original private data for the encrypted data y. We then encrypt the guessed original private data and compare it with the actual encrypted data. If the encrypted guess matches the actual encrypted data, the attack is considered successful.

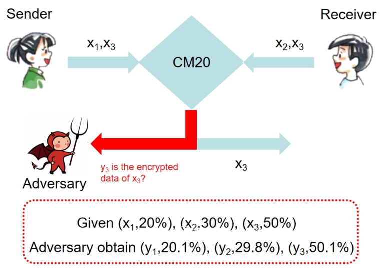

Sender
Receiver
x_1,x_3
CM20
x_2,x_3
Adversary
y_3 is the encrypted data of x_3?
x_3
Given (x_1,20%), (x_2,30%), (x_3,50%)
Adversary obtain (y_1,20.1%), (y_2,29.8%), (y_3,50.1%)

Figure 1: Probabilistic attacks

The effectiveness of the probabilistic attacks is evaluated by the accuracy, which is calculated as the ratio of the number of successful guesses to n. If the guessed original private data x matches the actual encrypted data y, it indicates that the attack method is effective or the guess is successful. In this case, the number of successful guesses is increased by 1.

Algorithm 1 Probabilistic Attacks

```
1: Input: Known probabilities $\Pr(x)$ for $n$ private data items $x$ and $l$ data sets $y$ (where $n$ and $l$ are variables, we adjust $n$ and $k$ to observe the effect of probabilistic attacks, with $y = \mathrm{PRF}(x)$)
2: Output: The original private data $x$ of encrypted data $y$
3: Step 1: Count the probability of occurrence of each encrypted item $y$.
4:     Calculate the frequency of each $y$, denoted as $f(y)$
5:     Estimate the probability of $y$ as $\Pr(y) = \frac{f(y)}{k}$
6: Step 2: Guess the original private data $x$ corresponding to encrypted data $y$, i.e., $y = \mathrm{PRF}(x)$
7:     For each $\Pr(y)$, find the closest $\Pr(x)$ such that $\Pr(x) \approx \Pr(y)$
8:     Use the distance minimization method to find original private data $x$ corresponding to encrypted data $y$
```

Remark 1. *The table 1 shows that the success rate of the probabilistic attacks depends on to both n and l. Although it may seem low, it is still non-negligible, which does not meet the security requirements of cryptographic protocol.*

Table 1: Accuracy of Probabilistic Attacks

	$l = 10^4$	$l = 10^5$	$l = 10^6$	$l = 10^7$
$n = 100$	0.11	0.35	0.6	0.79
$n = 500$	0.006	0.032	0.124	0.306
$n = 1000$	0.003	0.007	0.036	0.113
$n = 2000$	0.002	0.0055	0.0115	0.04

1.1.2 Perturbed Pseudorandom Generator. The term "perturbed pseudorandom generator" refers to perturbing the output of a pseudorandom generator. A pseudorandom generator helps to hide private information, but it is still a weak form of encryption. By adding a perturbation, it can be shown to resist probabilistic attacks.

Setup. Let $A \in \mathbb{Z}_{\{0,1\}}^{m \times n}$, $x \in \mathbb{Z}_{\{0,1\}}^{n}$, $e \in \mathbb{Z}_{\{0,1\}}^{m}$.
Enc. Compute

$$G_\gamma(x) = (Ax) \bmod 3 + e.$$

Figure 2: Perturbed pseudorandom generator

We replace the second hash function in the CM20 protocol with a perturbed pseudorandom generator. The purpose of this modification is to ensure that our protocol (the modified CM20 protocol) produces different encryption results for the same input, while also ensuring that both parties can correctly match their privacy intersections. As a result, the matching mechanism in our protocol differs from the one in CM20. CM20 performs exact matching, while our protocol relies on the ℓ_2-norm and a threshold for matching, due to the perturbation introduced by the pseudorandom generator.

Regarding the security or pseudorandomness of the perturbed pseudorandom generator, we reduce it to the dihedral coset problem, which is a known problem resistant to quantum attacks. A detailed proof will be given shortly.

1.1.3 IND-CPA Security Model for Our Protocol. The current security models for Private Set Intersection (PSI) are the semi-honest user model and the malicious user model. In the semi-honest model, all participants are assumed to be either semi-honest or honest. A semi-honest participant fully complies with the protocol, but collects all records during protocol execution and attempts to infer the inputs of other participants. Attackers in the semi-honest model are passive, i.e. they do not actively interfere with the protocol, but instead try to learn additional information from the protocol execution. In contrast, the malicious model assumes the presence of at least one malicious attacker who may attempt to disrupt protocol execution or steal private information from other participants. Our proposal is for a new model of security, one that is

Definition 1 (IND-CPA security model of the our protocol)**.** *Assume there exists a perturbed pseudorandom oracle $Pr\mathcal{O}M_\gamma$ (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.*
- **PRF Queries** *The adversary $\mathcal{A}$ performs 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 $Pr\mathcal{O}M_\gamma$.

This model gives the adversary the ability to intercept the encrypted data of others, which allows him to perform probabilistic attacks. Our IND-CPA security model ultimately reduces to $Pr\mathcal{O}M_\gamma$, which means that the perturbed pseudorandom generator is the core security component of our protocol. For $Pr\mathcal{O}M_\gamma$ we give the following definition.

Definition 2 (Perturbed pseudorandom oracle)**.** *For the query on x, two values are output, i.e.,*

$$\begin{tikzcd} & y_0 = G_\gamma^{\textit{black-box}}(x) \in \mathbb{Z}_4^{\ell_2}, \\ Pr\mathcal{O}M_\gamma(x) \to (y_0, y_1) \arrow[ur] \arrow[dr] & \\ & y_1 \in_R \mathbb{Z}_4^{\ell_2}. \end{tikzcd}$$

Here, $G_\gamma^{\textit{black-box}}(x) \in \mathbb{Z}_4^{\ell_2}$ is a black-box construction of a PPRG, y_0 and y_1 are indistinguishable. When the inputs x_1 and x_2 are equal, i.e., $x_1 = x_2$, we have

$$Pr\mathcal{O}M_\gamma(x_1) = (y_0^{x_1}, y_1^{x_1}),$$
$$Pr\mathcal{O}M_\gamma(x_2) = (y_0^{x_2}, y_1^{x_2}),$$

and it holds that $\|y_0^{x_1} - y_0^{x_2}\| < \gamma$, $\|y_1^{x_1} - y_1^{x_2}\| < \gamma$.

Note 1. *The $Pr\mathcal{O}M$ mentioned in this paper is different from [28]. In [28], $Pr\mathcal{O}M$ refers to a pseudorandom oracle 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.*

1.2 Related Work

Early solutions for 2-party PSI [25, 33] were primarily based on Diffie-Hellman and were secure against semi-honest adversaries. In recent years, a variety of new protocol paradigms have emerged, leveraging techniques such as key agreement [12, 26, 29], Bloom filters [13, 43], oblivious polynomial evaluation via additively homomorphic encryption [10, 30], circuit-based approaches [15, 24, 37, 38], and vector oblivious linear function evaluation [45], among others.

However, the OT paradigm is the primary basis for truly practical and scalable solutions for PSI. Their efficiency comes from OT extension [27], which minimises each OT instance's marginal cost to low-cost symmetric operations like AES calls. These OT instances facilitate the comparisons required for PSI. Pinkas, Schneider and Zohner [39] were the first to propose PSI based directly on OT. This approach has subsequently been the subject of a number of refinements and extensions [16, 20, 31, 34–36, 44, 45].

Structure-Aware PSI. Recently, Garimella et al. [17] introduced the concept of structure-aware PSI, where Alice holds a structured input, and Bob has an unstructured set of points. In this approach, the communication cost of the protocol scales with the size of the description of the structured set, rather than its cardinality. It's important to note that silent OT [2–4, 8, 47], which enables parties to generate an essentially unlimited number of oblivious transfer instances without communication, does not address the challenge of structure-aware PSI. This is because silent OT only generates random OT correlations, which still require conversion into chosen-input OT instances via communication [1], and this communication cost is proportional to the cardinality of the sets involved.

Malicious Model. To the best of our knowledge, the first specialised protocol for PSI in the malicious setting with less than quadratic complexity was introduced in [14]. Other approaches to malicious 2-party PSI have used techniques such as Diffie-Hellman key agreement, oblivious linear function evaluation and homomorphic encryption (e.g., [9, 11, 19, 22, 29]). More recent works [16, 18, 35, 43–46] have incorporated OT extension to achieve security in the malicious model.

1.3 Organizations

The structure of this paper is as follows. Section 2 provides the necessary definitions and lemmas as a basis for the reader's understanding. Section 3 describes in detail the execution process of the CM20 protocol, the recovery of pseudorandom function values using random oblivious transfer data, and discusses probabilistic attacks. Section 4 introduces the definition, construction, and security and correctness proof of the perturbed pseudorandom generator. Section 5 gives our PSI protocol, i.e. *private set intersection from the perturbed pseudorandom generator*, and the corresponding proof of the chosen security model for plaintext attacks. Section 6 shows the performance evaluation of our PSI protocol. Section 7 concludes the paper with a summary and future directions.

2 Preliminary

Definition 3 ([21]). *$\varepsilon(n)$ is negligible associated with n if $\varepsilon(n)$ can be expressed as*

$$\varepsilon(n) = \frac{1}{O(2^n)},$$

and the notation $O(n)$ represents a quantity that grows at most as fast as n approaches infinity.

Definition 4 ([6]). *Let $\mathcal{H}$ be a Hilbert space, and let $T : \mathcal{H} \to \mathcal{H}$ be an operator. If $T(\cdot)$ satisfies*

$$\|Tx - Ty\| < \|x - y\|, \ \forall x, \ y \in \mathcal{H},$$

then $T(\cdot)$ is called a contraction operator.

Lemma 1 ([6]). *If $\mathcal{H}$ is a closed set (every Cauchy sequence in $\mathcal{H}$ converges to a point within $\mathcal{H}$), and $T(\cdot)$ is a contraction operator, and $Fix(T)$ is a closed convex set, then the algorithm $x_{n+1} = Tx_n$ converges to some $x \in Fix(T)$, where $Fix(T)$ denotes the set of fixed points of the operator $T(\cdot)$.*

Remark 2. *The convergence mentioned in Lemma 2 should be considered as strong convergence. However, this paper does not discuss the difference between strong and weak convergence, because in finite dimensions strong and weak convergence are equivalent.*

Lemma 2 ([6]). *If $\mathcal{H}$ is a closed set (every Cauchy sequence in $\mathcal{H}$ converges to a point within $\mathcal{H}$), and $T(\cdot)$ is a contraction operator, and $Fix(T)$ is a closed convex set, then the algorithm $x_{n+1} = Tx_n$ converges to some $x \in Fix(T)$, where $Fix(T)$ denotes the set of fixed points of the operator $T(\cdot)$.*

Remark 3. *The convergence mentioned in Lemma 2 should be considered as strong convergence. However, this paper does not discuss the difference between strong and weak convergence, because in finite dimensions strong and weak convergence are equivalent.*

Definition 5 (Dihedral Coset Problem). *Given a security parameter κ, an instance DCP_q^ℓ is defined with N as the modulus and ℓ as the number of states. Each state is represented as*

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

where s and x are chosen randomly from $\mathbb{Z}_q$, and s needs to be determined with probability at least $poly(1/\log q)$ in $poly(\log q)$ time. If s can be efficiently computed under these conditions, then the DCP_q^ℓ problem is considered broken.

Definition 6 (Learning with error, [42]). *Let λ be the security parameter, n, m, q be integers. The LWE problem states that for $A \in \mathbb{Z}_q^{m\times n}$, $s \in \mathbb{Z}_q^n$, $u \in \mathbb{Z}_q^m$, $e \in \chi \subset \mathbb{Z}_q^m$ the following distributions are computationally indistinguishable: $(A, As + e) \approx_C (A, u)$.*

3 Probabilistic Attacks of CM20

This section will introduce the operational details of the CM20 and explain the principles of the probabilistic attacks. The CM20 Protocol, which involves two participants, P_1 and P_2, who use security parameters λ, σ, protocol parameters $m, \omega, \ell_1, \ell_2$, two hash functions $H_1 : \{0,1\}^* \to \{0,1\}^{\ell_1}$,

$H_2 : \{0,1\}^\omega \to \{0,1\}^{\ell_2}$, and a pseudorandom function $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$. The protocol is divided into three main steps: precomputation, oblivious transfer, and OPRF evaluation. That is

1. **Precomputation**
   - P_1 generates a random string $s \leftarrow \{0,1\}^w$.
   - P_2 performs the following steps:
     – Initialize an $m\times w$ binary matrix D, with all entries set to 1. Let the column vectors of D be denoted as $D_1, D_2, \ldots, D_w$, and initialize them as $D_1 = D_2 = \cdots = D_w = 1^m$.
     – Sample a random key for a pseudorandom function $k \leftarrow \{0,1\}^\lambda$.
     – For each $y \in Y$, compute $v = F_k(H_1(y))$. Then, for each $i \in [w]$, set $D_i[v[i]] = 0$.
2. **Oblivious Transfer**
   - P_1 and P_2 perform w random OTs with message length m, where P_1 is the receiver with inputs choice bits $s[1], \ldots, s[w]$. As a result, P_2 gets w pairs of random messages $\{r_i^{(0)}, r_i^{(1)}\}_{i\in[w]}$ and P_1 gets w messages $\{r_i\}_{i\in[w]}$ where $r_i = r^{(s[i])}$.
   - P_2 does the following:
     – Let $\{r^{(0)}\}_{i\in[w]}$ form the column vectors of the matrix A and compute the matrix $B = A \oplus D$.
     – Compute $\delta_i = B_i \oplus r_i^{(1)}$ for all $i \in [w]$ and send to P_1.
3. **OPRF Evaluation**
   (a) P_2 sends the PRF key k to P_1.
   (b) $\forall x \in \mathcal{X}$, P_1 computes the matrix C as follows: if $s[i] = 0$ then set $C_i = r_i$; otherwise set $C_i = r_i \oplus \delta_i$ and $v = F_k(H_1(x))$ and its OPRF value $\psi = H_2(C_1[v[1]]\|\cdots\|C_\omega[v[\omega]])$ 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 $v = F_k(y)$ and its OPRF value $\psi = H_2(A_1[v[1]]\|\cdots\|A_\omega[v[\omega]])$ and outputs y iff $\psi \in \Psi$.

Definition 7. *The notation $\mathbb{P}_X = \{\Pr(x_1), \ldots, \Pr(x_n)\}$ represents the probability of the occurrence of element x_i in the set X, where $i = 1, \ldots, n$ and $X = \{x_1, \ldots, x_n\}$.*

Theorem 1. *Suppose that the adversary $\mathcal{A}$ can intercept the transmission channel information $\{r_i^{(0)}, r_i^{(1)}\}_{i\in[w]}$ and $\{r_i = r^{(s[i])}, \delta_i\}_{i\in[w]}$, and by some means obtains the probability distribution $\mathbb{P}_X$ corresponding to the private data X. Suppose further that the matching party P_2 with user P_1 does not update the pseudorandom function key k for a short period of time. In this case, the advantage of adversary $\mathcal{A}$ in performing a probabilistic attack on CM20 is non-negligible.*

Proof. First, $\mathcal{A}$ intercepts the data $\{r_i^{(0)}, r_i^{(1)}\}_{i\in[w]}$ from the OT channel of user P_2 and the data $\{\delta_i\}_{i\in[w]}$ sent to another user P_1, and checks whether $D'_i = \delta_i \oplus r_i^{(0)} \oplus r_i^{(1)}$ belongs to $\{0,1\}^{m\times w}$. If it does, then $\mathcal{A}$ builds a list, Table D, and stores the data $D' = D'_1\|\cdots\|D'_w$ in the table for statistical purposes; otherwise, it continues to intercept new data $\{r_i^{(0)}, r_i^{(1)}\}_{i\in[w]}$. Then $\mathcal{A}$ performs statistics on the collected data in Table D, and performs the operations of Algorithm 1.

Probabilistic Attacks Feasibility Analysis Currently, the adversary can obtain transmission data from the channel using some mature techniques, such as eavesdropping attack[1]. This is because the CM20 encrypts the private data x during the unintentional transmission phase using a pseudorandom function, as shown below.

$$\delta_i = B_i \oplus r_i^{(1)} = D_i \oplus r_i^{(0)} \oplus r_i^{(1)}.$$

Therefore, $\mathcal{A}$ can recover D_i, which is an encrypted value related to the pseudorandom function value $v^x = F_k(H_1(x))$. Furthermore, for the same private data $x_1, x_2 \in X$, if $x_1 = x_2$, then $D_i^{x_1} = D_i^{x_2}$, where $D_i^{x_1}$ is the encrypted value related to the pseudorandom function value $v^{x_1} = F_k(H_1(x_1))$. □

4 Perturbed Pseudorandom Generator

4.1 Definition of PPRG

Definition 8 (Perturbed PRG)**.** *A perturbed pseudorandom generator, 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 $x_1 = x_2$, $\Pr(G_\gamma(x_1) = G_\gamma(x_2)) \le \varepsilon(n)$,
(2) When $x_1 = x_2$, such that $\|G_\gamma(x_1) - G_\gamma(x_2)\| < \gamma$, there exists N such that $\|G_\gamma(x_1) - G_\gamma(x_2)\| \ge \gamma \cdot N$ as $x_1 \ne x_2$, where clearly $N = 1$ is optimal.

Algorithm 2 Perturbed Pseudorandom Generator $G_\gamma(\cdot)$

```
1: Input: $x \in \mathbb{Z}^n_{\{0,1\}}$
2: Output: $G_\gamma(x)$
3: Setup: Initialize $A \in \mathbb{Z}^{m\times n}_{\{0,1\}}$, $e \in \mathbb{Z}^m_{\{0,1\}}$
4: Enc: Compute
       $G_\gamma(x) = (Ax) \bmod 3 + e$
```

4.2 Pseudorandomness proof of PPRG

Theorem 2 (Main theorem)**.** *Assume the construction of the $G_\gamma(\cdot)$ as depicted in Algorithm 2, then $G_\gamma(\cdot)$ is indistinguishable from $u \in \mathbb{Z}^m_{\{0,1,2,3\}}$ ($\mathbb{Z}^m_{\{0,1,2,3\}} = \mathbb{Z}^m_3$).*

Proof. To prove this theorem, we divide it into two parts: The first part proves that the indistinguishability probability between $(As) \bmod 3 + e$ and u is bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$ from Corollary 2.

The second part, Corollary 4, reduces the difficulty of solving the $G_\gamma(\cdot)$ as depicted in Algorithm 2 to the Extrapolated Dihedral Coset Problem. Lemma 4 and Lemma 5 further reduce the Extrapolated Dihedral Coset Problem to the Dihedral Coset Problem, thereby proving the difficulty of solving the $G_\gamma(\cdot)$. □

Part 1: The first part demonstrates that the indistinguishability probability between $(As) \bmod 3 + e$ and u is upper-bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.

[1]https://www.fortinet.com/resources/cyberglossary/eavesdropping

Fact 1. *Suppose that* $\mathbb{P}_{\{0,1\}} = \{\Pr(0), \Pr(1)\}$, $\mathbb{P}^{(1)}_{\{0,1\}} = \{\frac{1}{2}, \frac{1}{2}\}$, $\mathbb{P}^{(2)}_{\{0,1\}} = \{\frac{1}{\omega}, \frac{\omega-1}{\omega}\}$, *then*

$$\begin{aligned}\mathbb{P}^{(1)}_{\{0,1\}} \odot \mathbb{P}^{(2)}_{\{0,1\}} &= \mathbb{P}^{(3)}_{\{0,1\}}\\ &= \left\{\frac{1}{2}\cdot\frac{1}{\omega} + \frac{1}{2}\cdot\frac{1}{\omega} + \frac{1}{2}\cdot\frac{\omega-1}{\omega}, \frac{1}{2}\cdot\frac{\omega-1}{\omega}\right\}\\ &= \left\{\frac{1}{2} + \frac{1}{2\omega}, \frac{1}{2} - \frac{1}{2\omega}\right\}.\end{aligned}$$

Fact 2. *Suppose that* $\mathbb{P}^{(3)}_{\{0,1\}} = \{\frac{1}{2}+\frac{1}{2\omega}, \frac{1}{2}-\frac{1}{2\omega}\} = \{\Pr(0), \Pr(1)\}$, *then*

$\mathbb{P}_{\{0,1,2\}} = \{\overset{'}{\Pr}(0), \overset{'}{\Pr}(1), \overset{'}{\Pr}(2)\} = \mathbb{P}^{(3)}_{\{0,1\}} \uplus \mathbb{P}^{(3)}_{\{0,1\}}$. *Here,*

$$\begin{aligned}\overset{'}{\Pr}(0) &= \Pr(0)\Pr(0) = \left(\frac{1}{2}+\frac{1}{2\omega}\right)^2 = \frac{1}{3} + \frac{1}{2\omega} + \frac{1}{4\omega^2} - \frac{1}{12},\\ \overset{'}{\Pr}(1) &= \Pr(0)\Pr(1) + \Pr(1)\Pr(0) = 2\left(\frac{1}{2}+\frac{1}{2\omega}\right)\left(\frac{1}{2}-\frac{1}{2\omega}\right)\\ &= \frac{1}{3} + \frac{1}{6} - \frac{1}{2\omega^2},\\ \overset{'}{\Pr}(2) &= \Pr(1)\Pr(1) = \left(\frac{1}{2}-\frac{1}{2\omega}\right)^2 = \frac{1}{3} - \frac{1}{2\omega} + \frac{1}{4\omega^2} - \frac{1}{12}.\end{aligned}$$

Furthermore, there is also

$$\begin{aligned}\mathbb{P}'_{\{0,1,2\}} &= \{\overset{''}{\Pr}(0), \overset{''}{\Pr}(1), \overset{''}{\Pr}(2)\}\\ &= \mathbb{P}_{\{0,1,2\}} \uplus \mathbb{P}_{\{0,1,2\}} \bmod 3. \textit{ Here,}\\ \overset{''}{\Pr}(0) &= \overset{'}{\Pr}(0)\overset{'}{\Pr}(0) + \overset{'}{\Pr}(1)\overset{'}{\Pr}(2) + \overset{'}{\Pr}(2)\overset{'}{\Pr}(1), \qquad (1)\\ \overset{''}{\Pr}(1) &= \overset{'}{\Pr}(0)\overset{'}{\Pr}(1) + \overset{'}{\Pr}(1)\overset{'}{\Pr}(0) + \overset{'}{\Pr}(2)\overset{'}{\Pr}(2),\\ \overset{''}{\Pr}(2) &= \overset{'}{\Pr}(0)\overset{'}{\Pr}(2) + \overset{'}{\Pr}(1)\overset{'}{\Pr}(1) + \overset{'}{\Pr}(2)\overset{'}{\Pr}(0).\end{aligned}$$

The equation (1) *can be rewritten as*

$$\begin{aligned}\mathbb{P}'_{\{0,1,2\}} &= \begin{pmatrix} \Pr'(0) & \Pr'(2) & \Pr'(1)\\ \Pr'(1) & \Pr'(0) & \Pr'(2)\\ \Pr'(2) & \Pr'(1) & \Pr'(0)\end{pmatrix}\begin{pmatrix}\Pr'(0)\\ \Pr'(1)\\ \Pr'(2)\end{pmatrix}\\ &= M_{\mathbb{P}_{\{0,1,2\}}}\mathbb{P}_{\{0,1,2\}}.\end{aligned}$$

Fact 3. *For the sequence*

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}}\mathbb{P}^{(n-1)}_{\{0,1,2\}},$$

define

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = (a_n^{(0)}, a_n^{(1)}, a_n^{(2)}) = \left(\frac{1}{3}+\Delta_n^{(0)}, \frac{1}{3}+\Delta_n^{(1)}, \frac{1}{3}+\Delta_n^{(2)}\right).$$

Claim 1. *The sequence*

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}}\mathbb{P}^{(n-1)}_{\{0,1,2\}}$$

is a Cauchy sequence.

Proof. To prove that

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}}\mathbb{P}^{(n-1)}_{\{0,1,2\}}$$

forms a Cauchy sequence, it suffices to show that for any $\delta > 0$, there exists an$N > 0$ such that for any $n > N$,

$$\left\|\mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}}\right\| \leq \delta.$$

Because

$$\begin{aligned}\mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}} &= \begin{pmatrix} (\Delta_0^{(n-1)})^2 + 2\Delta_1^{(n-1)}\Delta_2^{(n-1)}\\ (\Delta_2^{(n-1)})^2 + 2\Delta_0^{(n-1)}\Delta_1^{(n-1)}\\ (\Delta_0^{(n-1)})^2 + 2\Delta_0^{(n-1)}\Delta_2^{(n-1)}\end{pmatrix}\\ &= \begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)}\\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)}\\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)}\end{pmatrix}\begin{pmatrix}\Delta_0^{(n-1)}\\ \Delta_1^{(n-1)}\\ \Delta_2^{(n-1)}\end{pmatrix}.\end{aligned}$$

And

$$\begin{aligned}&\left\|\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)}\\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)}\\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)}\end{pmatrix}\right\|\\ &\leq \left\|\begin{pmatrix} \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2}\\ \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2}\\ \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2} & \frac{1}{6}-\frac{1}{2\omega^2}\end{pmatrix}\right\|\\ &= \frac{1}{2} - \frac{3}{2\omega^2}.\end{aligned}$$

So, it is obtained that

$$\begin{aligned}&\left\|\mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}}\right\|\\ &\leq \left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^{n-1}\left\|\mathbb{P}^{(1)}_{\{0,1,2\}} - \mathbb{P}^{(0)}_{\{0,1,2\}}\right\|\\ &\leq \frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^{n}.\end{aligned}$$

□

Lemma 3. *For any initial vector* $a_0 = (a_0^{(0)}, a_0^{(1)}, a_0^{(2)})$, *where* $a_0^{(0)}, a_0^{(1)}, a_0^{(2)} \in [(\frac{1}{2}-\frac{1}{2\omega})^2, (\frac{1}{2}+\frac{1}{2\omega})^2]$ *and* $\sum_{i=0}^{2} a_0^{(i)} = 1$, *and* $\omega > 2$, *the matrix* M_{a_0} *is generated as follows:*

$$M_{a_0} = \begin{pmatrix} a_0^{(0)} & a_0^{(2)} & a_0^{(1)}\\ a_0^{(1)} & a_0^{(0)} & a_0^{(2)}\\ a_0^{(2)} & a_0^{(1)} & a_0^{(0)}\end{pmatrix}.$$

Then, let $a_{n+1} = M_{a_n}a_n = Ta_n$, *then* $\{a_n\}_{n=0}^{\infty}$ *is a Cauchy sequence and converges to* $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Proof. According to Claim 1, we know that

$$\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)}\\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)}\\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)}\end{pmatrix}$$

is a contraction operator, and

$$\left\|\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)}\\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)}\\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)}\end{pmatrix}\right\| \leq \frac{1}{2} - \frac{3}{2\omega^2}.$$

Therefore, the matrix

$$\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix}$$

is contractive, with $(0,0,0)$ being both a convergent point and a fixed point of this matrix sequence. Moreover, since $a_{n+1} = M_{a_n} a_n$ has been proven to be a Cauchy sequence, the sequence $\{a_n\}_{n=0}^{\infty}$ converges, and it converges to the fixed point of $T(\cdot)$. □

Theorem 3. *Given $\{a_j\}_{j=0}^{2}$ and $\{s_j\}_{j=1}^{2}$ such that $a_j \in_R \mathbb{Z}_{\{0,1\}}$, $s \in \mathbb{Z}_{\{0,1\}}$, and the probability that the components of s equal 0 is $\frac{1}{\omega}$, while the probability that they equal 1 is $\frac{\omega-1}{\omega}$, where $\omega > 2$. Then for any $i = 0,1,2$, we have*

$$\max_{i=0,1,2} \left| \Pr\left(\sum_{j=0}^{2}(a_j s_j) = i\right) - \Pr(u = i) \right| \leq \frac{1}{\sqrt{3}} \left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n.$$

Corollary 1. *For any $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$, $s \in \mathbb{Z}_{\{0,1\}}^{n}$ where the probability that the components of s equal 0 is $\frac{1}{\omega}$ and the probability that they equal 1 is $\frac{\omega-1}{\omega}$, with $\omega > 2$, and $u \in \mathbb{Z}_{\{0,1,2\}}^{m}$, then the indistinguishability probability between $As \bmod 3$ and u is bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.*

Corollary 2. *For any $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$, $s \in \mathbb{Z}_{\{0,1\}}^{n}$ where the probability that the components of s equal 0 is $\frac{1}{\omega}$ and the probability that they equal 1 is $\frac{\omega-1}{\omega}$, $e \in_R \mathbb{Z}_{\{0,1\}}^{n}$, $\omega > 2$, and $u \in \mathbb{Z}_{\{0,1,2,3\}}^{m}$, then the indistinguishability probability between $(As) \bmod 3 + e$ and u is bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.*

Part 2: The second part mainly reduces the difficulty of solving the $G_\gamma(\cdot)$ as depicted in Algorithm 2 to the Dihedral Coset Problem, thereby proving the difficulty of solving the $G_\gamma(\cdot)$.

Fact 4. *The Dihedral Coset Problem is a challenging problem in quantum computing, hence the time complexity for solving it is $O(e^n)$ or $O(n!)$.*

Lemma 4. *Currently, there is no efficient algorithm that can solve DCP_3^ℓ in polynomial time.*

Proof. By way of contradiction, assume $q = 2^n$ and there exists an efficient algorithm $\mathcal{W}$ that can solve DCP_3^ℓ in polynomial time. Then for an instance of DCP_9^ℓ, we have

$$|0\rangle|x_i\rangle + |1\rangle|(x_i + s) \bmod 9\rangle = |0\rangle|x_i'\rangle + |1\rangle|(x_i' + s') \bmod 3\rangle + 3(|0\rangle|x_i''\rangle + |1\rangle|(x_i'' + s'') \bmod 3\rangle), \quad i \leq \ell,$$

which implies running $\mathcal{W}$ twice can solve $DCP_{9=3^2}^\ell$. Similarly, running $\mathcal{W}$ four times can solve $DCP_{81=3^4}^\ell$, and so forth, requiring $\frac{n}{2}$ executions of $\mathcal{W}$ to solve DCP_q^ℓ. Let $O(\mathcal{W})$ denote the time complexity of $\mathcal{W}$. Thus, we have

$$\frac{n}{2}O(\mathcal{W}) \geq O(e^n) \Rightarrow O(\mathcal{W}) \geq \frac{2O(e^n)}{n},$$

or

$$\frac{n}{2}O(\mathcal{W}) \geq O(n!) \Rightarrow O(\mathcal{W}) \geq \frac{2O(n!)}{n}.$$

This contradicts the hypothesis that $\mathcal{W}$ can solve DCP_3^ℓ in polynomial time, thus proving the theorem. □

Definition 9 (Extrapolated Dihedral Coset Problem with model 3, [5])**.** *Given security parameter κ, input instance $EDCP_{n,3,\rho}^\ell$ where 3 denotes the modulus, ρ denotes the probability density function, and ℓ denotes the number of states, each state is represented as*

$$\sum_{j\in \mathrm{supp}(\rho)} \rho(j)|j\rangle|(x_i + js) \bmod 3\rangle, \quad i \leq \ell,$$

with 3 bits stored, where $x_i \in_R \mathbb{Z}_3^n$, $s \in \mathbb{Z}_4^n$. The problem $EDCP_{n,3,\rho}^\ell$ is considered broken if s can be computed with probability $\mathrm{poly}(1/(n\log 3))$ within $\mathrm{poly}(n\log 3)$ time.

Lemma 5. *If there exists an algorithm to solve $EDCP_{n,3,\rho}^\ell$, then there exists an algorithm to solve DCP_3^ℓ.*

Proof. Let

$$|b\rangle = \frac{1}{\sqrt{2}}|0\rangle|x_i\rangle + \frac{1}{\sqrt{2}}|1\rangle|(x_i + s) \bmod 3\rangle.$$

Thus, $\rho(0)|0\rangle = \frac{1}{\sqrt{2}}|0\rangle$, $\rho(1)|1\rangle = \frac{1}{\sqrt{2}}|1\rangle$, implying DCP_3^ℓ is a special case of $EDCP_{n,3,\rho}^\ell$. Therefore, if there exists an algorithm to solve $EDCP_{n,3,\rho}^\ell$, then there exists an algorithm to solve DCP_3^ℓ. □

Lemma 6 ([5])**.** *Let $(n, q, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP, and (n, q, α) an instance of LWE. If there exists an algorithm to solve $LWE_{n,q,\alpha}$, then there exists an algorithm to solve G-$EDCP_{n,q,\rho_r}^\ell$.*

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

Corollary 4. *Let $(n, 3, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP, and $G(x) = As \bmod 3 + e$, where $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$, $x \in \mathbb{Z}_{\{0,1\}}^{n}$, $e \in \mathbb{Z}_{\{0,1\}}^{m}$. If there exists an algorithm to recover x from G(x), then there exists an algorithm to solve G-$EDCP_{n,3,\rho_r}^\ell$.*

4.3 Correctness proof of PPRG

Theorem 4. *Assume the construction of the PRG $G_\gamma(\cdot)$ as depicted in Figure 2, then $G_\gamma(\cdot)$ satisfies Definition 8.*

Proof. Prove each statement separately. First, when $x_1 = x_2$, it be known that

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

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

0. P_1 and P_2 agree on security parameters λ, σ, protocol parameters $m, \omega, \ell_1, \ell_2$, a hash function $H_1 : \mathbb{Z}_3^* \to \{0,1\}^{\ell_1}$ and a $G_\gamma : \{0,1\}^\omega \to \mathbb{Z}_3^{\ell_2}$, a pseudorandom function $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$.
1. **Precomputation**
   - P_2 performs the following steps:
     – Initialize an $m \times w$ binary matrix D, with all entries set to 1. Let the column vectors of D be denoted as $D_1, D_2, \ldots, D_w$, and initialize them as $D_1 = D_2 = \cdots = D_w = 1^m$.
     – Sample a random key for a pseudorandom function $k \leftarrow \{0,1\}^\lambda$.
     – For each $y \in Y$, compute $v = F_k(H_1(y))$. Then, for each $i \in [w]$, set $D_i[v[i]] = 0$.
2. **PPRG Evaluation**
   (a) P_2 sends the PRF key k to P_1.
   (b) $\forall x \in \mathcal{X}$, P_1 computes $v = F_k(H_1(x))$ and its PRF value $\psi = G_\gamma(D)$ and sends ψ to P_2.
   (c) Let Ψ be the set of PRF values received from P_1. $\forall y \in \mathcal{Y}$, P_2 computes $v = F_k(y)$ and its PPRG value $\|\psi - G_\gamma(D)\| < \sqrt{\omega}\gamma$ and outputs y iff $\psi \in \Psi$.

Figure 3: Private Set Intersection from Perturbed Pseudorandom Generator

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

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

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

Therefore

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

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

5 Our PSI Protocol from PPRG

Theorem 5. *Suppose the PRF F is a random oracle. If it is hard to distinguish between the two outputs of the perturbed pseudorandom oracle $Pr\mathcal{O}M_\gamma$, then our protocol is provably secure in the definition 1 with reduction loss $L = q_H$, where q_H is the number of PRF queries to the random oracle.*

Proof. Suppose the adversary $\mathcal{A}$ can (t, ε)-break the scheme with non-negligible advantage. Now, the simulator $\mathcal{S}$ simulates the scheme. Suppose there exists a $Pr\mathcal{O}M_\gamma$ such that

$$PPr\mathcal{O}M_\gamma(x) \to (y_0, y_1) \begin{array}{l} \nearrow \quad y_0 = G_\gamma^{black-box}(x) \in \mathbb{Z}_4^{\ell_2}, \\ \searrow \quad y_1 \in_R \mathbb{Z}_4^{\ell_2}. \end{array}$$

$\mathcal{S}$ controls the random oracle, runs $\mathcal{A}$, and works as follows.

- **Setup** The simulator $\mathcal{S}$ generates some necessary parameters for the algorithms and selects a PRF $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$ with key $k \in \{0,1\}^\lambda$.
- **PRF-Query** The adversary $\mathcal{A}$ makes query about the PRF. The simulator $\mathcal{S}$ pre-establishes lists for handling PRF-Query. That is, for the i^{th} queries $x_i \in \{0,1\}^{\ell_1}$ corresponding to the value of PRF, the simulator $\mathcal{S}$ selects from the PRF $D_i \in \{0,1\}^{m\times w}$ list if available, otherwise selects a random $v_i \in [m]^\omega$, let $D_1 = \cdots = D_\omega = 1^m$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$ and adds (x_i, D_i) to the PRF list.
- **Challenge** $\mathcal{A}$ selects $x^* \in \mathcal{X}$ and sends it to $\mathcal{S}$. $\mathcal{S}$ using the corresponding pseudorandom function queries, inputs the queried values into the $Pr\mathcal{O}M_\gamma$, obtaining ψ_0 and ψ_1, and then sends ψ_0, ψ_1 to $\mathcal{A}$.
- **Guess** Based on the received ψ_0 and ψ_1, $\mathcal{A}$ guesses whether ψ_0 or ψ_1 is the ciphertext of the encrypted message m.

The simulator randomly selects one value x from the PRF list $(x_1, D_1), (x_2, D_2), \ldots, (x_{q_H}, D_{q_H})$ as the challenge PRF query. The simulator can immediately use the adversary's guess result to distinguish between the two outputs of the $Pr\mathcal{O}M_\gamma$. This completes the simulation and the solution. The correctness is analyzed as follows.

Indistinguishable simulation. The correctness of the simulation has been explained above. The randomness of the simulation includes all random numbers in the responses to PRF queries. They are

$$D_1, D_2, \ldots, D_{q_H}.$$

According to the setting of the simulation, where D_i are randomly chosen, it is easy to see that the randomness property holds, and thus the simulation is indistinguishable from the real attack.

Probability of successful simulation. There is no abort in the simulation, and thus the probability of successful simulation is 1.

Advantage of breaking the challenge ciphertext. If the adversary guesses with a non-negligible advantage that ψ_b is the encryption of x^*, where $b = 0$ or 1, then the simulator can use the adversary's guess to distinguish between the two outputs of the $Pr\mathcal{O}M_\gamma$ with a non-negligible advantage.

Advantage and time cost. Let T_s denote the time cost of the simulation. We have $T_s = O(1)$. Therefore, the simulator $\mathcal{B}$ will distinguish between the two outputs of the $Pr\mathcal{O}M_\gamma$ with $(t + T_s, \frac{\epsilon}{q_H})$.

This completes the proof of the theorem. □

6 Performance Evaluation

6.1 Analysis of Efficiency on PC

The tools used in this subsection are Python 3.8, the programs run on a Vostro Dell PC Desktop 10th Gen Intel(R)Core(TM) i5-11400@2.60GHz 2.59GHz, RAM 8.00GB. CM20's hash function H_2 is the built-in hash function in Python.

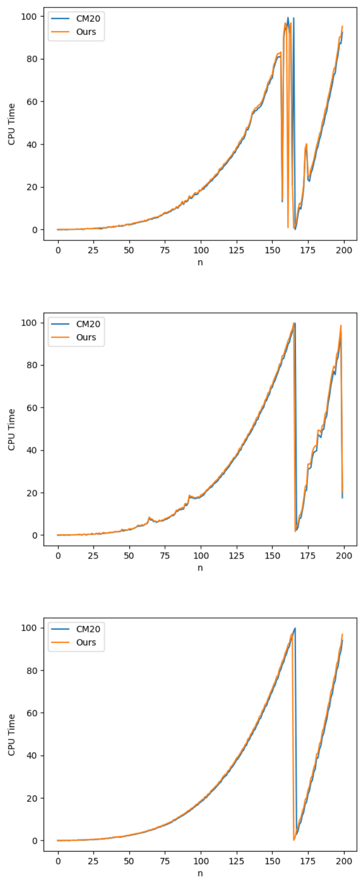


Figure 4: Parallel comparison of encryption on PC, where n represents the security parameter, unit is 10^4 microseconds

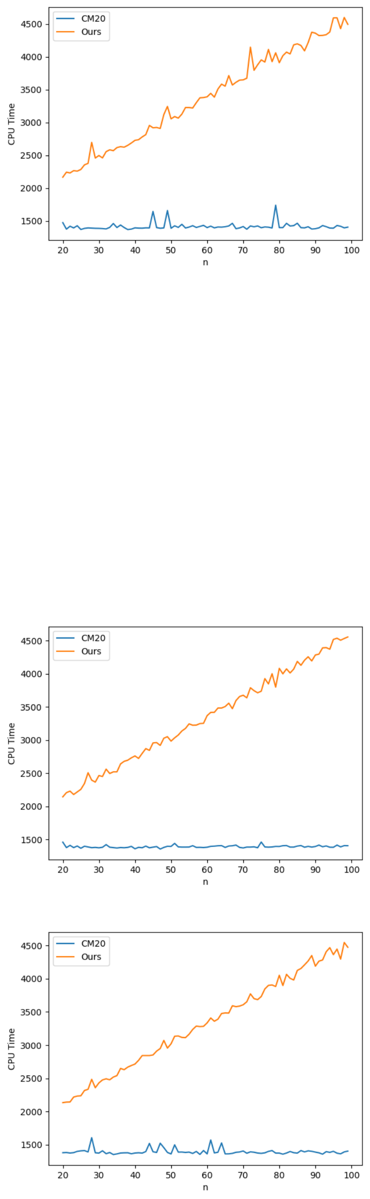


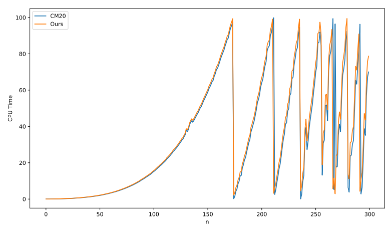
Figure 5: Parallel comparison of decryption on MAC, where n represents the number of elements in P_2's private set, with time measured in microseconds

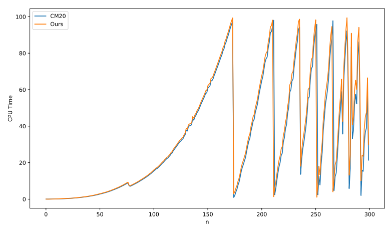
6.2 Analysis of Efficiency on MAC

The tools used in the subsection are Python 3.12, the programs are performed on MacBook Air MAC Desktop Apple M1, RAM 8.00GB. The hash function H_2 of CM20 is the built-in hash function in Python.

6.3 Analysis of Efficiency on Mobile Pads

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@3.2GHz, RAM 8.00+3.00GB.





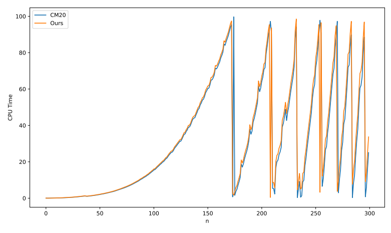
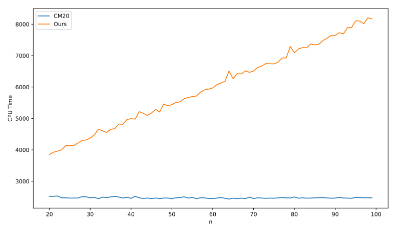
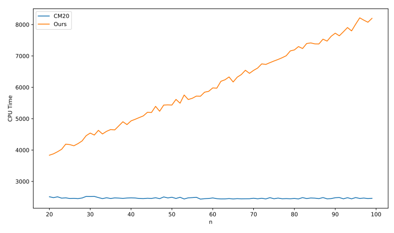


Figure 6: Parallel comparison of encryption on mobile pads, where n represents the security parameter, unit is 10^4 microseconds





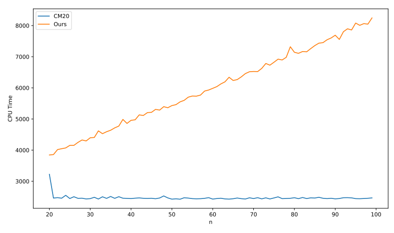


Figure 7: Parallel comparison of decryption on mobile pads, where n represents the number of elements in P_2's private set, with time measured in microseconds

6.4 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@2.2GHz, RAM 6.00GB.

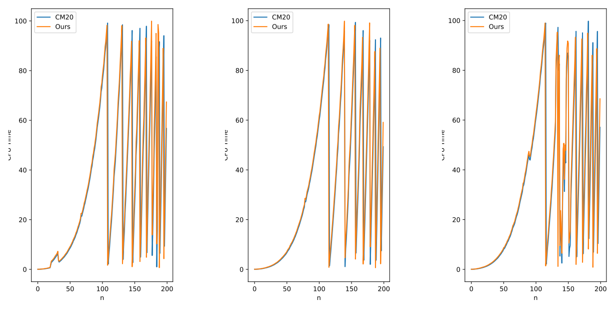


Figure 8: Parallel comparison of encryption on mobile phones, where n represents the security parameter, unit is 10^4 microseconds

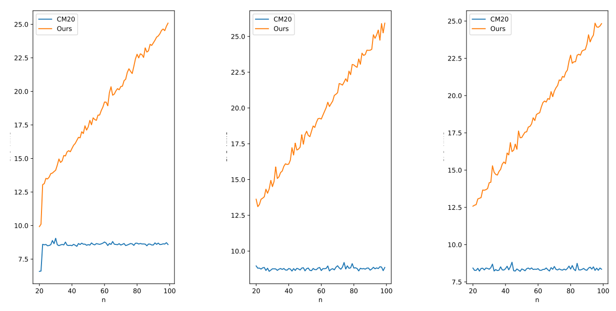


Figure 9: Parallel comparison of decryption on mobile phones, where n represents the number of elements in P_2's private set, with time measured in 10^3 microseconds

6.5 Summary of Data Comparison

Enc The encryption algorithm used in this paper shows similar efficiency to CM20, whether on PC or mobile devices. Among them, the CPU Times frequency of mobile device phone is significantly faster than that of PC and mobile device pad. The CPU times frequency of PC is slightly faster than that of mobile device pad. However, whether it is PC, mobile pad, or mobile phone, it seems that they have an upper limit of 100×10^4 milliseconds.

Dec The decryption algorithm used in this paper is significantly higher than CM20. The computational overhead on MAC shows a step-like pattern, while on mobile pad and phone, it roughly follows a linear trend. In comparison, the CPU times of CM20 can be almost disregarded.

7 Conclusion

7.1 Advantages

The main work of this paper is to analyse and improve the work of Miao and Chase, we find that the pseudorandom function values can be recovered by collecting data in the OT phase, and assuming that the user does not update the pseudorandom function key for a period of time, the strong encryption of the CM20 protocol degrades to weak encryption. The longer the pseudorandom function key update time, the less secure the user's privacy is, and we call this attack method a probabilistic attack. To resist this attack, we introduce the concept of PPRG and construct the PSI protocol based on PPRG. Compared with CM20, our protocol has less OT phase and second hash computation phase instead of P-PRG phase, which reduces the interaction between users and also maintains the strong encryption of the protocol, which effectively resists probabilistic attacks. Finally, we define the CPA security model of the new PSI protocol and prove that the new PSI protocol satisfies the CPA security model.

7.2 Disadvantages and Future Work

Although the protocol constructed in this paper is resistant to probabilistic attacks, it also increases the computational burden on the user. Therefore, a non-interactive OT technique is needed, or to improve the results of PPRG, such as constructing PPRG using ring LPN in combination with fast fourier transform. But even then, the theoretical computational complexity is still $O(n \log n)$.

Moreover, although the PPRG in this paper is reduced to the dihedral coset problem, this does not mean that our protocol is post-quantum secure. Lai, Liu and others have introduced the concept of a quantum adversary who can query an exponential number of quantum random oracle [32]. To resist such an adversary, a quantum PRF [48, 49] is needed instead of the random oracle of theorem 5.

References

[1] Donald Beaver. 1995. Precomputing Oblivious Transfer. In *Advances in Cryptology — CRYPT0' 95*, Don Coppersmith (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 97–109.

[2] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. 2018. Compressing Vector OLE. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 896912. https://doi.org/10.1145/3243734.3243868

[3] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. 2019. Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 291308. https://doi.org/10.1145/3319535.3354255

[4] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. 2019. Efficient Pseudorandom Correlation Generators: Silent OT Extension and More. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, Cham, 489–518.

[5] Zvika Brakerski, Elena Kirshanova, Damien Stehlé, and Weiqiang Wen. 2018. Learning with Errors and Extrapolated Dihedral Cosets. In *Public-Key Cryptography – PKC 2018*. Springer International Publishing, 702–727.

[6] Andrzej Cegielski. 2012. *Iterative Methods for Fixed Point Problems in Hilbert Spaces*. Springer Berlin, Heidelberg.

[7] Melissa Chase and Peihan Miao. 2020. Private Set Intersection in the Internet Setting from Lightweight Oblivious PRF. In *Advances in Cryptology – CRYPTO 2020*. Springer International Publishing, 34–63.

[8] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. 2021. Silver: Silent VOLE and Oblivious Transfer from Hardness of Decoding Structured LDPC Codes. In *Advances in Cryptology – CRYPTO 2021*, Tal Malkin and Chris Peikert (Eds.). Springer International Publishing, Cham, 502–534.

[9] Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. 2009. Efficient Robust Private Set Intersection. In *Applied Cryptography and Network Security*, Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 125–142.

[10] Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. 2011. Secure Efficient Multiparty Computing of Multivariate Polynomials and Applications. In *Applied Cryptography and Network Security*, Javier Lopez and Gene Tsudik (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 130–146.

[11] Emiliano De Cristofaro, Jihye Kim, and Gene Tsudik. 2010. Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model. In *Advances in Cryptology - ASIACRYPT 2010*, Masayuki Abe (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 213–231.

[12] Emiliano De Cristofaro and Gene Tsudik. 2010. Practical Private Set Intersection Protocols with Linear Complexity. In *Financial*

Cryptography and Data Security. Springer Berlin Heidelberg, 143–159.

[13] Changyu Dong, Liqun Chen, and Zikai Wen. 2013. When private set intersection meets big data: an efficient and scalable protocol. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*. Association for Computing Machinery, 789–800. https://doi.org/10.1145/2508859.2516701

[14] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. 2004. Efficient Private Matching and Set Intersection. In *Advances in Cryptology - EUROCRYPT 2004*. Springer Berlin Heidelberg, 1–19.

[15] Gayathri Garimella, Payman Mohassel, Mike Rosulek, Saeed Sadeghian, and Jaspal Singh. 2021. Private Set Operations from Oblivious Switching. In *Public-Key Cryptography – PKC 2021*, Juan A. Garay (Ed.). Springer International Publishing, Cham, 591–617.

[16] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2021. Oblivious Key-Value Stores and Amplification for Private Set Intersection. In *Advances in Cryptology – CRYPTO 2021*, Tal Malkin and Chris Peikert (Eds.). Springer International Publishing, Cham, 395–425.

[17] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. 2022. Structure-Aware Private Set Intersection, withApplications to-Fuzzy Matching. In *Advances in Cryptology – CRYPTO 2022*, Yevgeniy Dodis and Thomas Shrimpton (Eds.). Springer Nature Switzerland, Cham, 323–352.

[18] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. 2023. Malicious Secure, Structure-Aware Private Set Intersection. In *Advances in Cryptology – CRYPTO 2023*, Helena Handschuh and Anna Lysyanskaya (Eds.). Springer Nature Switzerland, Cham, 577–610.

[19] Satrajit Ghosh and Tobias Nilges. 2019. An Algebraic Approach to Maliciously Secure Private Set Intersection. In *Advances in Cryptology – EUROCRYPT 2019*. Springer International Publishing, 154–185.

[20] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. 1986. How to construct random functions. *Journal of ACM* 33, 4 (1986), 792–807. https://doi.org/10.1145/6490.6503

[21] Fuchun Guo, Willy Susilo, and Yi Mu. 2018. Foundations of Security Reduction. In *Introduction to Security Reduction*. Springer, 32–33.

[22] Carmit Hazay and Yehuda Lindell. 2008. Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In *Theory of Cryptography*, Ran Canetti (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 155–175.

[23] Yupu Hu, Siyue Dong, Baocang Wang, and Xingting Dong. 2023. On the Invalidity of LV16/Lin17 Obfuscation Schemes Revisited. Cryptology ePrint Archive, Paper 2023/1291.

[24] Yan Huang, David Evans, and Jonathan Katz. 2012. Private set intersection: are garbled circuits better than custom protocols?. In *NDSS 2012*. The Internet Society.

[25] Bernardo A. Huberman, Matt Franklin, and Tad Hogg. 1999. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (Denver, Colorado, USA) *(EC '99)*. Association for Computing Machinery, New York, NY, USA, 7886. https://doi.org/10.1145/336992.337012

[26] Bernardo A. Huberman, Matthew K. Franklin, and Tad Hogg. 1999. Enhancing privacy and trust in electronic communities. In *ACM Conference on Economics and Computation*.

[27] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. 2003. Extending Oblivious Transfers Efficiently. In *Advances in Cryptology - CRYPTO 2003*. Springer Berlin Heidelberg, 145–161.

[28] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. 2023. The Pseudorandom Oracle Model and Ideal Obfuscation. In *Advances in Cryptology – CRYPTO 2023*. Springer Nature Switzerland, 233–262.

[29] Stanisław Jarecki and Xiaomin Liu. 2010. Fast Secure Computation of Set Intersection. In *Security and Cryptography for Networks*, Juan A. Garay and Roberto De Prisco (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 418–435.

[30] Lea Kissner and Dawn Song. 2005. Privacy-Preserving Set Operations. In *Advances in Cryptology – CRYPTO 2005*, Victor Shoup (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 241–257.

[31] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. 2016. 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, 818–829. https://doi.org/10.1145/2976749.2978381

[32] Qiqi Lai, Feng-Hao Liu, Yang Lu, Haiyang Xue, and Yong Yu. 2024. Scalable Two-Round n-out-of-n and Multi-Signatures from Lattices in the Quantum Random Oracle Model. Cryptology ePrint Archive, Paper 2024/1574. https://eprint.iacr.org/2024/1574

[33] Catherine Meadows. 1986. A More Efficient Cryptographic Matchmaking Protocol for Use in the Absence of a Continuously Available Third Party. In *1986 IEEE Symposium on Security and Privacy*. 134–134. https://doi.org/10.1109/SP.1986.10022

[34] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2019. SpOT-Light: Lightweight Private Set Intersection from Sparse OT Extension. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, Cham, 401–431.

[35] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2020. PSI from PaXoS: Fast, Malicious Private Set Intersection. In *Advances in Cryptology – EUROCRYPT 2020*. Springer International Publishing, 739–767.

[36] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. 2015. Phasing: private set intersection using permutation-based hashing. In *USENIX Security 2015*, J. Jung and T. Holz (Eds.). USENIX Association, 515–530.

[37] Benny Pinkas, Thomas Schneider, Oleksandr Tkachenko, and Avishay Yanai. 2019. Efficient Circuit-Based PSI with Linear Communication. In *Advances in Cryptology – EUROCRYPT 2019*. Springer International Publishing, 122–153.

[38] Benny Pinkas, Thomas Schneider, Christian Weinert, and Udi Wieder. 2018. Efficient Circuit-Based PSI via Cuckoo Hashing. In *Advances in Cryptology – EUROCRYPT 2018*. Springer International Publishing, 125–157.

[39] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2014. Faster private set intersection based on OT extension. In *23rd USENIX Security Symposium (USENIX Security 2014)*. 797–812.

[40] Michael O. Rabin. 1981. *How to Exchange Secrets with Oblivious Transfer*. Technical Report TR-81. Aiken Computation Lab, Harvard University. Technical Report.

[41] Srinivasan Raghuraman and Peter Rindal. 2022. Blazing Fast PSI from Improved OKVS and Subfield VOLE. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. Association for Computing Machinery, 2505–2517. https://doi.org/10.1145/3548606.3560658

[42] Oded Regev. 2005. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC '05)*. Association for Computing Machinery, 84–93. https://doi.org/10.1145/1060590.1060603

[43] Peter Rindal and Mike Rosulek. 2017. Improved Private Set Intersection Against Malicious Adversaries. In *Advances in Cryptology – EUROCRYPT 2017*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.). Springer International Publishing, Cham, 235–259.

[44] Peter Rindal and Mike Rosulek. 2017. Malicious-Secure Private Set Intersection via Dual Execution. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 12291242. https://doi.org/10.1145/3133956.3134044

[45] Peter Rindal and Phillipp Schoppmann. 2021. VOLE-PSI: Fast OPRF and Circuit-PSI from Vector-OLE. In *Advances in Cryptology – EUROCRYPT 2021*, Anne Canteaut and François-Xavier Standaert (Eds.). Springer International Publishing, Cham, 901–930.

[46] Mike Rosulek and Ni Trieu. 2021. Compact and Malicious Private Set Intersection for Small Sets. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 11661181. https://doi.org/10.1145/3460120.3484778

[47] Phillipp Schoppmann, Adrià Gascón, Leonie Reichert, and Mariana Raykova. 2019. Distributed Vector-OLE: Improved Constructions and Implementation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 10551072. https:

//doi.org/10.1145/3319535.3363228
[48] Mark Zhandry. 2012. How to Construct Quantum Random Functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. 679–687. https://doi.org/10.1109/FOCS.2012.37
[49] Mark Zhandry. 2021. How to Construct Quantum Random Functions. *Journal of ACM* 68, 5, Article 33 (Aug. 2021), 43 pages. https://doi.org/10.1145/3450745

Received 20 February 2024; revised 12 March 2024; accepted 5 June 2024