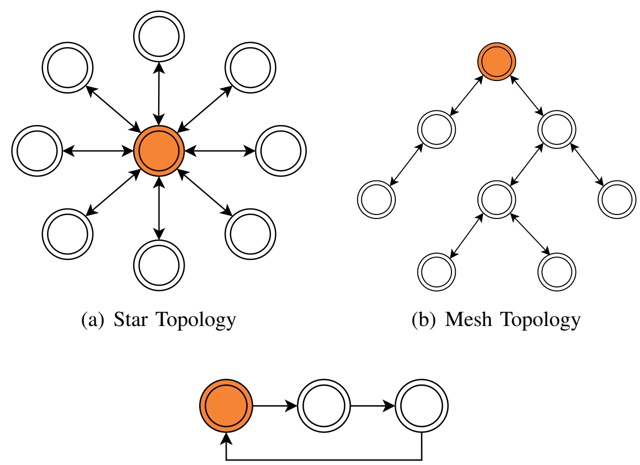
Communication Efficient Multiparty Private Set Intersection from Multi-Point Sequential OPRF

Xinyu Feng[†‡*], Yukun Wang[†*], Cong Li[§✉], Wu Xin[¶], Ming Yao[†], Dian Zhang[†], Wanwan Wang[†] and Hao He[†]
[†]*InsightOne Tech Co., Ltd.*
[‡]*Zhonghe Tech (Xiong'an) Co., Ltd.*
[§]*School of Software and Microelectronics, Peking University*
[¶]*School of Economics, Peking University*
[✉]*Corresponding Author*
[*]*The two authors contributed equally to this work.*

***Abstract*—Multiparty private set intersection (MPSI) allows multiple participants to compute the intersection of their locally owned data sets without revealing them. MPSI protocols can be categorized based on the network topology of nodes, with the star, mesh, and ring topologies being the primary types, respectively. Given that star and mesh topologies dominate current implementations, most existing MPSI protocols are based on these two topologies. However, star-topology MPSI protocols suffer from high leader node load, while mesh topology protocols suffer from high communication complexity and overhead. In this paper, we first propose a multi-point sequential oblivious pseudorandom function (MP-SOPRF) in a multi-party setting. Based on MP-SOPRF, we then develop an MPSI protocol with a ring topology, addressing the challenges of communication and computational overhead in existing protocols. We prove that our MPSI protocol is semi-honest secure under the Hamming correlation robustness assumption. Our experiments demonstrate that our MPSI protocol outperforms state-of-the-art protocols, achieving a reduction of 74.8% in communication and a 6% to 287% improvement in computational efficiency.**

1. Introduction

Private set intersection (PSI) is a versatile cryptographic instrument that facilitates two participants, each with distinct local data sets, in determining the common elements of their datasets without exposing the nonintersecting components. As an extension of PSI, multiparty PSI (MPSI) enables multiple participants to ascertain their common data elements without revealing nonintersecting data, in a similar manner. MPSI has numerous practical applications across various domains. In particular, within vertical federated learning (VFL), MPSI is employed for data alignment based on the local data identifiers of each participant, serving as an essential prerequisite [1]. In the domain of medical data collection, MPSI presents a promising technology for consolidating medical data from numerous disparate entities while ensuring the privacy of sensitive data [2]. Furthermore, in the field of cybersecurity protection, MPSI is instrumental for multiple organizations to collaboratively identify malicious intrusions on public networks without compromising the confidentiality of other users' information [3], [4].



(a) Star Topology (b) Mesh Topology

(c) Ring Topology

Figure 1. Network Topologies of MPSI

Within existing MPSI protocols, there are primarily three types of topological structures from a communication model perspective, namely, star topology, mesh topology, and ring topology [1], as depicted in Fig. 1. The star topology is characterized by multiple assistants engaging with a sole leader, which is predominantly observed among MPSI protocols such as public-key-based MPSI [6]–[8], or those utilizing oblivious programmable pseudorandom functions (OPPRF) and oblivious key-value stores (OKVS) [9], [10]. Nevertheless, this "centralized" configuration results in substantial bandwidth and computational demands on the leader. The mesh topology serves as an enhancement to the star topology by mitigating the communication load and computational

1. In various academic works, the ring topology is occasionally referred to as the ring topology [5].

pressure on the leader, bearing resemblance to a binary tree structure. This topology is adopted by several of the most efficient MPSI protocols [3], [11]. However, the complexity of this network increases, necessitating a robust overall network and sufficient bandwidth for nodes to possibly engage with multiple other nodes. Ring topology, originally proposed by Kavousi et al. [12], involves the sequential transmission of messages among all participants, with both the leader and the assistants assuming comparable protocol roles. In this configuration, the responsibility for the determination of the intersection resides exclusively with the leader at the end of the protocol. Unlike the other two topological frameworks, the ring topology offers distinct advantages: (1) It equitably allocates the communication load among all participants, thus preventing potential bottlenecks associated with a single leader in the star topology. (2) The ring structure reduces the dependency on any single node, thereby improving the resilience of the system to node failures and dropout attacks. (3) It supports scalability for larger networks by minimizing communication overhead, which increases exponentially within the mesh topology as the number of participants increases.

1.1. Motivation

In ring topology, embedding multi-party data in a secure end efficient way is quite challenging since each node merely performs two information transmissions, that is, receiving data from the previous node and sending data to the next node. For this reason, each party is required to assemble their own data and received data, then pass it to the next party. Hence, it is critical to carefully ensure that the leader's capability, only determining the overall intersection of all participants' data without revealing any partial intersection.

There are currently two main approaches to build MPSI protocols, partially homomorphic encryption (PHE)-based MPSI and OT-based MPSI. PHE-based protocols typically leverage polynomials or bloom filters to conceal the data sets and then share the coefficients among the participants to calculate the intersection. And PHE is employed to safeguard the security of coefficients throughout the sharing process. However, the coefficient sharing process depends on interactions among multiple participants, inherently forming a mesh topology, thus making the implementation of a ring communication topology be a tough task. On the other hand, most of the existing OT-based MPSI protocols rely on fundamental cryptographic primitives to obfuscate the data sets of the parties, such as oblivious pseudorandom function (OPRF) [13], oblivious programmable PRF (OPPRF) [3], and oblivious key-value stores (OKVS) [9]. These cryptographic primitives are inherently based on a two-party setup. As a result, to compute the intersection, it is essential to share the intermediate results produced by these primitives among various parties. The secret sharing scheme is leveraged to protect the sharing process. Unfortunately, it is still a protocol within a mesh topology, thereby becoming a bottleneck in realizing the ring topology. Consequently, we raise the following issue:

TABLE 1. COMPARISON OF SIMILAR MPSI PROTOCOLS, n IS THE NUMBER OF PARTIES, t IS THE NUMBER OF MAXIMUM COLLUSION PARTIES, N IS THE SIZE OF INPUT SETS, h IS THE NUMBER OF HASH FUNCTIONS.

Protocol	**Communication**			**Computation**	
	Topology	Leader	Assistant	Leader	Assistant
KMPRT [3]	Mesh	$\mathcal{O}(nN)$	$\mathcal{O}(tN)$	$\mathcal{O}(n)$	$\mathcal{O}(tN)$
KMS [12]	Ring	$\mathcal{O}(nN)$	$\mathcal{O}(hN)$	$\mathcal{O}(nN)$	$\mathcal{O}(hN)$
NTY [11]	Mesh	$\mathcal{O}(tN)$	$\mathcal{O}(N)$	$\mathcal{O}(nN-tN)$	$\mathcal{O}(tN)$
VCE [8]	Star	$\mathcal{O}(tN)$	$\mathcal{O}(N)$	$\mathcal{O}(nhN)$	$\mathcal{O}(N)$
Ours	Ring	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$

Can we design an efficient MPSI protocol equipped with ring topology?

1.2. Contribution

To cope with the aforementioned issue, in this paper, we design a novel MPSI protocol employing the ring topology, which significantly decreases communication and computational overhead compared to the existing state-of-the-art (SOTA) protocols. Table 1 shows a brief comparison between our MPSI and the SOTA ones on communication and computational overhead. Our contributions are summarized as follows.

- We for the first time propose the MPSI protocol with the ring topology alone. Unlike the existing MPSI protocols, Ours maintains identical and independent asymptotic complexities of communication and computation (i.e. $\mathcal{O}(N)$ and N denotes the input set size) between the leader and assistant. It effectively reduces the communication and computational burdens on the Leader, which renders him operate at the same level with assistant in regard to the bandwidth and computational capacity.
- To build the MPSI protocol with the ring topology, we present the notion of multi-point sequential oblivious pseudorandom function (MP-SOPRF). Our MP-SOPRF leverages the output of the previous round of oblivious transfer (OT) for each subsequent node and works in a multi-party setting. Besides, all participants can compute the auxiliary matrix and carry out operations of base OT simultaneously in the initial step. Then in the final OT step, the random OT (ROT) is introduced to substitute the traditional OT for improving the efficiency of protocol.
- We conducted numerous experiments to validate the effectiveness of our MPSI protocol. Results revealed that our MPSI significantly outperforms the SOTA MPSI protocols in both of communication and computational efficiencies. Concretely, it achieves communication 74.8% reduction and computational efficiency by up to $2.87\times$ in contrast to SOTA ones (15 parties and set size is 2^{20}).

2. Related work

Meadows [14] proposed the first PSI protocol. Then a series of lituratures [13], [15]–[19] explored PSI with various features and better performance. Freedman et al. [20] presented the first MPSI protocol, which is an extension of PSI in the multi-party setting. Since then, numerous efficient MPSI protocols have been proposed. They can be primarily divided into three categories, namely, public key-based MPSI, OT-based MPSI and other MPSI.

Kissner and Song [21] presented an MPSI protocol based on the polynomial root representation. In their protocol, After encoding the sets using the roots of a polynomial, each party encrypts the coefficients using an additive homomorphic encryption system, such as Paillier encryption. Li and Wu [22] employed the secret sharing scheme to replace the homomorphic encryption. Subsequently, a series of works [23], [24] made efforts to improve the Li and Wu [22]'s protocol. Ruan et al. [25] put forward the first bitset-based PSI protocol. Bay et al. [6], [7] further extended Ruan's protocol [25] to support a multi-party setting, enabling to resist the collusion attack among up to $n-1$ parties. In 2022, Vos et al. [8] strengthened the 'AND' operation for private set elements based on elliptic curves, making MPSI protocols sutiable for both of the small and large sets.

OT-based MPSI protocols utilize OT as the fundamental tool during the protocol execution. More specific, there are two types for these MPSI protocols according to the role of OT in the entire protocol. Fristly, OT works a underling building block to construct other cryptographic tools such as OPRF, OPPRF, OKVS, etc. Then these tools are involved in the MPSI protocol. Secondly, OT combined with garbled bloom filter (GBF) is utilized to build the MPSI protocol. In 2017, Kolesnikov et al. [3] brought forward an efficient MPSI protocol and provided an implementation of it. It is the first coded MPSI protocol, which is significant in the development of MPSI protocols. Chandran et al. [9] strengthened Kolesnikov et al.'s [3] protocol, removing the expensive XOR-based secret sharing scheme and then replacing it with the efficient Shamir's secret sharing scheme. Garimella et al. [10] proposed OKVS to reduce the communication costs required in OPPRF-based protocols. Nevo et al. [11] designed novel MPSI protocols based on OKVS. Their protocols consider the condition that the participants collude after being corrupted by a malicious adversary. Inbar et al. [4] provided three protocols that extend the PSI protocol based GBF [26] to a multi-party setting. Recently, Ben-Efraim et al. [27] further extended Inbar et al's protocol [4] to achieve security in the malicious model.

Apart from the aforementioned traditional MPSI protocols, there also exist MPSI protocols focusing on achieving special functionalities. Extensive works [28]–[31] explored to threshold MPSI, which adds threshold constraints to traditional multi-party PSI. In the meantime, some other works [32]–[36] studied on MPSI with cardinality (MPSI-CA), supporting multiple parties to collaboratively calculate the size of their set intersection without revealing the actual intersection elements.

3. Preliminaries

3.1. Notation

We use λ, σ to denote the computational and statistical security parameters, and use $[n]$ to represent the set of positive integers less than n, i.e., $\{1, 2, 3, \ldots, n\}$. The symbols $P_1, \ldots, P_n$ denote the set of n parties, where each party owns the input set X_i for $i \in [n]$. P_1 is assumed to be the leader who calculates the final intersection results, while other parties are regarded as assistants. For a vector v, we use $v[j]$ to denote its j-th element. For $n \times m$ size matrix M, we use M_j to denote its j-th column where $j \in [m]$. M^i represents the matrix M belongs to party P_i and $||x||_{\mathrm{H}}$ denotes the hamming weight of string x. The symbol $\mathsf{negl}(\lambda)$ denotes a negligible function that $\mathsf{negl}(\lambda) < 1/p(\lambda)$ holds for any polynomial $p(\cdot)$.

3.2. Random oblivious transfer

Oblivious Transfer (OT) [37], [38] is a fundamental cryptographic protocol allowing the sender to transfer one of multiple messages to the receiver without revealing which message was sent, and the receiver learns nothing about the other messages. Fig. 2 depicts the functionality of 1-out-of-2 OT, where the sender inputs two messages (m_0, m_1) and the receiver inputs a chosen bit c. As a result, the receiver learns m_c without knowing any information about m_{1-c} and the sender learns nothing about c. In our MPSI protocol, we introduce a variant of 1-out-of-2 OT, i.e., random OT (ROT) [39], where the sender inputs noting and the receiver inputs a chosen bit c, then the sender gets two random string (m_0, m_1) and the receiver gets m_c, thus reducing the communication overhead. Fig. 3 shows the functionality of ROT.

Parameters: Message length ℓ.
Input: The sender inputs two message (m_0, m_1) and the receiver inputs a choice bit $c \in \{0, 1\}$.
Output: the sender gets nothing and the receiver gets m_c.

Figure 2. Ideal functionality for Oblivious Transfer $\mathcal{F}_{\mathrm{OT}}$

Parameters: Message length ℓ.
Input: The sender inputs nothing and the receiver inputs a choice bit $c \in \{0, 1\}$.
Output: the sender gets message (m_0, m_1) where $m_i \in \{0, 1\}^\ell$ and the receiver gets m_c.

Figure 3. Ideal functionality for Random Oblivious Transfer $\mathcal{F}_{\mathrm{ROT}}$

3.2.1. Hamming correlation robustness. The security of our MPSI protocol relies on the *correlation robustness assumption* [15], [16], we extend this assumption to a new definition which we call *t-party* correlation robustness assumption to support the security of our protocol.

Definition 1 (Hamming Correlation Robustness). A hash function H with input length n is d-hamming correlation robust if for any $a_1, \dots, a_m, b_1, \dots, b_m \in \{0,1\}^n$ with $||b_j||_H \geq d$ where $j \in [m]$, the following distribution induced by $s \xleftarrow{\$} \{0,1\}^n$ is pseudorandom. Namely, for a random function F, we have $(H(a_1 \oplus [b_1 \cdot s]), \dots, H(a_m \oplus [b_m \cdot s])) \overset{c}{\approx} (F(a_1 \oplus [b_1 \cdot s]), \dots, F(a_m \oplus [b_m \cdot s]))$, where $\cdot$ denotes bit-wise AND and $\oplus$ denotes bit-wise XOR.

Corollary 1 (t-party Hamming Correlation Robustness). A hash function H with input length n and $t \geq 2$ parties is t-*party* d-hamming correlation robust if for any $a_1, \dots, a_m, b_1^i, \dots, b_m^i \in \{0,1\}^n$ with $||b_j^i||_H \geq d$ where $j \in [m]$ and $i \in [t-1]$, the following distribution induced by random sampling of $s_i \xleftarrow{\$} \{0,1\}^n$ where $i \in [t-1]$ is pseudorandom. Namely, for a random function F, we have

$$\begin{aligned}&(H(a_1 \oplus [b_1^1 \cdot s_1] \oplus [b_1^2 \cdot s_2] \dots \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, \\ &\quad H(a_m \oplus [b_m^1 \cdot s_1] \oplus [b_m^2 \cdot s_2] \dots \oplus [b_m^{t-1} \cdot s_{t-1}])) \\ \overset{c}{\approx}\ &(F(a_1 \oplus [b_1^1 \cdot s_1] \oplus [b_1^2 \cdot s_2] \dots \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, \\ &\quad F(a_m \oplus [b_m^1 \cdot s_1] \oplus [b_m^2 \cdot s_2] \dots \oplus [b_m^{t-1} \cdot s_{t-1}])),\end{aligned}$$

where $\cdot$ denotes bit-wise AND and $\oplus$ denotes bit-wise XOR.

Proof 1. Let $a_1, \dots, a_m, b_1^i, \dots, b_m^i \in \{0,1\}^n$ where $\forall j \in [m]$ and $\forall i \in [t-1]$, $||b_j^i||_H \geq d$ holds, set $s_i \xleftarrow{\$} \{0,1\}^n$. For $j \in [m]$, calculate $c_j = a_j \oplus [b_j^1 \cdot s_1] \oplus [b_j^2 \cdot s_2] \dots \oplus [b_j^{t-2} \cdot s_{t-2}]$, then we have $c_1, \dots, c_m, b_1^{t-1}, \dots, b_m^{t-1} \in \{0,1\}^n$ with $||b_j^{t-1}||_{\mathrm{H}} \geq d$ for $j \in [m]$ and $s_{t-1} \xleftarrow{\$} \{0,1\}^n$. This satisfies the selection range of each parameter in Definition 1. Therefore, through Definition 1, we have

$$\begin{aligned}&(H(c_1 \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, H(c_m \oplus [b_m^{t-1} \cdot s_{t-1}])) \\ \overset{c}{\approx}\ &(F(c_1 \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, F(c_m \oplus [b_m^{t-1} \cdot s_{t-1}])).\end{aligned}$$

By expanding c_j ($j \in [m]$) we can yield the equivalence relationships that follow

$$\begin{aligned}&(H(a_1 \oplus [b_1^1 \cdot s_1] \oplus [b_1^2 \cdot s_2] \dots \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, \\ &\quad H(a_m \oplus [b_m^1 \cdot s_1] \oplus [b_m^2 \cdot s_2] \dots \oplus [b_m^{t-1} \cdot s_{t-1}])) \\ \overset{c}{\approx}\ &(F(a_1 \oplus [b_1^1 \cdot s_1] \oplus [b_1^2 \cdot s_2] \dots \oplus [b_1^{t-1} \cdot s_{t-1}]), \dots, \\ &\quad F(a_m \oplus [b_m^1 \cdot s_1] \oplus [b_m^2 \cdot s_2] \dots \oplus [b_m^{t-1} \cdot s_{t-1}])),\end{aligned}$$

and this concludes the proof.

3.3. Multi-Point OPRF

Oblivious Pseudorandom Function (OPRF) combines the concepts of OT and pseudorandom function (PRF), in OPRF, the sender inputs nothing and the receiver inputs $y_1, \dots, y_n \in Y$, as a result, the sender obtains a PRF key k and the receiver obtains the PRF values $\mathsf{OPRF}_k(y_1), \dots, \mathsf{OPRF}_k(y_n)$ about its input. The sender learns nothing about the receiver's input $y_1, \dots, y_n$ and the receiver learns nothing about the PRF key k. Based on OPRF, multi-point OPRF (MP-OPRF) [16] is achieved through the following approaches: The sender with the input set X picks a random bit string $s \xleftarrow{\$} \{0,1\}^w$ of length w and the receiver with the input set Y generates two binary $m \times w$ matrices A and B. A is a random matrix, and $B = A \oplus D$ where D embeds the information of the input set Y. After running w OTs between parties where the sender acts as the OT receiver, the sender obtains a $m \times w$ matrix C with each column of C being A_i or B_i depending on the chosen bit s_i. The PRF key is the matrix C and the OPRF value ψ is calculated as $\psi = H(C_1[v[1]]|| \dots ||C_w[v[w]])$ where $v = F_k(x)$, then the sender sends ψ it to the receiver. The receiver evaluates the OPRF value as $\phi = H(A_1[v[1]]|| \dots ||A_w[v[w]])$ where $v = F_k(y)$. Consequently, $\psi = \phi$ indicates $x = y$ as $A_i[v[i]] = B_i[v[i]] = C_i[v[i]]$. Otherwise, from receiver's perspective, the value of ψ appears pseudorandom.

3.4. Security model

We adopt a semi-honest security model to assess the security of our proposed MPSI protocol. Fig. 4 shows the ideal functionality of our MPSI.

> **Parameters:** Party number n and the upper bound of party input set size N.
> **Input:** Each party P_i has an input set
> $$X_i = \{x_1^i, \dots, x_{n_i}^i\},$$
> where $x_j^i \in \{0,1\}^*$ for $j \in [n_i]$.
> **Output:** Party P_1 obtain the intersection
> $$I = X_1 \cap \dots \cap X_n.$$

Figure 4. Ideal functionality for MPSI $\mathcal{F}_{\mathrm{MPSI}}$

3.4.1. Adversary model. In the semi-honest security model, the adversary will adhere to all protocol specifications while attempting to maximize its understanding of the information it observes. The adversary's goal is to gather details regarding other participants' local data or the outcomes of the protocol. It is important to note that the leader is not colluding with any assistants, a standard condition in existing literature [40], [41]. Under this assumption, our protocol demonstrates resilience against corruption up to $n-1$.

Definition 2 (Semi-Honest Security). Let the view of P_i in protocol Π be as $view_i^{\Pi}(X_1, \dots, X_n)$. Let $\mathcal{F}(X_1, \dots, X_n)$ be the output of P_1 in ideal functionality. Let $out^{\Pi}(X_1, \dots, X_n)$ be the output of P_1 in the protocol. The protocol Π is semi-honest secure if there exist PPT simulators $\mathcal{S}_1$ and $\mathcal{S}_2$ such that for all inputs $X_1, \dots, X_n$, $\{\mathcal{S}_2(\lambda, X_i), (X_1, \cdots, X_n)\} \overset{c}{\approx} \{view_i^{\Pi}(X_1, \dots, X_n), out^{\Pi}(X_1, \dots, X_n)\}$, and for $i \in [2, n]$, $\mathcal{S}_1(\lambda, X_1, (X_1, \cdots, X_n)) \overset{c}{\approx} view_1^{\Pi}(X_1, \dots, X_n)$.

4. Construction

4.1. Technical overview

To construct a novel MPSI protocol with ring topology, we first extend the MP-OPRF protocol to the MP-SOPRF protocol to fit the multi-party setting. As MP-OPRF relies on the basic 1-out-of-2 OT, and in multi-party setting, P_i have to wait $P_1, \dots, P_{i-2}$ to complete before it can interact with P_{i-1}. Thus, to minimize the waiting time in the protocol execution, we replace traditional OT with ROT. Through ROT, both parties initiate a set of random bits r_0, r_1 based on a random bit b, where the sender gets r_0, r_1 and the receiver gets r_b, and no further interaction is needed. After that, the sender masks its own messages m_0, m_1 with r_0, r_1 as $r_0 \oplus m_0, r_1 \oplus m_1$ and sends the masked values to the receiver. Since the receiver only has r_b, it can only reveal one of the two messages $m_b = r_b \oplus c_b$. As the initialization process is independent of the messages, all participants are able to execute this process in advance (or in parallel). We use two forms of ROT here. The first form is conducted between P_1 and P_2, where P1 is the leader. In the subsequent process, P_1 chooses a random message m_0 as the input of OT. In our approach, P_1 directly utilizes the output r_0 of ROT as m_0, making the communication overhead halved at the beginning of the protocol (P_1 only needs to send c_1 to P_2). On the other hand, $P_2, \dots, P_n$ just execute the normal ROT protocol.

4.2. Construction of MP-SOPRF

We regard multi-point sequential oblivious pseudorandom function (MP-SOPRF) as a special MP-OPRF protocol for multi-party scenarios. Let X_i denote the data set of party P_i where $i \in [n-1]$. The input of MP-SOPRF is no longer a single party's data set, but multiple parties' data sets, i.e., $\{X_1, \dots, X_{n-1}\}$. After MP-SOPRF, P_1 receives the OPRF values corresponding to the data in X_1 and P_n receives the OPRF key. Let $I = \{X_1 \cap \dots \cap X_{n-1}\}$, each value obtained by P_1 is meaningful only when $x \in I$, otherwise it is a random string. Note that P_1 cannot distinguish whether the obtained values are meaningful values or random values. Assuming matrix C is a PRF key. The pseudorandom function can be computed as $\mathsf{OPRF}_C(x) = H(C_1[v[1]]||\dots||C_w[v[w]])$. Finally, P_1 obtains a matrix A and computes the output values by $\mathsf{OPRF}_A(x)$. On the other hand, P_n obtains the PRF key C. For $x \in I$, we have $\mathsf{OPRF}_A(x) = \mathsf{OPRF}_C(x)$, otherwise if $x \in X_1 \setminus I$, $\mathsf{OPRF}_C(x)$ should be a random string in P_1's view. The functionality of MP-SOPRF is defined in Fig. 5.

In MP-SOPRF, the intersection determination method is reusable, and regardless of how the random seed of the OT receiver is chosen, if the data is in the intersection, the value at position M_0 of the OT sender's inputs (M_0 and M_1) will be obtained, because when designing the OT input matrices M_0 and M_1, the values at corresponding positions in M_0 and M_1 for data mapped from the OT sender are the same. This ensures that in the OT results obtained by the OT receiver, as long as the data used belongs to the OT sender, the computed results will match the values in M_0 at the corresponding positions.

Thus, when identifying intersections, the OT sender (who acts as the PSI receiver in PSI protocols) only needs to compute and judge through its self-selected random number. This approach is also feasible in a multi-party structure. Only when the data is in the intersection of all parties, regardless of how any party's OT random seed is selected, the outcome of the entire OT link will be m_0. Finally, the leader (also referred to as the receiver in two-party PSI) only needs to match its generated random matrix A with the matrix C sent by P_n during the final intersection matching. When the leader's data x is in the intersection of all parties will occur $A(x) = C(x)$.

Input: Each party P_i has an input set X_i for $i \in [n-1]$.
Output: Party P_n obtain the MP-OPRF key k. Party P_1 obtain the MP-OPRF value set $V(x)$ for each x in his input set X_1. If and only if $x \in \{X_1 \cap \dots \cap X_{n-1}\}$, there is $\mathsf{OPRF}_k(x) = V(x)$

Figure 5. Ideal functionality for MP-SOPRF $\mathcal{F}_{\mathsf{MP\text{-}SOPRF}}$

4.3. Construction of our MPSI protocol

We first give a highlight overview of our MPSI protocol consisting of 4 parties, as is shown in Fig. 6, P_1 is the leader with data set x. P_2, P_3, P_4 are assistants and the data sets are $\{y_1, y_2\}, \{z_1, z_2\}, \{w\}$. P_1 prepare an all-ones binary matrix D and compute a vector $v = F_k(x)$. Subsequently, based on the vector v, the corresponding elements in D are set to 0 and the result is denoted as D^1. P_2 and P_3 generate matrices D^2 and D^3 using the method described above based on their own datasets. It should be noted that if a corresponding position in the matrix is already 0, no further action is needed. Firstly, P_1 randomly generates a binary matrix A and runs OT with P_2, where the OT input is $A, B = A \oplus D^1$. Let C^2 be the OT result obtained by P_2 and then P_2 runs OT with P_3, where the OT input is $C^2, E^2 = C^2 \oplus D^2$ and the OT output is C^3 . Following this approach, $P3$ and P_4 continue to run OT, with the OT input being $C^3, E^3 = C^3 \oplus D^3$ and the OT output being C^4. Afteward, P_4 computes the vector ψ using C^4 and its own data set $\{w\}$, and sends it to P_1. Finally P_1 computes $H(A(x))$ using A and its own data set $\{x\}$ and outputs x is in the intersection if $H(A(x)) = \psi$. Regardless of how the random seeds for the OT are selected by the participating parties, there are the following relationships between the matrices. $A(x) = C^2(y)$ if and only if $x = y$. $C^2(y) = C^3(z)$ if and only if $y = z$. $C^3(y) = C^4(w)$ if and only if $z = w$. Therefore, $A(x) = C^4(w)$ if and only if $x = y = z = w$ which means that x is in the set of $\{x\} \cap \{y_1, y_2\} \cap \{z_1, z_2\} \cap \{w\}$.

Generally, a multiparty PSI protocol involves n participants $P_1, \dots, P_n$, each having a dataset set X_i. They aim to compute the intersection of theses datasets $X_1 \cap \dots \cap X_n$

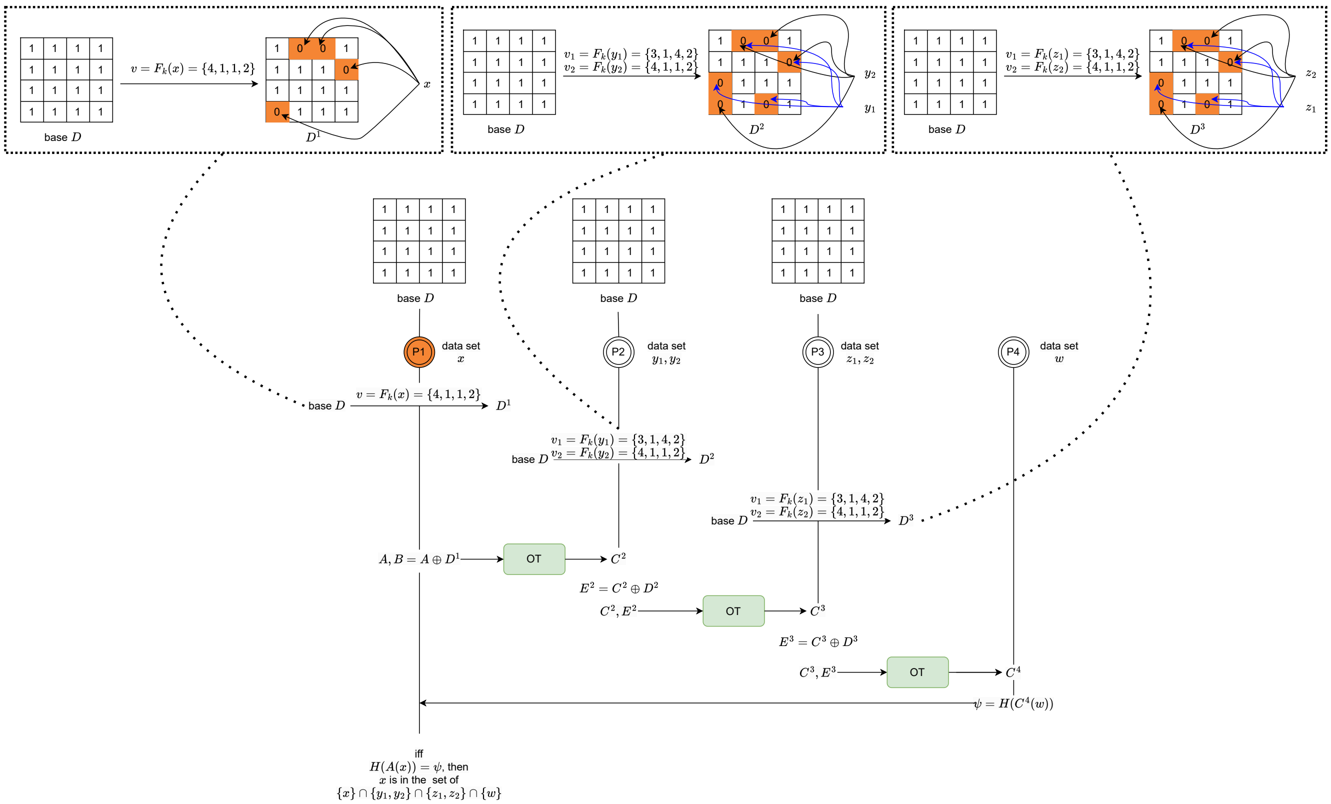


Figure 6. Overview of our MPSI protocol

without revealing any information beyond the common intersection among the parties. We consider P_1 as the leader party that gets the intersected set. we describe the flow of our protocol in Figure 7. In Step 6 and 8, involving the execution of OT protocols between two parties, to enhance efficiency, we can instantiate it using random OT. Specific procedures are described in Figures 8 and 9, respectively.

4.4. Correctness

At a high level, for each $x \in X_1$, let $v = F_k(H_1(x)$, we can get $A_j[v[j]] = B_j[v[j]]$ for all $j \in [\omega]$. After OT between P_1 and P_2, regardless of what P_2 choose for s_2, $A_j[v[j]] = B_j[v[j]] = C_j^2[v[j]]$ for all $j \in [\omega]$ iff $x \in X_1 \cap X_2$. Through a similar analysis, we can conclude that after OT between P_i and P_{i+1}, let $v = F_k(H_1(x)$ regardless of what P_{i+1} choose for s_{i+1}, $C_j^i[v[j]] = E_j^i[v[j]] = C_j^{i+1}[v[j]]$ for all $j \in [\omega]$ iff $x \in X_i \cap X_{i+1}$ for all $i \in \{2, 3, \ldots, n-1\}$. At the end of the entire protocol flow, P_n will get matrix C^n. By transitivity, let $v = F_k(H_1(x)$ we can deduce $C_j^n[v[j]] = E_j^{n-1}[v[j]] = C_j^{n-1}[v[j]] = \cdots = E_j^2[v[j]] = C_j^2[v[j]] = B_j[v[j]] = A_j[v[j]]$ for all $j \in [\omega]$ iff $x \in X_1 \cap \cdots \cap X_n$, which means that $C_j^n[v[j]] = A_j[v[j]]$. Then for each $x \in X_1$, We can conclude $x \in X_1 \cap \cdots \cap X_n$ iff $\phi \in \Psi$.

5. Security analysis

As we assume the leader will not collude with any assistant, we consider the security of our MPSI protocol under two scenarios: 1) only the leader is corrupted, and 2) the assistants are corrupted (i.e., the leader is not corrupted). Our security analysis relies on the following two lemmas [16].

Lemma 1. If F is a pseudorandom function, $H_1(x)$ is different for each $x \in X_1 \cup \cdots \cup X_n$, the probability that the number of "1"s in the sequence $D_1^i[v[1]], \ldots, D_w^i[v[w]]$ does not exceed d (the parameter of Hamming correlation robustness defined in Corollary 1) is negligible.

Lemma 2. If H_2 satisfies t-party Hamming correlation robustness, for $x \in X_1, y \in X_n \backslash I$, $x \neq y$, $v = F_k(H_1(x))$, $u = F_k(H_1(y))$, the probability that $H_2(A_1[v[1]]|| \ldots ||A_w[v[w]]) = H_2(C_1^n[u[1]]|| \ldots ||C_w^n[u[w])$ holds is negligible.

Theorem 1. If F is a pseudorandom function, H_1 is a collision resistant hash function, and H_2 is a t-party d-Hamming correlation robust hash function, then our MPSI protocol securely realizes $\mathcal{F}_{\mathrm{MPSI}}$ (Figure 4) in the semi-honest mode.

Proof 2 (P_1 is not corrupted). We construct a probabilistic polynomial time (PPT) simulator $\mathcal{S}_Z$ where Z is a subset

Parameters: Parties P_1, P_2,..., and P_n agree on security parameters λ and σ, two hash functions $H_1 : \{0,1\}^* \to \{0,1\}^{\ell_1}$ and $H_2 : \{0,1\}^{\omega} \to \{0,1\}^{\ell_2}$, pseudorandom function $F : \{0,1\}^{\lambda} \times \{0,1\}^{\ell_1} \to [m]^{\omega}$ where ℓ_1, ℓ_2, m, and ω are protocol parameters common to all parties.

Preprocessing Stage:

1) Each party P_i samples a random string $s_i \leftarrow\$ \{0,1\}^{\omega}$ as the OT receiver inputs for $i \in \{2, 3, \ldots, n\}$.
2) All participating parties share the PRF key $k \leftarrow\$ \{0,1\}^{\lambda}$.
3) Each party P_i generates an $m \times \omega$ matrix D^i with all elements equal to 1 for all $i \in [n]$.
4) For each $x \in X_i$, party P_i computes $v = F_k(H_1(x))$ and set $D^i_j[v[j]] = 0$ for all $j \in [\omega]$ where $i \in [n-1]$.

Wheeled Oblivious Transfer Interaction:

5) P_1 samples a random binary matrix $A \xleftarrow{\$} \{0,1\}^{m\times\omega}$. Computes matrix $B = A \oplus D^1$.
6) P_1 as the OT sender with input $\{A_j, B_j\}_{j\in\omega}$ and P_2 as the OT receiver with input s_2 run OT ω times. After all OTs are completed, P_2 obtains an $m \times \omega$ matrix C^2 whose column vectors is a m-bit OT result string.
7) P_2 computes $E^2 = C^2 \oplus D^2$.
8) Next each party P_i runs wheeled OT interaction for $i \in \{2, 3, \ldots, n-1\}$, which means that party P_i as the OT sender runs OT with the next party P_{i+1} as described in step 6. The OT result matrix for party P_{i+1} is denoted as C^{i+1}. When party P_i is the OT sender, his input is $\{C^i, E^i = C^i \oplus D^i\}$ where matrix C^i is the outcome of OT conducted between P_i as the OT receiver and P_{i-1}.

Intersection Result Calculation:

9) After Wheeled OT, party P_n will get a matrix C^n, then P_n computes $v = F_k(H_1(x))$ for each $x \in X_n$ and its OPRF value $\psi = H_2(C^n_1[v[1]]||\ldots||C^n_\omega[v[\omega]])$ and sends ψ to P_1.
10) The set of elements received by P_1 from P_n is denoted as Ψ. P_1 computes $v = F_k(H_1(x))$ for each $x \in X_1$ and his own OPRF value as $\phi = H_2(A_1[v[1]]||\ldots||A_\omega[v[\omega]])$. If $\phi \in \Psi$, P_1 puts x into the intersection set I.
11) Party P_1 output I as the intersection of all parties.

Figure 7. Our multiparty private set intersection protocol

1) Party P_1 and P_2 perform ω instances random OTs with message length m where P_2 is the OT receiver with input bits $s_2 \in \{0,1\}^{\omega}$. At the end of random OT, P_1 obtains ω message pairs $\{r^0_j, r^1_j\}_{j\in[\omega]}$. P_2 obtains ω random message $\{r_j\}_{j\in[\omega]}$ where $r_j = r_j^{s_2[j]}$.
2) P_1 constructs matrix A, set the column vectors of A as $\{r^0_j\}_{j\in[\omega]}$, then compute matrix $B = A \oplus D^1$. Then it calculates matrix $\Delta_j = B_j \oplus r^1_j$ for all $j \in [\omega]$ and send $\Delta = \Delta_1||\ldots||\Delta_\omega$ to P_2.
3) P_2 constructs matrix C^2 as:

$$\text{for } j \in [\omega],\ C^2_j = \begin{cases} r_j, & s_2[j] = 0, \\ r_j \oplus \Delta_j, & \text{otherwise.} \end{cases}$$

Figure 8. The instantiated using random OT in Step 6

1) P_i and P_{i+1} perform ω instances ROTs with message length m where P_{i+1} is the OT receiver with input bits $s_{i+1} \in \{0,1\}^{\omega}$. At the end of random OT, P_i obtains ω message pairs $\{r^0_j, r^1_j\}_{j\in[\omega]}$. P_{i+1} obtains ω random message $\{r_j\}_{j\in[\omega]}$ where $r_j = r_j^{s_{i+1}[j]}$.
2) P_i does following steps:
   a) Construct matrix R_0 which column vectors are $\{r^0_j\}_{j\in[\omega]}$ and matrix R_1 which column vectors are $\{r^1_j\}_{j\in[\omega]}$. Then compute matrix $E^i = C^i \oplus D^i$.
   b) Compute matrix $\Gamma = R_0 \oplus C^i$, $\Delta = R_1 \oplus E^i$ and send Δ, Γ to P_{i+1}.
3) P_{i+1} constructs matrix C^{i+1} as:

$$\text{for } j \in [\omega], C^{i+1}_j = \begin{cases} r_j \oplus \Gamma_j & \text{if} \quad s_{i+1}[j] = 0 \\ r_j \oplus \Delta_j & \text{otherwise} \end{cases}$$

Figure 9. The instantiated using random OT in Step 8

of which the parties in $P_2, \ldots, P_n$ are corrupted by the adversary. Given $\{X_i\}_{i\in Z}$, $\mathcal{S}_Z$ can generate simulated views, and these simulated views are computationally indistinguishable from the views of the joint distribution of corrupted parties in the real protocol. $\mathcal{S}_Z$ generates the random string $\{s_i \xleftarrow{\$} \{0,1\}^w\}_{i\in Z}$ honestly and chooses the random matrix $\{C^i\}_{i\in Z} \in \{0,1\}^{m\times w}$. It runs the OT simulator to simulate the view of the OT receiver for the corrupted party $P_i \in Z$ with inputs $s_i[1], \ldots, s_i[w]$ and outputs $C^i_1, \ldots, C^i_w$. Furthermore, $\mathcal{S}_Z$ sends a uniform random PRF key k to the corrupted parties. Finally $\mathcal{S}_Z$ outputs the view of corrupted parties.

We prove that $\{\mathcal{S}_Z(\lambda, \{X_i\}_{i\in Z}), \mathcal{F}(X_1, \ldots, X_n)\} \overset{c}{\approx} \{view^{\Pi}_Z(X_1, \ldots, X_n), out^{\Pi}(X_1, \ldots, X_n)\}$ via a sequence of computationally indistinguishable hybrid arguments.

$\mathbf{Hybrid_0}$: The view of corrupted parties $\{P_i\}_{i\in Z}$ and the output of P_1 in the real protocol.

$\mathbf{Hybrid_1}$: Same as $\mathbf{Hybrid_0}$ except that for party P_2, for each $j \in [w]$, if $s_2[j] = 0$, P_1 samples a random column $A_j \xleftarrow{\$} \{0,1\}^m$ and computes $B_j = A_j \oplus D^1_j$;

otherwise it samples a random column $B_j \xleftarrow{\$} \{0,1\}^m$ and computes $A_j = B_j \oplus D_j^1$. This hybrid is identical to $\mathbf{Hybrid_0}$.

$\mathbf{Hybrid_2}$: Same as $\mathbf{Hybrid_1}$ except that $\mathcal{S}_Z$ chooses a random PRF key k. This is statistically identical to $\mathbf{Hybrid_1}$.

$\mathbf{Hybrid_3}$: Same as $\mathbf{Hybrid_2}$ except that the protocol aborts if there exist $x \neq y \in X_1 \cup \dots \cup X_n$ such that $H_1(x) = H_1(y)$. This hybrid is identical to $\mathbf{Hybrid_2}$ except negligible probability because H_1 is collision resistant and then the aborting probability is negligible.

$\mathbf{Hybrid_4}$: Same as $\mathbf{Hybrid_3}$ except that the protocol aborts if there exist $x \in X_n \backslash I$, for $v = F_k(H_1(x))$, there are fewer than d 1's in $D_1^i[v[1]], \dots, D_w^i[v[w]]$ where $i \in [n-1]$. The choice of parameter m, w ensure that if F is a random function and $H_1(x)$ is different for each $x \in X_1 \cup \dots \cup X_n$, the probability of abort is negligible by Lemma 1.

$\mathbf{Hybrid_5}$: Same as $\mathbf{Hybrid_4}$ except that the output of P_1 is replaced by $\mathcal{F}(X_1, \dots, X_n) = I = X_1 \cap \dots \cap X_n$. This hybrid changes the output of P_1 iff there are $x \in X_1, y \in X_n \setminus I$, $x \neq y$, such that $v = F_k(H_1(x))$, $u = F_k(H_1(y))$, $H_2(A_1[v[1]]||\dots||A_w[v[w]]) = H_2(C_1^n[u[1]]||\dots||C_w^n[u[w]])$. Since $H_2(C_1^n[u[1]]||\dots||C_w^n[u[w])$ is pseudorandom by the t party d-Hamming correlation robustness of H_2, this hybrid is identical to $\mathbf{Hybrid_4}$ except for a negligible probability for sufficiently large ℓ_2 by Lemma 2.

$\mathbf{Hybrid_6}$: Same as $\mathbf{Hybrid_5}$ except that the protocol does not abort. From the above discussion, it is evident that the probability of a protocol abort is negligible.

$\mathbf{Hybrid_7}$: Same as $\mathbf{Hybrid_6}$ except that for corrupted parties $\{P_i\}_{i \in Z}$, $\mathcal{S}_Z$ samples the matrixes $\{C^i\}_{i \in Z}$ and runs the OT simulator to simulate the view of an OT receiver for P_i. This hybrid is computationally indistinguishable from $\mathbf{Hybrid_6}$ by the security of OT protocol.

Proof 3** (P_1 **is corrupted). We construct a PPT simulator $\mathcal{S}_1$ as follows. $\mathcal{S}_1$ generate simulated views wiht P_1's input set X_1, the max input set size N (Here we can also input the size n_n of P_n's input set, as in traditional MPSI, the size of each party's sets are not information requiring protection and do not affect the correctness of our proof or the security of the protocol) and the intersection $I = \mathcal{F}(X_1, \dots, X_n)$, and these simulated views are computationally indistinguishable from the views of P_1 in the real execution of the protocol. $\mathcal{S}_1$ computes the matrix A and B honestly and run the OT simulator to simulate the view of the OT sender for the corrupted party P_1. For each $x \in I$, it computes $v = F_k(H_1(x))$ and the OPRF value $\psi = H_2(A_1[v[1]]||\dots||A_w[v[w]])$. Let the set of OPRF values computed by $x \in I$ be Ψ_I. Subsequently it selects $N - |I|$ random ℓ_2-bit random strings and let the set of these strings be Ψ_R. In Step 9, it send $\Psi_I \cup \Psi_R$ to P_1 and finally outputs the P_1' simulated view. We prove $\mathcal{S}_1(\lambda, X_1, \mathcal{F}(X_1, \dots, X_n)) \overset{c}{\approx} view_1^{\Pi}(X_1, \dots, X_n)$ via a sequence of computationally indistinguishable hybrid argument.

$\mathbf{Hybrid_0}$: The view of P_1 in the real protocol.

$\mathbf{Hybrid_1}$: Same as $\mathbf{Hybrid_0}$ except that the protocol aborts if there exist $x \neq y \in X_1 \cup \dots \cup X_n$ such that $H_1(x) = H_1(y)$. This hybrid is identical to $\mathbf{Hybrid_0}$ except negligible probability because H_1 is collision resistant and then the aborting probability is negligible.

$\mathbf{Hybrid_2}$: Same as $\mathbf{Hybrid_1}$ except that the protocol aborts if there exist $x \in X_n \setminus I$ such that for $v = F_k(H_1(x))$, there are fewer than λ 1's in $D_1^1[v[1]], \dots, D_w^1[v[w]]$. The choice of parameter m, w ensure that if F is a random function and $H_1(x)$ is different for each $x \in X_1 \cup \dots \cup X_n$, the probability of abort is negligible by Lemma 1.

$\mathbf{Hybrid_3}$: Same as $\mathbf{Hybrid_2}$ except that $\mathcal{S}_1$ run the OT simulator to simulate the view of an OT sender for P_1. This hybrid is computationally indistinguishable from $\mathbf{Hybrid_2}$ by the security of OT protocol.

$\mathbf{Hybrid_4}$: Same as $\mathbf{Hybrid_3}$ except that for $x \in X_n \setminus I$, $\mathcal{S}_1$ replaces the OPRF value of x by ℓ_2-bit random string. This hybrid is computationally indistinguishable from $\mathbf{Hybrid_4}$ by the t party d-Hamming correlation robustness. For each $x \in X_n \setminus I$, let $v = F_k(H_1(x))$, $a = A_1[v[1]]||\dots||A_w[v[w]]$ and $b_i = D_1^i[v[1]]||\dots||D_w^i[v[w]]$ for $i \in [n-1]$. We can compute that $C_1^n[v[1]]||\dots||C_w^n[v[w]] = a \oplus [b_1 \cdot s_2] \oplus \dots \oplus [b_{n-1} \cdot s_n]$. Since $||b_i||_H \geq d$ and s_{i+1} is randomly sampled and unknown to P_1 for $i \in [n-1]$, the OPRF value of x which is $H_2(C_1^n[v[1]]||\dots||C_w^n[v[w]])$ is pseudorandom for P_1 by the t party d-Hamming correlation robustness of H_2.

$\mathbf{Hybrid_5}$: Same as $\mathbf{Hybrid_4}$ except that the protocol does not abort. From the above discussion, it is evident that the probability of a protocol abort is negligible. This hybrid is P_1's view simulated by $\mathcal{S}_1$.

6. Experiments

6.1. Theoretical analysis

6.1.1. Choice of m, w. The parameter m, w in our protocol satisfy that if F is a random function and $H_1(x)$ is different for each $x \in X_1 \cup \dots \cup X_n$, then for $x \in X_n \setminus I$ and $v = F_k(H_1(x))$, there are at least λ 1's in $D_1^1[v[1]], \dots, D_w^1[v[w]]$ with all but negligible probability. This is to ensure that in the final step of the protocol, P_1 cannot brute-force information beyond obtaining the intersection from the OPRF values. In other words, the OPRF values for any item in the P_n's input set which is not in the intersection I remain pseudorandom to P_1. This is achieved because of the correlation robustness property of H_2. We will determine the value of m (which is typically N in this protocol) and then compute w as follows. The column D_i was initialized as 1^m, then for each $x \in X_1$, P_1 computes $v = F_k(H_1(x))$ and sets $D_i[v[i]] = 0$. Since F is a random function and $H_1(x)$ is different for $x \in X_1$,

TABLE 2. RUNNING TIME (IN MILLISECOND) OF THE MPSI PROTOCOLS IN LAN AND WAN SETTINGS.

Set size	Protocol	Total running time for different parties(ms)													
		LAN (5000M, 0.2ms RTT)							WAN (200M, 2ms RTT)						
		4	6	8	10	12	14	15	4	6	8	10	12	14	15
2^{12}	KMPRT [3]	364	499	864	1140	1524	1986	2268	947	2153	4046	6629	9881	13788	16009
	KMPRT(AUG) [3]	400	431	482	499	589	618	625	1592	2274	3092	3880	4679	5498	5949
	NTY [11]	232	238	198	260	297	320	395	365	557	728	893	1114	1345	1434
	Ours	363	374	394	466	457	465	728	366	417	493	595	660	796	673
2^{13}	KMPRT [3]	551	885	1435	1853	2488	3132	3602	1831	4537	8598	14061	20909	29186	33839
	KMPRT(AUG) [3]	681	802	833	915	1037	1141	1219	3417	5197	6689	8488	10231	12072	12937
	NTY [11]	229	294	323	339	414	427	516	599	1041	1355	1726	2122	2645	2754
	Ours	395	453	428	527	538	524	585	439	575	669	790	886	991	1315
2^{14}	KMPRT [3]	996	1521	2326	3500	4525	5715	6327	3547	8641	16809	27696	41296	57663	66818
	KMPRT(AUG) [3]	1288	1434	1585	1742	1893	2088	2171	6194	9663	13252	16774	20298	23850	25753
	NTY [11]	334	508	542	567	632	753	686	1086	1751	2541	3295	4055	4964	5388
	Ours	475	548	559	600	594	632	969	646	868	988	1221	2160	2660	3339
2^{15}	KMPRT [3]	1735	3077	4600	6597	8806	10552	11882	6869	17035	33101	54600	81490	113746	131870
	KMPRT(AUG) [3]	2645	2882	3259	3568	3730	4236	4273	13516	20494	26359	33339	40494	47818	51368
	NTY [11]	593	819	919	960	1134	1157	1255	2103	3363	4935	6447	7997	9572	10398
	Ours	635	663	701	764	1521	2058	2070	952	1330	1868	2164	3356	3362	4801
2^{16}	KMPRT [3]	3087	5982	9854	12995	16776	20642	23348	13788	34382	66066	108869	162357	226912	262982
	KMPRT(AUG) [3]	5482	5929	6477	7285	7543	8235	8788	26920	38385	52767	66796	81011	95177	101991
	NTY [11]	1083	1348	1620	1769	1936	2225	2308	3993	6700	9717	12777	15821	19019	20559
	Ours	1003	1037	1070	3646	3435	3522	4242	1633	2409	4601	4259	6054	8599	9255
2^{17}	KMPRT [3]	6549	11218	17340	24655	32832	41030	44658	29658	75154	144507	238440	355742	496675	576085
	KMPRT(AUG) [3]	10664	12934	13655	15498	17592	18354	19229	59653	85886	116037	147080	178280	209563	225231
	NTY [11]	2133	2505	3041	3225	3798	4166	4474	8628	14166	20867	27532	34252	40943	44400
	Ours	2814	3598	3672	4582	5293	5585	5465	6367	7335	8300	9809	12753	14148	14530
2^{18}	KMPRT [3]	12338	21912	35974	49526	63299	81296	91360	60064	148683	288943	476566	711234	992698	1152083
	KMPRT(AUG) [3]	24767	26423	29833	32954	35867	39160	39499	119125	180798	232095	295276	356877	423846	452325
	NTY [11]	4425	5161	5770	6540	7831	8930	9634	17246	28314	41741	54827	68289	81717	88472
	Ours	3205	4133	4480	6136	6929	7902	8085	7503	12085	13582	18579	22203	24799	27074
2^{19}	KMPRT [3]	27061	42049	70068	98985	131225	164695	180946	122911	301132	578093	952718	1421702	1985126	2301667
	KMPRT(AUG) [3]	50014	64313	65266	67896	75725	77460	86233	246857	361365	488146	618449	746404	849883	934572
	NTY [11]	9056	10179	11537	13194	15445	17475	18861	34880	57035	83062	109812	136550	165247	176370
	Ours	8476	8540	8653	8698	8692	9476	9970	12193	21440	23439	32219	39165	45423	48800
2^{20}	KMPRT [3]	63815	85631	140710	204628	256021	325696	-	247219	596205	1159041	1908268	2845787	3974837	-
	KMPRT(AUG) [3]	122619	116997	140912	148125	157234	170809	182509	483092	733914	982592	1248201	1501481	1722380	1878569
	NTY [11]	19177	21927	25099	30163	33463	36118	38696	71108	123544	167425	219416	276702	339043	371039
	Ours	13109	13859	14170	15284	16755	16893	16753	27000	39202	50541	62909	75512	89505	95911

v is random and independent for $x \in X_1$ which means that the probability $\Pr[D_i^1[j] = 1]$ is same for all $j \in [m]$. especially, $\Pr[D_i^1[j] = 1] = (1 - \frac{1}{m})^{n_1}$. Let $p = (1 - \frac{1}{m})^{n_1}$. For any $x \in X_n \setminus I$, the probability that there are d 1's in $D_1^1[v[1]], \ldots, D_w^1[v[w]]$ is

$$\binom{w}{d} p^d (1-p)^{w-d},$$

since $\Pr[D_i^1[v[i]] = 1] = p$ is independent for all $i \in [w]$. Then for $x \in X_n \setminus I$, the proper w that there are at least d 1's with all but negligible probability can be computed by the union bound:

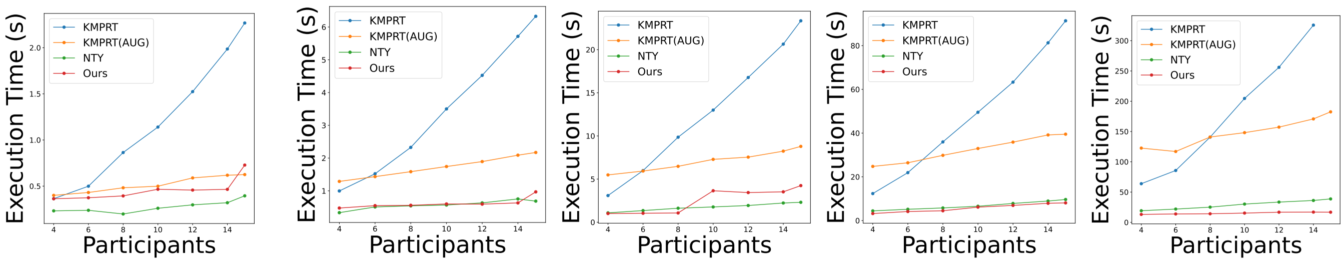
$$N \cdot \sum_{k=0}^{d-1} \binom{w}{k} p^k (1-p)^{w-k} \leq \mathsf{negl}(\sigma).$$

6.1.2. Choice of ℓ_1 and ℓ_2. The parameters ℓ_1 and ℓ_2 are respectively the output lengths of hash functions H_1 and H_2. We need to set $\ell_1 = 2\lambda$ to guarantee collision resistance against the birthday attack and set $\ell_2 = \sigma + 2\log(N)$ to guarantee that the probability of collision in the MPSI protocol (Step 10) is negligible for a semi-honest model, similarly as in [15], [16].

6.1.3. Complexity analysis. In our MPSI protocol, we rely solely on lightweight cryptographic tools such at OT extension, hash functions, AES and bitwise operations. Without loss of generality, we consider the upper bound on the input set sizes of the parties as N^2 and set $m = N$ as in [16]. For the fixed m, N and security parameter λ, we can compute w, ℓ_2 using the method mentioned in Section 6.1.

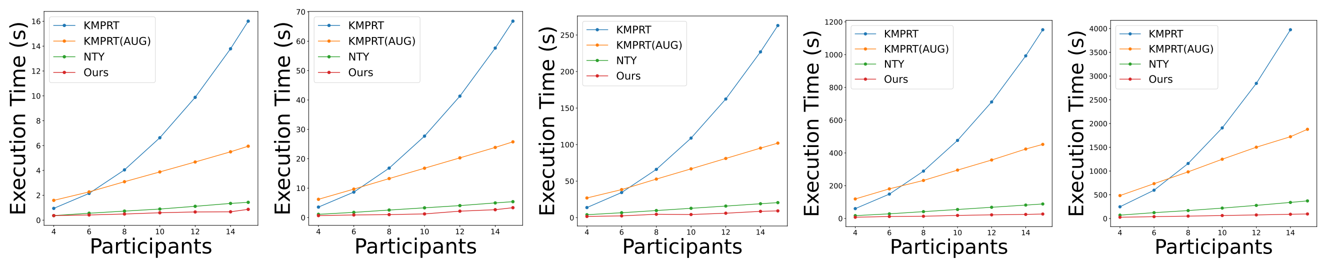
We denote party P_1 as the leader and party $P_2, \ldots, P_{n-1}$ as the assistant. P_1 generates matrix A and B which cost linear complexity in N. Then P_1 runs w OTs with P_2 that cost linear complexity in N. In this step, we significantly reduce

2. Typically, during testing, we assume that the size of the sets of all participants is N. Participants with sets smaller than N can expand their set sizes to N by adding random data.



(a) $|S| = 2^{12}$ (b) $|S| = 2^{14}$ (c) $|S| = 2^{16}$ (d) $|S| = 2^{18}$ (e) $|S| = 2^{20}$

Figure 10. Comparison of MPSI protocols under different data set sizes in LAN setting



(a) $|S| = 2^{12}$ (b) $|S| = 2^{14}$ (c) $|S| = 2^{16}$ (d) $|S| = 2^{18}$ (e) $|S| = 2^{20}$

Figure 11. Comparison of MPSI protocols under different data set sizes in WAN setting

communication and computational overhead by adopting the random OT scheme illustrated in Figure 8. In the final intersection calculation stage, P_1 only needs to perform hash functions and bitwise operation with a complexity of N. For party $P_2, \dots, P_{n-1}$, they first act as the OT receiver and run OT to get the matrix C^i for $i \in \{2, \dots, n-1\}$. Then they generate matrix D^i and run w OTs with P_{i+1} and use random OT in Figure 9 to decrease communication and computational overhead which cost linear complexity in N. P_n finally compute the OPRF value by hash functions and bitwise operation.

Our protocol is currently the only one under the semi-honest model where the leader's and assistant's communication and computational complexities are equal. In traditional protocols, the leader typically bears a significant burden of communication and computation, which can impose considerable bandwidth and computational pressure on the leader. The main reason for this situation is that most existing protocols predominantly employ mesh and star topology, which contribute to the imbalance in communication and computation workload, primarily borne by the leader. Kavousi et al. in [12] proposed a protocol based on a star + wheel structure. In their protocol, all parties communicate with the leader once in the first step, followed by the protocol running in a wheel-like structure. Our protocol is a simple ring structure, where each participant communicates once with its adjacent party. This allows our protocol's communication and computational complexity for the leader and assistant to depend solely on the size of the input set, independent of the number of parties.

In table 1, we compared the communication and computational complexity of our protocol with several currently efficient protocols [3], [8], [11], [12]. The criteria we selected for comparison include two points: first, the protocol needs to provide up to $n-1$ collusion resistance similar to our protocol, and second, the protocol needs to demonstrate high operational efficiency.

6.2. Experimental analysis

6.2.1. Experimental setup. Our MPSI was implemented based on libOTe (https://github.com/osu-crypto/libOTe) using C++-11. We compared our MPSI with the state-of-the-art ones, i.e., the KMPRT protocol [3] and the NTY protocol [11]. All protocols in comparison were run on Ubuntu 22.04 servers with 2× Intel(R) Xeon(R) 2.10 GHz CPU and 256GB RAM. The input length was set to be 128 bits, the statistical security parameter is $\kappa = 40$, and the computational security parameter is $\lambda = 128$. All protocols were tested in both the local area network (LAN) and wide area network (WAN) settings, the LAN setting has 0.2 milliseconds round-trip latency and 5000 Mbps network bandwidth, and the WAN setting has 2 milliseconds round-trip latency and 200 Mbps network bandwidth. The dataset sizes ranges from 2^{12} to 2^{20}, the number of parties ranges from 4 to 15.

6.2.2. Comparison on total running time. Table 2 shows the total running time of the related protocols. It is evident that as the dataset size expands, our MPSI protocol not only maintains a shorter overall execution time compared to other protocols but also increasingly outperforms them.

TABLE 3. RUNNING TIME (IN MILLISECOND) THE MPSI PROTOCOLS OVER LARGE DATA SETS

Set size	Protocol	Total running time for different parties(ms)									
		LAN (5000M, 0.2 ms RTT)					WAN (200M, 2 ms RTT)				
		$\|\mathcal{C}\| = 16$	$\|\mathcal{C}\| = 17$	$\|\mathcal{C}\| = 18$	$\|\mathcal{C}\| = 19$	$\|\mathcal{C}\| = 20$	$\|\mathcal{C}\| = 16$	$\|\mathcal{C}\| = 17$	$\|\mathcal{C}\| = 18$	$\|\mathcal{C}\| = 19$	$\|\mathcal{C}\| = 20$
2^{20}	NTY [11]	43076	46154	49192	50213	55077	383345	409400	434396	463339	492568
	Ours	15094	15887	16185	15940	17154	101832	108228	113779	119942	125969
2^{21}	NTY [11]	91441	92183	99581	109904	107423	837114	902238	946369	998135	1069145
	Ours	29694	33125	30514	29988	31427	200637	213479	228281	238873	253829
2^{22}	NTY [11]	-	-	-	-	-	-	-	-	-	-
	Ours	64822	67968	66434	69497	67659	406194	430619	457885	485455	511640
2^{23}	NTY [11]	-	-	-	-	-	-	-	-	-	-
	Ours	138428	137397	136229	138187	138208	809523	862368	916236	970604	1022665
2^{24}	NTY [11]	-	-	-	-	-	-	-	-	-	-
	Ours	282774	281785	288982	281654	295784	1624301	1721287	1842714	1949102	2053595

TABLE 4. COMMUNICATION COST (IN MB) OF MPSI PROTOCOLS.

Set size	**Protocol**	**Communication cost (MB)**		
		$\|\mathcal{C}\| = 4$	$\|\mathcal{C}\| = 10$	$\|\mathcal{C}\| = 15$
2^{12}	KMPRT [3]	19.68	147.6	344.4
	KMPRT(AUG) [3]	7.36	18.4	27.75
	NTY [11]	7.36	18.4	27.75
	Ours	**1.6**	**5.20**	**8.24**
2^{16}	KMPRT [3]	311.2	2334.1	5446.35
	KMPRT(AUG) [3]	126.76	316.9	475.35
	NTY [11]	126.76	316.9	475.35
	Ours	**21.53**	**79.52**	**126.4**
2^{20}	KMPRT [3]	5608	42080	98205
	KMPRT(AUG) [3]	2225.56	5563.9	8345.85
	NTY [11]	2225.56	5563.9	8345.85
	Ours	**357.84**	**1253.42**	**2106.23**

This improvement is primarily attributed to the substantial reduction in communication overhead achieved by our MPSI scheme as the dataset grows, as clearly demonstrated by comparing experimental results in WAN and LAN environments. Fig. 10 and Fig. 11 further illustrates the comparative trends of the MPSI protocol's execution time as the number of participants increases across various dataset sizes in the LAN and WAN settings, respectively. It is evident that the execution time of our MPSI protocol grows linearly at a very slow pace. Particularly in WAN settings, our protocol yields the best results compared to all others.

6.2.3. Comparison over large data sets. For data sets over the size of 2^{20}, and for the number of participants, values greater than or equal to 16. We only selected NTY [11] and our own protocol for comparison. The NTY protocols was unable to run due to insufficient memory when the data volume exceeded 21. The experimental results are shown in Table 3.

6.2.4. Comparison on communication overhead. Table 4 demonstrates that our scheme substantially reduces communication overhead. Specifically, with a dataset size of 2^{20} and 15 participants, our protocol achieves a 97.8% reduction compared to the KMPRT protocol and a 74.8% reduction compared to the NTY protocol.

7. Conclusion

In this paper, we propose an efficient Multi-Party Private Set Intersection (MPSI) protocol aimed at addressing the bottlenecks in computational and communication overhead in existing MPSI protocols. By constructing a new multi-party sequential oblivious pseudorandom function (MP-SOPRF), our protocol not only enhances computational efficiency but also significantly reduces the communication cost required for multi-party interactions. Our protocol is the first MPSI protocol based solely on a ring topology. Experiment results show that our MPSI achieves an approximate $2.87\times$ acceleration in total running time and achieves a 74.8% reduction about communication, which is crucial for large-scale data set applications.

References

[1] L. Lu and N. Ding, "Multi-party private set intersection in vertical federated learning," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 707–714.

[2] A. Miyaji, K. Nakasho, and S. Nishida, "Privacy-preserving integration of medical data: a practical multiparty private set intersection," *Journal of medical systems*, vol. 41, pp. 1–10, 2017.

[3] V. Kolesnikov, N. Matania, B. Pinkas, M. Rosulek, and N. Trieu, "Practical multi-party private set intersection from symmetric-key techniques," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1257–1272.

[4] R. Inbar, E. Omri, and B. Pinkas, "Efficient scalable multiparty private set-intersection via garbled bloom filters," in *International conference on security and cryptography for networks*. Springer, 2018, pp. 235–252.

[5] J. Vos, M. Conti, and Z. Erkin, "Sok: Collusion-resistant multi-party private set intersections in the semi-honest model," *Cryptology ePrint Archive*, 2023, unpublished.

[6] A. Bay, Z. Erkin, M. Alishahi, and J. Vos, "Multi-party private set intersection protocols for practical applications," in *18th International Conference on Security and Cryptography, SECRYPT 2021*. SciTePress, 2021, pp. 515–522.

[7] A. Bay, Z. Erkin, J.-H. Hoepman, S. Samardjiska, and J. Vos, "Practical multi-party private set intersection protocols," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1–15, 2021.

[8] J. Vos, M. Conti, and Z. Erkin, "Fast multi-party private set operations in the star topology from secure ands and ors," *Cryptology ePrint Archive*, 2022, unpublished.

[9] N. Chandran, N. Dasgupta, D. Gupta, S. L. B. Obbattu, S. Sekar, and A. Shah, "Efficient linear multiparty psi and extensions to circuit/quorum psi," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1182–1204.

[10] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "Oblivious key-value stores and amplification for private set intersection," in *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II 41*. Springer, 2021, pp. 395–425.

[11] O. Nevo, N. Trieu, and A. Yanai, "Simple, fast malicious multiparty private set intersection," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1151–1165.

[12] A. Kavousi, J. Mohajeri, and M. Salmasizadeh, "Efficient scalable multi-party private set intersection using oblivious prf," in *Security and Trust Management: 17th International Workshop, STM 2021, Darmstadt, Germany, October 8, 2021, Proceedings 17*. Springer, 2021, pp. 81–99.

[13] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, "Efficient batched oblivious prf with applications to private set intersection," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 818–829.

[14] C. Meadows, "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*. IEEE, 1986, pp. 134–134.

[15] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "Spot-light: lightweight private set intersection from sparse ot extension," in *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39*. Springer, 2019, pp. 401–431.

[16] M. Chase and P. Miao, "Private set intersection in the internet setting from lightweight oblivious prf," in *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part III 40*. Springer, 2020, pp. 34–63.

[17] P. Rindal and P. Schoppmann, "Vole-psi: fast oprf and circuit-psi from vector-ole," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2021, pp. 901–930.

[18] S. Raghuraman and P. Rindal, "Blazing fast psi from improved okvs and subfield vole," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2505–2517.

[19] D. Bui and G. Couteau, "Private set intersection from pseudorandom correlation generators." *IACR Cryptol. ePrint Arch.*, vol. 2022, p. 334, 2022, unpublished.

[20] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2*. Springer, 2005, pp. 303–324.

[21] L. Kissner and D. Song, "Privacy-preserving set operations," in *Annual International Cryptology Conference*. Springer, 2005, pp. 241–257.

[22] R. Li and C. Wu, "An unconditionally secure protocol for multi-party set intersection," in *International Conference on Applied Cryptography and Network Security*. Springer, 2007, pp. 226–236.

[23] A. Patra, A. Choudhary, and C. P. Rangan, "Information theoretically secure multi party set intersection re-visited," in *International Workshop on Selected Areas in Cryptography*. Springer, 2009, pp. 71–91.

[24] ——, "Round efficient unconditionally secure mpc and multiparty set intersection with optimal resilience," in *International Conference on Cryptology in India*. Springer, 2009, pp. 398–417.

[25] O. Ruan, Z. Wang, J. Mi, and M. Zhang, "New approach to set representation and practical private set-intersection protocols," *IEEE Access*, vol. 7, pp. 64 897–64 906, 2019.

[26] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: an efficient and scalable protocol," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 789–800.

[27] A. Ben-Efraim, O. Nissenbaum, E. Omri, and A. Paskin-Cherniavsky, "Psimple: Practical multiparty maliciously-secure private set intersection," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 1098–1112.

[28] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 1–19.

[29] S. Ghosh and M. Simkin, "The communication complexity of threshold private set intersection," in *Annual International Cryptology Conference*. Springer, 2019, pp. 3–29.

[30] S. Badrinarayanan, P. Miao, S. Raghuraman, and P. Rindal, "Multi-party threshold private set intersection with sublinear communication," in *IACR International Conference on Public-Key Cryptography*. Springer, 2021, pp. 349–379.

[31] F.-H. Liu, E. Zhang, and L. Qin, "Efficient multiparty probabilistic threshold private set intersection," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 2188–2201.

[32] P. Mohassel, P. Rindal, and M. Rosulek, "Fast database joins and psi for secret shared data," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1271–1287.

[33] S. K. Debnath, P. Stănică, N. Kundu, and T. Choudhury, "Secure and efficient multiparty private set intersection cardinality." *Advances in Mathematics of Communications*, vol. 15, no. 2, 2021.

[34] E. Fenske, A. Mani, A. Johnson, and M. Sherr, "Accountable private set cardinality for distributed measurement," *ACM Transactions on Privacy and Security*, vol. 25, no. 4, pp. 1–35, 2022.

[35] N. Trieu, A. Yanai, and J. Gao, "Multiparty private set intersection cardinality and its applications." *IACR Cryptol. ePrint Arch.*, vol. 2022, p. 735, 2022.

[36] Y. Yang, X. Dong, Z. Cao, J. Shen, R. Li, Y. Yang, and S. Dou, "Empsi: Efficient multiparty private set intersection (with cardinality)," *Frontiers of Computer Science*, vol. 18, no. 1, p. 181804, 2024.

[37] M. O. Rabin, "How to exchange secrets with oblivious transfer," *Cryptology ePrint Archive*, 2005, unpublished.

[38] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols." in *SODA*, vol. 1, 2001, pp. 448–457.

[39] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology—CRYPTO'91: Proceedings 11*. Springer, 1992, pp. 420–432.

[40] E. Zhang, F.-H. Liu, Q. Lai, G. Jin, and Y. Li, "Efficient multi-party private set intersection against malicious adversaries," in *Proceedings of the 2019 ACM SIGSAC conference on cloud computing security workshop*, 2019, pp. 93–104.

[41] A. Abadi, S. Terzis, R. Metere, and C. Dong, "Efficient delegated private set intersection on outsourced private datasets," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 608–624, 2017.