A ROUND-OPTIMAL NEAR-LINEAR THIRD-PARTY PRIVATE SET INTERSECTION PROTOCOL

FOO YEE YEO AND JASON H. M. YING

Abstract. Third-party private set intersection (PSI) enables two parties, each holding a private set to compute their intersection and reveal the result only to an inputless third party. In this paper, we present an efficient round-optimal third-party PSI protocol. Our work is motivated by real-world applications such as contact tracing whereby expedition is essential while concurrently preserving privacy. Our construction only requires 2 communication rounds and attains a near-linear computational complexity of $O(n^{1+\varepsilon})$ for large dataset size n, where $\varepsilon > 0$ is any fixed constant. Our improvements stem from algorithmic changes and the incorporation of new techniques along with precise parameter selections to achieve a tight asymptotic bound. Furthermore, we also present a third-party PSI cardinality protocol which has not been explored in prior third-party PSI work. In a third-party PSI cardinality setting, only the third-party obtains the size of the intersection and nothing else. Our construction to achieve the cardinality functionality attains a quasilinear computational complexity for the third-party.

1. Introduction

Private set intersection (PSI) [30] is a cryptographic primitive used for secure computation, which allows two or more parties to compute the intersection of their sets while keeping their inputs secret. The applications of PSI arise in numerous diverse settings ranging from botnet detection [32], private proximity testing [33], human genomes testing [2], private contact discovery [23], online advertising [37], as well as contact tracing [14, 45] in the event of a pandemic such as COVID-19. Due to its wide range of applications, a long series of notable works [15, 39, 37, 26, 9, 48, 40, 49, 38, 36, 31, 7, 43, 16, 17, 29, 46, 19, 18] have been carried out to advance the development of efficient PSI protocols in both the theoretical and practical aspects.

Existing PSI solutions can be broadly classified from a variety of approaches. The initial constructions of PSI arose from Diffie-Hellman based oblivious pseudorandom functions (OPRFs) [30]. There exist several modern protocols [4, 20] which are designed based upon DH-OPRF due to the low communication cost which it offers. Oblivious transfer (OT) extension first introduced in [21], followed by improvements due to [1], enables computation of a very large number of OTs at low cost by using just a relatively small number of base-OTs. OT extensions engendered a class of protocols [26, 42, 40, 36], which provide a lower computational cost with a higher communication overhead trade-off as compared to DH-OPRF approaches. Homomorphic encryption (HE) is a core building block in several PSI protocols. The PSI protocol [15] applies oblivious polynomial evaluation by utilizing an additive partially homomorphic encryption scheme, such as the Paillier cryptosystem [35]. The work in [9] is based on leveled HE and applies techniques such as batching to reduce the communication cost. Fully homomorphic encryption (FHE) is employed in the works of [8, 12] for a labeled PSI setting, where the sender holds a label associated with each item, and the functionality outputs the labels from the items in the intersection to the receiver. FHE is also applied in the work of [19] to compute a variety of enhanced functionalities over the set intersection. General HE techniques are computationally expensive but can be useful in certain scenarios such as in unbalanced PSI where one party's set is significantly smaller than the other. Circuit-based PSI [38, 43, 6] has the added potential to privately compute functions over the set intersection but requires many communication rounds. Hashing techniques have been used by some PSI protocols [15, 39, 37] to reduce the number of comparisons performed between the set elements to obtain the intersection, thereby achieving higher efficiency.

Third-Party PSI. Yeo and Ying [47] introduced a variant of PSI, known as third-party private set intersection, that enables the private computation of the intersection of datasets held by two different parties P_1 and P_2, while revealing the result only to an inputless third party Q. A key challenge in efficiently achieving third-party PSI comes from the observation that the inputless third party Q does not himself have any information that can be used to constrain the elements that might appear in the intersection.

1.1. **Motivation and Use Cases.** Third-party PSI possesses practical utility and is relevant in settings when the intersection output is only made known to a third-party for privacy reasons. Instances of such scenarios occur when a regulatory authority intends to obtain relevant information from two organizations. A third-party PSI protocol prevents sensitive information from being exposed to the participating parties, while enabling the regulator to achieve the intended objective. For example, in the event of a disease outbreak during a pandemic, it is essential for the public health authority to be able to quickly identify potential asymptomatic sources of transmission. In this situation, the public health authority assumes the role of the third-party while the premises which have records of the people who visit along with their time stamps are the participating parties. This allows the health regulatory authority to easily obtain a database of people who are present at both locations at specific times in a privacy-preserving manner.

1.2. **Related Work and Challenges.** It should be noted that existing PSI protocols with applications to contact tracing operate in a different context. In most settings being considered, there are two main roles, one sender and one receiver, whereby conventional PSI protocols can be applied more directly. For instance, in the use cases of [14, 45], the receiver is a user who holds a set of identifiers within proximity while the sender is the public health authority which is assumed to already own a database of contact tokens from infected users through prior collection. The user can then perform a PSI protocol with the public health authority to determine the extent of exposure with other infected individuals. To minimize workload on the receiver, which is typically a user's mobile device, the protocol of [14] delegates a majority of the user-side computations to untrusted servers. In our use case, there are two senders and one inputless receiver whereby the third-party role of the public health authority seeks to gather the database of potential individuals at risk in the event of outbreaks at the premises.

There are other variants of PSI, such as utilizing a server to either increase efficiency [28] or to outsource the computational workload [25], as well as multi-party PSI [27, 6, 34] which can be regarded as a generalization of conventional two-party PSI, where there are more than two participants' sets to compute over.

However, these do not provide effective solutions to the specified task. In the server aided setting, the receiver is the party with the inputs, while the receiver is inputless in third-party PSI. The latter results in a more complex problem when the participating parties with inputs are not allowed to obtain any information throughout the process. In the case of muti-party PSI, one can adopt a solution by assigning the third-party the entire universe of possible input elements, but this is clearly not ideal both in theory and in practice.

In [47], Yeo and Ying introduced two different third-party PSI protocols, the first based on the use of Diffie-Hellman, and the second based on the use of a generic key agreement protocol. Both protocols are communication efficient, requiring only a low amount of communication. However, the protocols can incur significant computational costs.

Let us briefly explain the main ideas behind Protocol 2 of [47], which is itself based on techniques from a PSI protocol introduced by Rosulek and Trieu [44]. Suppose P_1 and P_2 have sets S_1 and S_2 respectively. Essentially, the protocol carries out a key exchange for each element in S_2, such that the key exchange succeeds if and only if the element also lies in S_1.

In the protocol, each key exchange is associated to some element of S_2. To keep S_2 private, P_2 hides these elements by encoding the key exchange messages into a polynomial using polynomial interpolation. The set of keys that should have been obtained if the key exchanges were carried out successfully are also encoded by P_2 into a polynomial q, while the set of keys K obtained by P_1 is sent to Q.

If S_1 and S_2 contain some common element s_i, then the key obtained by P_2 that is associated to s_i will be in the set K. The most expensive part of the protocol lies in the final step, in which the third party Q solves $q(t) = k_i$ for each $k_i \in K$ to obtain the desired intersection.

There are other possible solutions to the third-party PSI problem. One possible solution is to use circuit PSI [43, 6, 41, 3], in which the parties' outputs are shares of the intersection result. By having the parties P_1 and P_2 run a two-party circuit PSI protocol, and then provide their output shares of the intersection result to the third-party Q, we obtain a solution to the third-party PSI problem. However, in general, circuit PSI suffers from the disadvantage of having a high round complexity. While some works on circuit PSI have constant round complexity, they still require many rounds of communication.

Another possible solution to the third-party PSI problem is to apply a generic three-party secure multi-party computation (MPC) protocol. In fact, a round-optimal solution for the third-party PSI problem can be obtained using the generic MPC protocol of Ishai et al [22]. Such a solution, however, incurs high computational complexity.

In this work, we ask the question:

> *Can round-optimal private set intersection protocols having near-linear computational and communication complexities be achieved in the third-party setting?*

Attaining near-linear complexities is of practical importance for large set sizes based on the highlighted applications while achieving a round-optimal complexity is of theoretical relevance. The results in this paper present techniques in enabling the first construction of a round-optimal third-party private set intersection protocol which attains near-linear complexities in both computation and communication.

1.3. **Our Contributions.** We improve upon the current state-of-the-art for third-party PSI, by introducing a 2-round protocol with near-linear computational and communication complexity. This provides a significant improvement over the round complexity of the 4-round Diffie-Hellman based protocol in [47], and achieves a reduction in the computational cost of the key agreement based protocol in [47] from $O(n^{2.5+o(1)})$ to $O(n^{1+\varepsilon})$, where ε is any positive constant and n is the size of the each dataset.[1] Our solution is round-optimal in contrast to the many rounds of communication required when using a circuit-PSI approach, while our solution is much more computationally efficient compared to a solution using generic MPC.

In concurrent work, Chen et al. [10] introduced two new approaches for third-party PSI, one based on homomorphic encryption and the other using OPRFs. While the authors of [10] claim a round complexity of 2 rounds for both protocols, both their protocols in fact require 3 rounds. Indeed, for the homomorphic encryption approach, one communication round is needed for Q to send his public key to P_1 and P_2 giving a total of 3 rounds for the protocol; while for the OPRF approach, a total of 3 rounds is also required as the OPRF takes a minimum of 2 communication rounds.

We achieve our round-optimal near-linear third-party PSI protocol by incorporating multiple improvements to the key agreement based protocol in [47]. First, by using a single-round key agreement protocol instead of a two-round key agreement protocol, we can combine some of the steps in the protocol in [47] to save a communication round. Second, we modify the protocol such that the set of keys K computed by P_1 during the execution of the protocol is not sent directly to Q, but rather, it is used to interpolate a polynomial which is then sent to Q. By making this modification, when Q is computing the intersection, it suffices for Q to compute the roots of a single polynomial, rather than roots of n different polynomials, thus saving a factor of approximately n in the computational cost. Third, we use a hash function to hash the inputs of the parties into some number of bins, before applying the protocol separately to each of these bins. This reduces the size of the instances on which we apply the existing third-party PSI protocol at the cost of having to perform a large number of instances of the original protocol. With a careful choice of parameters, we achieve a protocol with a greatly reduced computational cost compared to the original protocol.

In addition, we consider a variant of the third-party PSI problem, where the aim is to privately compute the size, but not the exact contents, of the intersection of datasets held by P_1 and P_2, again revealing the result only to Q. In the conventional two-party setting, this problem, known as PSI cardinality, was introduced and studied by Cristofaro et al. [13]. We introduce a protocol for the third-party PSI cardinality problem, which further improves the computational complexity for Q compared to both our third-party PSI protocols.

1.4. **Organization.** In Section 2, we describe formal definitions of third-party PSI related functionalities and the complexities of standard polynomial operations. In Section 3, we present a first protocol incorporating multiple improvements, resulting in a round-optimal protocol with significantly improved computational complexity.

[1] While the authors of [47] state a computational complexity of $O(n^4)$ for their protocol, the complexity of their protocol can in fact be improved to $O(n^{2.5+o(1)})$ by replacing the Cantor–Zassenhaus algorithm [5] (which is used in one of the steps of their protocol) with an algorithm by Kedlaya and Umans [24].

A second protocol which further reduces the computational complexity to near-linear is presented in Section 4. Section 5 describes a technical overview of the third-party PSI cardinality protocol with details. We provide a conclusion of this work in Section 6.

2. Preliminaries

2.1. **Definitions.** We recall the definition of a third-party PSI protocol from [47].

Definition 1 (Third-party PSI protocol). *In a third-party PSI protocol,* 2 *parties* P_1 *and* P_2 *each holds a dataset with elements in* $\{0,1\}^*$*, while a third-party* Q *has no input. At the end of the protocol,* Q *outputs the set intersection functionality, and the other parties output* $\perp$*.*

Ideal-world/real-world simulation-based definitions can be used to define the security of such a protocol. The protocol is secure if it achieves the ideal functionality shown in Figure 1.

(1) Get P_1's input set S_1.
(2) Get P_2's input set S_2.
(3) Send $S_1 \cap S_2$ to Q.

Figure 1. Third-party PSI ideal functionality

We define a third-party PSI cardinality protocol in a similar manner (see Definition 2). Such a protocol is secure if it achieves the ideal functionality in Figure 2.

Definition 2 (Third-party PSI cardinality protocol). *In a third-party PSI cardinality protocol,* 2 *parties* P_1 *and* P_2 *each holds a dataset with elements in* $\{0,1\}^*$*, while a third-party* Q *has no input. At the end of the protocol,* Q *outputs the cardinality of the set intersection, and the other parties output* $\perp$*.*

(1) Get P_1's input set S_1.
(2) Get P_2's input set S_2.
(3) Send $|S_1 \cap S_2|$ to Q.

Figure 2. Third-party PSI cardinality ideal functionality

2.2. **Complexity of Standard Polynomial Operations.** Let $\mathbb{F}$ be a field and $\mathbb{F}[X]$ be the ring of polynomials over $\mathbb{F}$. We write $\mathbb{F}[X]_{\leq d}$ for the subset of $\mathbb{F}[X]$ containing polynomials of degree $\leq d$.

Let $M(d) = O(d \log d \log\log d)$ be the complexity of multiplying two polynomials of degree $\leq d$. Table 1 lists the complexity of various common operations on polynomials over $\mathbb{F}$ (see, for example, Table 1 in [24]).

3. A Round-Optimal Third-Party PSI Protocol

3.1. **An Overview.** We will build upon Protocol 2 of [47] to achieve our round-optimal third-party PSI protocol. First, we improve the round complexity of the protocol by using a single-round key agreement protocol, which will allow us to combine multiple steps of the protocol and perform them in the same communication round, improving the efficiency of the protocol.

	input	output	number of $\mathbb{F}$ operations
multiplication	$f(X), g(X) \in \mathbb{F}[X]_{\leq d}$	$f(X) \cdot g(X)$	$M(d)$
remainder	$f(X), g(X) \in \mathbb{F}[X]_{\leq d}$	$f(X) \bmod g(X)$	$O(M(d))$
GCD	$f(X), g(X) \in \mathbb{F}[X]_{\leq d}$	$\gcd(f(X), g(X))$	$O(M(d) \log d)$
interpolation	$\alpha_0, \ldots, \alpha_d, \beta_0, \ldots, \beta_d$	$f(X)$ s.t. $f(\alpha_i) = \beta_i$	$O(M(d) \log d)$

Table 1. Complexity of standard polynomial operations

Second, we further improve upon the computational complexity of the protocol and reduce the computational cost by a factor of approximately n. Let us recall from Section 1.2 that in Protocol 2 of [47], Q recieves a set of keys K from P_1 and a polynomial q from P_2. For each key $k \in K$, Q then finds the roots of the polynomial $q(t) = k$ to obtain the element corresponding to the key k (if the element lies in the intersection). This means that Q needs to find the roots of n different polynomials.

This gives us an opportunity to reduce the computational cost by modifying the last few steps of the protocol from [47]. Instead of having P_1 directly sending the set K of keys he computed to Q as in the existing protocol, we use the set of keys computed by P_1 to interpolate a polynomial r, which is then sent to Q.

In our modified protocol, Q also receives a polynomial q from P_2 (as in [47]) which encodes the keys that result from running the key agreement protocol for each element of P_2's dataset. The desired intersection can then be obtained by Q by finding the roots of $q - r$. This greatly improves upon the existing protocol as the computations by Q are often the bottleneck in the entire protocol.

Unfortunately, this makes the protocol fail to correctly compute the set intersection when the P_1 and P_2 have identical sets. We make one additional modification to the protocol to deal with this specific case to avoid failure of the protocol.

3.2. **Details of the Protocol.** Let the size of each of P_1 and P_2's datasets be n, and let $S_1 = \{s_1, \ldots, s_n\} \subseteq \{0,1\}^\ell$ and $S_2 = \{t_1, \ldots, t_n\} \subseteq \{0,1\}^\ell$.

Fix some $\lambda > 0$, which is both the correctness and the security parameter, and fix some $\delta > 0$. Let $\lambda' = \max(\lambda, n^\delta)$. We shall identify $\{0,1\}^\ell$ with a subset S of a finite field $\mathbb{F}$ satisfying $|\mathbb{F}| \geq 2^{\ell + \lambda' + 2\log n}$. Choose

- a single-round key agreement protocol KA (see Figure 3) with space of randomness $\mathsf{KA}.\mathcal{R}$, message space $\mathsf{KA}.\mathcal{M} = \mathbb{F}$ and key space $\mathsf{KA}.\mathcal{K} = \mathbb{F}$, and
- an ideal permutation $\Pi : \mathbb{F} \to \mathbb{F}$.

(1) P_1 picks $a \leftarrow \mathsf{KA}.\mathcal{R}$ and sends $m_1 = \mathsf{KA}.\mathsf{msg}_1(a)$ to P_2, while P_2 picks $b \leftarrow \mathsf{KA}.\mathcal{R}$ and sends $m_2 = \mathsf{KA}.\mathsf{msg}_2(b)$ to P_1.
(2) P_1 and P_2 output $\mathsf{KA}.\mathsf{key}_1(a, m_2)$ and $\mathsf{KA}.\mathsf{key}_2(b, m_1)$ respectively.

Figure 3. A single-round key agreement protocol between P_1 and P_2

For two probability distributions X and Y (each indexed by a security parameter), we write $X \approx Y$ if X and Y are computationally indistinguishable. The key agreement protocol KA should satisfy the following three properties:

Property 1. *A single-round key agreement protocol* KA *is correct if, for all* $a, b \in \mathsf{KA}.\mathcal{R}$,

$$\mathsf{KA}.\mathsf{key}_1(a, \mathsf{KA}.\mathsf{msg}_2(b)) = \mathsf{KA}.\mathsf{key}_2(b, \mathsf{KA}.\mathsf{msg}_1(a)).$$

Property 2. *A single-round key agreement protocol* KA *has pseudorandom second messages if*

$$\{\mathsf{KA.msg}_2(b)\}_{b\leftarrow \mathsf{KA}.\mathcal{R}} \approx \{m_2\}_{m_2\leftarrow \mathsf{KA}.\mathcal{M}}.$$

Property 3. *A single-round key agreement protocol* KA *has pseudorandom keys if, for all* $a \in \mathsf{KA}.\mathcal{R}$,

$$\{\mathsf{KA.key}_2(b, \mathsf{KA.msg}_1(a))\}_{b\leftarrow \mathsf{KA}.\mathcal{R}} \approx \{k\}_{k\leftarrow \mathsf{KA}.\mathcal{K}}.$$

Fix some $u \in \mathbb{F}\backslash S$. If h is a positive integer, we denote by $[h]$ the set $\{1, 2, \ldots, h\}$. Recall that $S_1 = \{s_1, \ldots, s_n\}$ and $S_2 = \{t_1, \ldots, t_n\}$. Our 2-round third-party PSI protocol works as follows:

(1) P_1 picks a random $a \leftarrow \mathsf{KA}.\mathcal{R}$.
(2) P_1 sends $m = \mathsf{KA.msg}_1(a)$ to P_2.
(3) For each $i \in [n]$, P_2 picks a random $b_i \leftarrow \mathsf{KA}.\mathcal{R}$ and computes $m'_i = \mathsf{KA.msg}_2(b_i)$ and $f_i = \Pi^{-1}(m'_i)$.
(4) P_2 computes the unique polynomial p of degree $\le n-1$ such that $p(t_i) = f_i$ for all $i \in [n]$, and sends p to P_1.
(5) For each $i \in [n]$, P_1 computes $k_i = \mathsf{KA.key}_1(a, \Pi(p(s_i)))$.
(6) P_1 picks a random $k \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial r of degree $\le n$ such that $r(u) = k$ and $r(s_i) = k_i$ for all $i \in [n]$, and sends r to Q.
(7) P_2 picks a random $k' \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial q of degree $\le n$ such that $q(u) = k'$ and $q(t_i) = \mathsf{KA.key}_2(b_i, m)$ for all $i \in [n]$, and sends q to Q.
(8) Q computes all solutions t to the equation $q(t) - r(t) = 0$ with $t \in S$, and outputs $\{t \in S : q(t) - r(t) = 0\}$.

PROTOCOL 1. A 2-round third-party PSI protocol

Note that steps 2 and 4 of Protocol 1 can be performed in the same communication round, as can steps 6 and 7, giving Protocol 1 a round complexity of 2 rounds.

In steps 6 and 7 of the Protocol 1, we require P_1 and P_2 to each choose a random element in $\mathsf{KA}.\mathcal{K}$, and interpolate a polynomial such that the polynomial has this chosen value at some fixed point u. Essentially, instead of choosing the unique polynomials of degree $\le n-1$ satisfying their respective constraints, P_1 and P_2 are choosing random polynomials of degree $\le n$ that satisfy the constraints. This is needed to deal with the edge case where $S_1 = S_2$; otherwise, in this particular case, the polynomials q and r will be identical, and hence Q will be unable to determine the intersection.

Compared to Protocol 2 in [47], the computations needed by Q to determine the intersection has been reduced from solving n equations $q(t) = k_i$, for $i \in [n]$, to solving a single equation $q(t) - r(t) = 0$. Using the fast polynomial factorization algorithm by Kedlaya and Umans [24], the computational complexity of the protocol is $O(n^{1.5+o(1)} \log^{1+o(1)} |\mathbb{F}| + n^{1+o(1)} \log^{2+o(1)} |\mathbb{F}|)$ bit operations. The communication cost is $3(n+1)\log|\mathbb{F}|$ bits.

3.3. **Correctness.** We now prove that Protocol 1 correctly computes the set intersection functionality except with negligible probability.

Proposition 1. *Assume that* KA *satisfies Properties 1, 2 and 3 with security parameter λ', and that Π is an ideal permutation. Then Protocol 1 is correct except* *with probability* $\leq 2^{-\lambda'+1} + n^2\eta(\lambda')$*, where* $\eta(\lambda')$ *is a negligible function of* λ'*. In particular, Protocol* 1 *is correct except with probability negligible in* λ*.*

Proof. Protocol 1 outputs $S_1 \cap S_2$ unless

(i) for some $i \in [n]$ and $t_j \in S_2$ such that $t_j \neq s_i$, we have $k_i = \mathsf{KA.key}_2(b_j, m)$ where $b_j \in \mathsf{KA}.\mathcal{R}$ is the randomness corresponding to t_j, or
(ii) $q(t) - r(t) = 0$ has a solution $t \in S$ with $t \notin S_1 \cap S_2$.

By Property 3 of KA, for fixed $i, j \in [n]$ such that $t_j \neq s_i$, the probability that $k_i = \mathsf{KA.key}_2(t_j, m)$ is negligibly close to $1/|\mathsf{KA}.\mathcal{K}|$. Taking the union bound over $i, j \in [n]$, we see that the probability that (i) holds is $\leq n^2/|\mathsf{KA}.\mathcal{K}| + n^2\eta(\lambda') = 2^{-\ell-\lambda'} + n^2\eta(\lambda')$, where $\eta(\lambda')$ is a negligible function of λ'.

Now suppose that (i) does not occur. Note that Properties 1, 2 and 3 together imply that $\{\mathsf{KA.key}_1(a, m)\}_{m\leftarrow\mathsf{KA}.\mathcal{M}} \approx \{k\}_{k\leftarrow\mathsf{KA}.\mathcal{K}}$ for all $a \in \mathsf{KA}.\mathcal{R}$.

Since outputs of $\mathsf{KA.key}_1$ and $\mathsf{KA.key}_2$ are both indistinguishable from uniformly random, the pair (q, r) is indistinguishable from a pair of random polynomials of degree $\leq n$ in $\mathbb{F}[X]$ such that $q(t) = r(t)$ for $t \in S_1 \cap S_2$. As we are assuming that (i) does not occur, the polynomials q and r must be distinct if $S_1 \neq S_2$. In the case where $S_1 = S_2$, the probability that q and r are identical is equal to $1/|\mathbb{F}| = 2^{-\ell-\lambda'-2\log n}$.

Now, assume the polynomials q and r are distinct, so that $q - r$ is not the zero polynomial. Then, the roots of $q - r$ in $\mathbb{F}$ are $(S_1 \cap S_2) \cup \{\gamma_1, \dots, \gamma_{n'}\}$, with $n' \leq n - |S_1 \cap S_2|$, and $\gamma_1, \dots, \gamma_{n'}$ being indistinguishable from uniformly random elements of $\mathbb{F}$. By the union bound, the probability that some γ_j lies in $S \subset \mathbb{F}$ is $\leq n|S|/|\mathbb{F}| = n2^{-\lambda'-2\log n}$.

Thus, Protocol 1 gives the correct output except with probability $\leq 2^{-\ell-\lambda'} + n^2\eta(\lambda') + 2^{-\ell-\lambda'-2\log n} + n2^{-\lambda'-2\log n} \leq 2^{-\lambda'+1} + n^2\eta(\lambda')$. Since $n^2 \leq (\lambda')^{\frac{2}{3}}$ is bounded above by a polynomial in λ', Protocol 1 is correct except with probability negligible in λ'. As $\lambda' \geq \lambda$, this probability is also negligible in λ. □

3.4. **Security.**

Proposition 2. *Assume* KA *satisfies Property 2 with security parameter* λ'*, and* Π *is an ideal permutation. Then Protocol* 1 *is secure against a semi-honest* P_1*.*

Proof. As KA satisfies Property 2, changing $m'_i = \mathsf{KA.msg}_2(b_i)$ to $m'_i \leftarrow \mathsf{KA}.\mathcal{M}$ (for some i) cannot be distinguished by P_1 except with probability negligible in λ'. As n is bounded above by a polynomial in λ', performing this change for all i is still indistinguishable to P_1 except with probability negligible in λ'. Thus, the polynomial p can be simulated by a uniformly random polynomial of degree $\leq n - 1$. □

Proposition 3. *Protocol* 1 *is secure against a semi-honest* P_2*.*

Proof. This is clear as P_2 only receives the message m from P_1, which does not depend on the input S_1. □

Proposition 4. *Assume* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* Π *is an ideal permutation. Then Protocol* 1 *is secure against a semi-honest* Q*.*

Proof. Hybrid 0: The real interaction.

Hybrid 1: We abort if there exists $s^* \in S_1 \setminus S_2$ and $t^* \in S_2$ such that $p(s^*) = p(t^*)$. Since p is indistinguishable from a uniformly chosen polynomial of degree

$\leq n-1$, by the union bound, the probability of abort is $\leq n^2/|\mathbb{F}| = 2^{-\ell-\lambda'} < 2^{-\lambda'}$, which is negligible.

Hybrid 2: We shall change how the ideal permutation Π is simulated. Since we have not aborted, we know there has been no query to Π at $p(s_i)$ in steps 1 to 4 for each $s_i \in S_1 \setminus S_2$. In this hybrid, we choose $r_i \leftarrow \mathsf{KA}.\mathcal{R}$, and set $\Pi(p(s_i)) = \mathsf{KA}.\mathsf{msg}_2(r_i)$. Since $\mathsf{KA}.\mathsf{msg}_2(r_i)$ is indistinguishable from uniformly random by Property 2 of KA, and $|S_1 \setminus S_2| \leq n$ is bounded by a polynomial in λ', this hybrid is indistinguishable from Hybrid 1.

Hybrid 3: We shall change how the k_i values are computed. If $s_i = t_j$ for some $t_j \in S_2$, we set $k_i = \mathsf{KA}.\mathsf{key}_2(b_j, m)$, else we set $k_i = \mathsf{KA}.\mathsf{key}_2(r_i, m)$.

Hybrid $(4, h)$ *for* $h \in [n+1]$: We again change how the k_i values are computed. We set:

$$k_i = \begin{cases} \mathsf{KA}.\mathsf{key}_2(b_j, m) & \text{if } s_i = t_j \text{ for some } t_j \in S_2, \\ k'_i \text{ where } k'_i \leftarrow \mathsf{KA}.\mathcal{K} & \text{if } s_i \neq t_j \text{ for all } t_j \in S_2 \text{ and } i < h, \\ \mathsf{KA}.\mathsf{key}_2(r_i, m) & \text{otherwise.} \end{cases}$$

Hybrid $(4, 1)$ is identical to Hybrid 3. By Property 3 of KA, Hybrid $(4, h)$ is indistinguishable from Hybrid $(4, h+1)$ for each $h \in [n]$.

Hybrid $(5, h)$ *for* $h \in [n+1]$: We let q be the unique polynomial of degree $\leq n$ such that $q(u) = k'$ where $k' \leftarrow \mathsf{KA}.\mathcal{K}$ and

$$q(t_j) = \begin{cases} \mathsf{KA}.\mathsf{key}_2(b_j, m) & \text{if } t_j = s_i \text{ for some } i, \text{ or } j \geq h, \\ k''_j \text{ where } k''_j \leftarrow \mathsf{KA}.\mathcal{K} & \text{otherwise.} \end{cases}$$

Hybrid $(5, 1)$ is identical to Hybrid $(4, n+1)$, and Hybrid $(5, h)$ is indistinguishable from Hybrid $(5, h+1)$ for each $h \in [n]$, again, by Property 3 of KA.

Simulator: We simulate

$$q, r \leftarrow \{\rho \in \mathbb{F}[X] : \rho(t_j) = \mathsf{KA}.\mathsf{key}_2(b_j, m) \text{ for } t_j \in S_1 \cap S_2,\ \deg(\rho) \leq n\}.$$

□

4. A Round-Optimal Near-Linear Third-Party PSI Protocol

4.1. **An Overview.** In this section, we shall further improve the computational complexity of our protocol to achieve a round-optimal third-party PSI protocol with near-linear computation and communication.

This is achieved by using a hash function to hash the inputs of the parties into some number of bins before applying Protocol 1. We make a careful analysis and choice of parameters to ensure the correctness and security of this improved protocol.

Crucially, a "large" number of bins is essential for us to obtain a low computational cost, as only then will each bin have a "small" number of elements, thus reducing the size of the instances on which we apply Protocol 1 to. However, Protocol 1 has a small negligible probability of producing an incorrect output. As we are now running the protocol many times, once on each bin, the number of bins cannot be too large so that the probability of obtaining even an incorrect output is still negligible. By carefully balancing these two requirements, we obtain a suitable choice for the number b of bins.

Now, since the hash function behaves essentially like a random function, each of the b bins will on average have n/b elements, where n is the size of each party's dataset. However, due to the randomness inherent in the process, bins are unlikely to contain exactly n/b elements. To preserve the privacy of the datasets, the exact number of elements in each bin cannot be leaked. Thus, it is necessary to pad each bin with dummy elements up to some maximum size M. Choosing too small a value for M will result in a non-negligible probability of some bin overflowing and the protocol aborting. while a value of M that is too large will affect computation and communication costs. Again, we have to strike a balance between these two contrasting requirements to obtain a suitable choice for the maximum bin size M.

Finally, we need to ensure that there are enough dummy elements that can be used to pad the bins up to the maximum size M. We achieve this by embedding the set of all possible elements into a larger set. This set is chosen just large enough so that there are enough dummy elements with high probability, while, at the same time, not causing a significant increase in the computation and communication costs.

The idea of applying hashing techniques to PSI has been explored before by various works such as [15, 39, 37], where the number of bins used is $\tilde{\Theta}(n)$ (i.e. linear in n up to logarithmic terms). In this paper, however, we use a choice of $b = \lceil n^\alpha \rceil$ bins, where α is some constant satisfying $0 < \alpha < 1$. This complicates the analysis, but as we will see, choosing a value of $\alpha < 1$ allows us to achieve a lower communication complexity compared to $\alpha = 1$, and thus results in a more communication efficient protocol.

4.2. **Details of the Protocol.** We will modify the setup used in Section 3. We start by identifying $\{0,1\}^\ell$ as a subset of $\{0,1\}^\kappa$ for some $\kappa > \ell$. (Most commonly, we will let $\kappa = \ell + 1$.)

Fix some positive integer b, and let $H : \{0,1\}^\kappa \to [b]$ be an ideal hash function. We introduce a parameter $0 < \mu < 1$ such that the probability that any bin contains more than $(1+\mu)n/b$ elements or less than $(1-\mu)n/b$ elements of S_1 or S_2 is negligible in λ, where, as above, $\lambda > 0$ is both the correctness and security parameter. The precise value of μ will be chosen later.

Assume that, for each $j \in [b]$, there are at least $\lceil 4\mu n/b \rceil + 4$ elements of $\{0,1\}^\kappa \setminus \{0,1\}^\ell$ that hashes to the j-th bin, and let us fix any two disjoint subsets $R_{1,j}, R_{2,j} \subseteq \{0,1\}^\kappa \setminus \{0,1\}^\ell$, each of size $\lceil 2\mu n/b \rceil + 2$, such that elements in $R_{1,j}$ and $R_{2,j}$ both mapped to the j-th bin under the hash function H.

Fix some $\delta > 0$ and let $\lambda' = \max(\lambda, n^\delta)$. We shall identify $\{0,1\}^\kappa$ with a subset S of a finite field $\mathbb{F}$ with $|\mathbb{F}| \geq 2^{\kappa+\lambda'+2\log n}$. We choose

- a single-round key agreement protocol KA with space of randomness $\mathsf{KA}.\mathcal{R}$, message space $\mathsf{KA}.\mathcal{M} = \mathbb{F}$ and key space $\mathsf{KA}.\mathcal{K} = \mathbb{F}$, and
- ideal permutations $\Pi_1, \ldots, \Pi_b : \mathbb{F} \to \mathbb{F}$.

We now present our round-optimal third-party PSI protocol which has near-linear computation and communication costs:

(1) P_1 and P_2 use H to hash their elements into b bins. Let $s_{i,j} = |\{s \in S_i : H(s) = j\}|$ be the size of the j-th bin for P_i. Abort if $s_{i,j} > (1+\mu)n/b$ or $s_{i,j} < (1-\mu)n/b$ for some i, j.

(2) For each $j \in [b]$:

(a) Let $M = \lceil (1+\mu)n/b \rceil$. P_i chooses a subset $R'_{i,j} \subseteq R_{i,j}$ of size $M - s_{i,j}$, and defines
$$S_{i,j} = \{s \in S_i : H(s) = j\} \cup R'_{i,j}.$$
Write $S_{1,j} = \{s_{j,1}, \ldots, s_{j,M}\}$ and $S_{2,j} = \{t_{j,1}, \ldots, t_{j,M}\}$.

(b) P_1 picks a random $a_j \leftarrow \mathsf{KA}.\mathcal{R}$.

(c) P_1 sends $m_j = \mathsf{KA.msg}_1(a_j)$ to P_2.
(d) For each $i \in [M]$, P_2 picks a random $b_{j,i} \leftarrow \mathsf{KA}.\mathcal{R}$ and let $m'_{j,i} = \mathsf{KA.msg}_2(b_{j,i})$ and $f_{j,i} = \Pi_j^{-1}(m'_{j,i})$.
(e) P_2 computes the unique polynomial p_j of degree $\leq M - 1$ such that $p_j(t_{j,i}) = f_{j,i}$ for all $i \in [M]$, and sends p_j to P_1.
(f) For each $i \in [M]$, P_1 computes
$$k_{j,i} = \mathsf{KA.key}_1(a_j, \Pi_j(p_j(s_{j,i}))).$$
(g) P_1 picks a random $k'_j \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial r_j of degree $\leq M$ such that $r_j(u) = k'_j$ and $r_j(s_{j,i}) = k_{j,i}$ for all $i \in [M]$, and sends r_j to Q.
(h) P_2 picks a random $k''_j \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial q_j of degree $\leq M$ such that $q_j(u) = k''_j$ and $q_j(t_{j,i}) = \mathsf{KA.key}_2(b_{j,i}, m_j)$ for all $i \in [M]$, and sends q_j to Q.
(i) Q computes all solutions t to the equation $q_j(t) - r_j(t) = 0$ with $t \in S$, and sets $I_j = \{t \in S : q_j(t) - r_j(t) = 0\}$.

(3) Q outputs $\bigcup_{j=1}^{b} I_j$.

PROTOCOL 2. A 2-round near-linear third-party PSI protocol

Essentially, steps 2(b) to 2(i) correspond to running Protocol 1 a total of b times, once on each bin.

4.3. **Parameter Choices.**

4.3.1. *Choice of μ.* In Protocol 2, the parties abort if any bin contains more then $(1+\mu)n/b$ elements or less then $(1-\mu)n/b$ elements of S_1 or S_2, hence μ must be chosen so that the probability of abort is negligible. We will now obtain an upper bound on this probability using the Chernoff bound [11]:

Proposition 5 (Chernoff bound)**.** *Let X be a binomial random variable with N trials and success probability p. If $0 < \mu < 1$, then*
$$\Pr[X < (1-\mu)pN] \leq \exp\left(-\frac{\mu^2 pN}{2}\right),$$
and
$$\Pr[X > (1+\mu)pN] \leq \exp\left(-\frac{\mu^2 pN}{3}\right).$$

Proposition 6. *Let $X_{i,j}$ be the number of elements of S_i in the j-th bin.*
$$\Pr\left[X_{i,j} > \frac{(1+\mu)n}{b} \textit{ or } X_{i,j} < \frac{(1-\mu)n}{b} \textit{ for some } i,\, j\right]$$
is negligible in λ for $\mu = \sqrt{\frac{3b}{n}(\lambda + \ln 2b)}$.

Proof. Each $X_{i,j}$ is a binomial random variable with $N = n$ and $p = 1/b$. By Proposition 5,
$$\Pr\left[X_{i,j} > \frac{(1+\mu)n}{b} \text{ or } X_{i,j} < \frac{(1-\mu)n}{b}\right] \leq \exp\left(-\frac{\mu^2 n}{2b}\right) + \exp\left(-\frac{\mu^2 n}{3b}\right).$$
Now, applying the union bound over i, j, we have
$$\Pr\left[X_{i,j} > \frac{(1+\mu)n}{b} \text{ or } X_{i,j} < \frac{(1-\mu)n}{b} \text{ for some } i,\, j\right]$$

$$< 2b\exp\left(-\frac{\mu^2 n}{3b}\right) = \exp(-\lambda).$$

□

4.3.2. *Choice of b.* We fix some $0 < \alpha < 1$ and let $b = \lceil n^\alpha \rceil$. As we shall see later, such a choice of b allows us achieve a low computational cost.

4.3.3. *Choice of κ.* We need to choose κ such that, there are at least $\lceil 4\mu n/b \rceil + 4$ elements of $\{0,1\}^\kappa \setminus \{0,1\}^\ell$ that hashes to the j-th bin for each $j \in [b]$. Since

$$\lceil 4\mu n/b \rceil + 4 \leq \left\lceil 4\sqrt{3n^{1-\alpha}(\lambda + \ln(2n^\alpha + 2))} \right\rceil + 4 = \Theta\left(n^{\frac{1-\alpha}{2}}\sqrt{\log n}\right),$$

$$\frac{n}{b} = \frac{n}{\lceil n^\alpha \rceil} = \Theta(n^{1-\alpha}),$$

it follows that, for sufficiently large n, we have $\lceil 4\mu n/b \rceil + 4 < n/2b$.

Hence, by Proposition 5, given any set of at least n elements (with n sufficiently large), the probability that there are $< \lceil 4\mu n/b \rceil + 4$ elements that hashes to the j-th bin (for any fixed j) is bounded above by $\exp(-n/8b)$. By the union bound, the probability that the above happens for some $j \in [b]$ is bounded above by $b\exp(-n/8b) < 2n^\alpha \exp\big(-n^{1-\alpha}/16\big) = o(1)$. Since $|\{0,1\}^{\ell+1} \setminus \{0,1\}^\ell| = 2^\ell \geq n$, this shows that $\kappa = \ell + 1$ will work with high probability.

4.4. **Communication and Computational Costs.** From the protocol description, we note that Protocol 2 requires $3b(M+1)(\kappa + \lambda' + 2\log n)$ bits of communication. With the above choice of parameters and assuming that $\kappa = \ell + 1$, this is bounded above by

$$3\left(n + \sqrt{3(n^{1+\alpha} + n)(\ln(2n^\alpha + 2) + \lambda)} + 2n^\alpha + 2\right)(n^\delta + 2\log n + \lambda + \ell + 1),$$

which is $O(n^{1+\delta})$.[2]

The computational cost of Protocol 2 is dominated by step 2(i), which, using the algorithm of Kedlaya and Umans [24], has a complexity of $O(M^{1.5+o(1)}\log^{1+o(1)}|\mathbb{F}| + M^{1+o(1)}\log^{2+o(1)}|\mathbb{F}|)$ bit operations. Since step 2(i) is performed b times, this gives us a total complexity of $O(bM^{1.5+o(1)}\log^{1+o(1)}|\mathbb{F}| + bM^{1+o(1)}\log^{2+o(1)}|\mathbb{F}|)$, and this becomes $O\big(n^{1.5-0.5\alpha+\delta+o(1)} + n^{1+2\delta+o(1)}\big)$ with our choice of parameters. By picking $0 < \alpha < 1$ and $\delta > 0$ appropriately, the computational complexity can be made $O(n^{1+\varepsilon})$ for any $\varepsilon > 0$.

4.5. **Correctness.** From this point on, we will assume that $(1+\mu)/b \leq 1$, i.e. the maximum size M of each bin satisfies $M = \lceil (1+\mu)n/b \rceil \leq n$. This assumption is equivalent to

$$1 + \sqrt{\frac{3\lceil n^\alpha \rceil}{n}(\lambda + \ln 2\lceil n^\alpha \rceil)} \leq \lceil n^\alpha \rceil,$$

which is clearly satisfied for sufficiently large n.

A straightforward modification of Proposition 1 yields the following:

Lemma 7. *Let $j \in [b]$. Assume that KA satisfies Properties 1, 2 and 3 with security parameter λ', and that Π_j is an ideal permutation. Then*

$$I_j = S_{1,j} \cap S_{2,j} = \{s \in S_1 \cap S_2 : H(s) = j\}$$

except with probability $\leq \frac{M^2}{n^2}2^{-\lambda'+1} + M^2\eta(\lambda')$, where $\eta(\lambda')$ is negligible in λ'.

[2]For fixed $\delta > 0$, choosing $\alpha = 1$ instead of $\alpha < 1$ gives us a comparatively worse communication complexity of $O(n^{1+\delta}\sqrt{\log(n)})$.

Proposition 8. *Assume that* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* $\Pi_1, \ldots, \Pi_b$ *are ideal permutations. Then Protocol* 2 *is correct except with probability negligible in* λ.

Proof. By the choice of μ, the probability of abort in step 1 of the protocol is negligible in λ. If

$$I_j = \{s \in S_1 \cap S_2 : H(s) = j\} \tag{1}$$

for all $j \in [b]$, then Q outputs $\bigcup_{j=1}^{b} I_j = \{s \in S_1 \cap S_2 : H(s) \in [b]\} = S_1 \cap S_2$. By Lemma 7, for each $j \in [b]$, condition (1) holds except with probability $\leq \frac{M^2}{n^2} 2^{-\lambda'+1} + M^2 \eta(\lambda')$, where $\eta(\lambda')$ is a negligible function of λ'. Thus, by the union bound, condition (1) holds for all $j \in [b]$ except with probability

$$\leq \frac{bM^2}{n^2} 2^{-\lambda'+1} + bM^2 \eta(\lambda') \leq b(2^{-\lambda'+1}) + bn^2 \eta(\lambda').$$

Since $\lambda' \geq n^{\delta}$, both $b = \lceil n^{\alpha} \rceil$ and $bn^2 = n^2 \lceil n^{\alpha} \rceil$ are bounded above by some polynomial in λ', hence $b(2^{-\lambda'+1}) + bn^2 \eta(\lambda')$ is a negligible function of λ'. As $\lambda' \geq \lambda$, it too is a negligible function of λ. $\square$

4.6. **Security.** The following propositions prove that Protocol 2 is secure against a semi-honest adversary corrupting a single party.

Proposition 9. *Assume* KA *satisfies Property 2 with security parameter* λ'*, and* $\Pi_1, \ldots, \Pi_b$ *are ideal permutations. Then Protocol* 2 *is secure against a semi-honest* P_1.

Proof. We argue as in the proof of Proposition 2, noting that $Mb \leq n \lceil n^{\alpha} \rceil$ is bounded above by a polynomial in λ'. Thus, for each $j \in [b]$, the polynomial p_j can be simulated by a uniformly random polynomial of degree $\leq M - 1$. $\square$

Proposition 10. *Protocol* 2 *is secure against a semi-honest* P_2.

Proof. This is clear as P_2 only receives the messages $m_1, \ldots, m_b$ from P_1. $\square$

The next lemma follows immediately from the proof of Proposition 4:

Lemma 11. *Assume* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* Π_j *is an ideal permutation. Then simulating* q_j *and* r_j *by*

$$q_j, r_j \leftarrow \{\rho \in \mathbb{F}[X]_{\leq M} : \rho(t_{j,i}) = \mathsf{KA.key}_2(b_{j,i}, m_j) \textit{ for } t_{j,i} \in S_{1,j} \cap S_{2,j}\}$$

is indistinguishable to Q *except with probability negligible in* λ'.

Proposition 12. *Assume* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* $\Pi_1, \ldots, \Pi_b$ *are ideal permutations. Then Protocol* 2 *is secure against a semi-honest* Q.

Proof. By Lemma 11, for each $j \in [b]$, we can simulate

$$q_j, r_j \leftarrow \{\rho \in \mathbb{F}[X]_{\leq M} : \rho(t_{j,i}) = \mathsf{KA.key}_2(b_{j,i}, m_j) \text{ for } t_{j,i} \in S_{1,j} \cap S_{2,j}\}.$$

By the union bound, this change is indistinguishable to Q except with probability at most $b\zeta(\lambda')$, where $\zeta(\lambda')$ is a negligible function of λ'. Again, since $b = \lceil n^{\alpha} \rceil$ is bounded above by a polynomial in λ', the probability $b\zeta(\lambda')$ is negligible in λ', hence negligible in λ. $\square$

5. A Third-Party PSI Cardinality Protocol

5.1. **An Overview.** To obtain a third-party PSI cardinality protocol, we make a small modification to Protocol 1 and have P_1 and P_2 first apply a pseudorandom permutation (PRP) to their elements using a common key, so that the actual intersection elements are hidden from Q. This small change already gives us a secure third-party PSI cardinality protocol. However, we can further improve its computational costs to obtain a more efficient protocol.

Recall that the most computational expensive step in Protocol 1 is the final step, which involves Q solving a polynomial to detemine the intersection elements. As we do not now require the actual intersection elements, the computational complexity of this step can be improved by replacing it with a more efficient algorithm that determines only the number of roots, but not the set of roots, of the polynomial.

To make this more efficient algorithm work, we make a slight modification to the setup used in Protocol 1 so that the set $\{0,1\}^\ell$ is now identified with a subfield $\mathbb{S}$ (instead of an arbitrary subset) of $\mathbb{F}$.

5.2. **Details of the Protocol.** We use the same setup as in Section 3.2, except that $\{0,1\}^\ell$ is now identified with the unique subfield $\mathbb{S}$ of cardinality 2^ℓ of a finite field $\mathbb{F}$ (which satisfies $|\mathbb{F}| \geq 2^{\ell+\lambda'+2\log n}$). Furthermore, let $E : \mathcal{K} \times \mathbb{S} \to \mathbb{S}$ be a PRP with key space $\mathcal{K}$, and fix some $u \in \mathbb{F} \setminus \mathbb{S}$.

(1) P_1 and P_2 agree on a random key $k \leftarrow \mathcal{K}$.
(2) P_1 picks a random $a \leftarrow \mathsf{KA}.\mathcal{R}$.
(3) P_1 sends $m = \mathsf{KA.msg}_1(a)$ to P_2.
(4) For each $i \in [n]$, P_2 picks a random $b_i \leftarrow \mathsf{KA}.\mathcal{R}$ and computes $m'_i = \mathsf{KA.msg}_2(b_i)$ and $f_i = \Pi^{-1}(m'_i)$.
(5) P_2 computes the unique polynomial p of degree $\leq n-1$ such that $p(E_k(t_i)) = f_i$ for all $i \in [n]$, and sends p to P_1.
(6) For each $i \in [n]$, P_1 computes
$$k_i = \mathsf{KA.key}_1(a, \Pi(p(E_k(s_i)))).$$
(7) P_1 picks a random $k' \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial r of degree $\leq n$ such that $r(u) = k'$ and $r(E_k(s_i)) = k_i$ for all $i \in [n]$, and sends r to Q.
(8) P_2 picks a random $k'' \leftarrow \mathsf{KA}.\mathcal{K}$, computes the unique polynomial q of degree $\leq n$ such that $q(u) = k''$ and $q(E_k(t_i)) = \mathsf{KA.key}_2(b_i, m)$ for all $i \in [n]$, and sends q to Q.
(9) Let $f(X) = q(X) - r(X)$. Q computes $g(X) = X^{2^\ell} \bmod f(X)$ using repeated squaring and reduction modulo $f(X)$.
(10) Q computes $h(X) = \gcd(f(X), g(X) - X)$ and outputs $\deg h(X)$.

Protocol 3. A third-party PSI cardinality protocol

Note that step 9 takes $\ell(M(n) + O(M(2n))) = O(n \log n \log\log n)$ field operations (where $M(n)$ is the complexity of multiplying two polynomials of degree $\leq n$), while step 10 takes $O(M(n)\log n) = O(n \log^2 n \log\log n)$ field operations, giving a total computational complexity for Q that is quasilinear. The communication cost of Protocol 3 is $3(n+1)\log|\mathbb{F}| + \log|\mathcal{K}|$ bits.

5.3. **Correctness and Security.**

Proposition 13. *Assume that* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* Π *is an ideal permutation. Then Protocol* 3 *is correct except with probability negligible in* λ*.*

Proof. Following the proof of Proposition 1, the roots of the polynomial $f = q - r$ which lie in $\mathbb{S}$ are $E_k(s)$ for $s \in S_1 \cap S_2$ except with probability negligible in λ. As $\mathbb{S}$ is the unique subfield of $\mathbb{F}$ of cardinality 2^ℓ, the roots of the polynomial $X^{2^\ell} - X = \prod_{\alpha \in \mathbb{S}}(X - \alpha)$ are precisely the elements of $\mathbb{S}$.

Since $h(X) = \gcd(f(X), g(X) - X) = \gcd(f(X), X^{2^\ell} - X)$, it follows that the roots of h are precisely $E_k(s)$ for $s \in S_1 \cap S_2$ except with probability negligible in λ, so h has degree $|S_1 \cap S_2|$, as required. □

Proposition 14. *Assume* KA *satisfies Properties 1, 2 and 3 with security parameter* λ'*, and that* Π *is an ideal permutation. Then Protocol* 3 *is secure against a semi-honest adversary corrupting a single party.*

The proof of Proposition 14 essentially follows from the proofs of Propositions 2, 3 and 4, and is therefore omitted.

6. Conclusion

Third-party private set intersection was recently introduced in [47]. They presented two protocols, one of which is a Diffie-Hellman based approach and the other based on key agreement. While both their solutions achieve a low communication cost, the Diffie-Hellman based approach suffers from a high number of communication rounds, and the key agreement approach incurs a high computational overhead. In this paper, we overcome the limitations of existing work by developing an improved protocol which significantly lowers the computational cost of the key agreement based protocol and uses only 2 communication rounds, which is optimal for a third-party PSI protocol.

We do so by proposing multiple improvements to the key agreement based third-party PSI protocol of [47], and we present two protocols using these improvements. Both our protocols only require 2 rounds of commuication. The first protocol already gives a significant reduction in the computational cost from $O(n^{2.5+o(1)})$ to $O(n^{1.5+o(1)})$ and works even for small n, while the second improvement is an asymptotic improvement that is important for large n and further reduces the computational cost to $O(n^{1+\varepsilon})$ for any constant $\varepsilon > 0$.

Finally, we also introduce a protocol with an even lower computational complexity of $O(n \log^2 n \log\log n)$ for the third-party Q, in the situation where it is desired that only the size, but not the contents, of the intersection is revealed.

References

[1] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions. *Journal of Cryptology*, 30:805–858, 2017.

[2] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Countering GATTACA: efficient and secure testing of fully-sequenced human genomes. In *Proceedings of the 18th ACM conference on Computer and communications security (CCS'11)*, pages 691–702. ACM, 2011.

[3] Alexander Bienstock, Sarvar Patel, Joon Young Seo, and Kevin Yeo. Near-optimal oblivious key-value stores for efficient PSI, PSU and volume-hiding multi-maps. In *SEC '23: Proceedings of the 32nd USENIX Conference on Security Symposium*, pages 301–318. USENIX Association, 2023.

[4] Prasad Buddhavarapu, Andrew Knox, Payman Mohassel, Shubho Sengupta, Erik Taubeneck, and Vlad Vlaskin. Private matching for compute. *IACR Cryptology ePrint Archive, 2020:599*, 2020. `https://eprint.iacr.org/2020/599`.

[5] David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):587–592, 1981.

[6] Nishanth Chandran, Nishka Dasgupta, Divya Gupta, Sai Lakshmi Bhavana Obbattu, Sruthi Sekar, and Akash Shah. Efficient linear multiparty PSI and extensions to circuit/quorum PSI. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1182–1204. ACM, 2021.

[7] Melissa Chase and Peihan 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*, pages 34–63. Springer International Publishing, 2020.

[8] Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. Labeled PSI from fully homomorphic encryption with malicious security. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1223–1237. ACM, 2018.

[9] Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1243–1255. ACM, 2017.

[10] Kai Chen, Yongqiang Li, and Mingsheng Wang. An efficient toolkit for computing third-party private set intersection. In *Progress in Cryptology – INDOCRYPT 2024*, pages 258–281. Springer Nature Switzerland, 2025.

[11] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[12] Kelong Cong, Radames Cruz Moreno, Mariana Botelho da Gama, Wei Dai, Ilia Iliashenko, Kim Laine, and Michael Rosenberg. Labeled PSI from homomorphic encryption with reduced computation and communication. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1135–1150. ACM, 2021.

[13] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Fast and private computation of cardinality of set intersection and union. In *Cryptology and Network Security*, pages 218–231. Springer Berlin Heidelberg, 2012.

[14] Thai Duong, Duong Hieu Phan, and Ni Trieu. Catalic: Delegated PSI cardinality with applications to contact tracing. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2020)*, pages 870–899. Springer, 2020.

[15] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology - EUROCRYPT 2004*, pages 1–19. Springer, Berlin, Heidelberg, 2004.

[16] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Oblivious key-value stores and amplification for private set intersection. In *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021*, pages 395–425. Springer International Publishing, 2021.

[17] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022*, pages 323–352. Springer Nature Switzerland, 2022.

[18] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In *Advances in Cryptology – CRYPTO 2023: 43rd Annual International Cryptology Conference, CRYPTO 2023*, pages 577–610. Springer Nature Switzerland, 2023.

[19] Jingwei Hu, Junyan Chen, Wangchen Dai, and Huaxiong Wang. Fully homomorphic encryption-based protocols for enhanced private set intersection functionalities. *IACR Cryptology ePrint Archive, 2023:1407*, 2023. `https://eprint.iacr.org/2023/1407`.

[20] Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On deploying secure computing: Private intersection-sum-with-cardinality. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 370–389. IEEE, 2020.

[21] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Annual International Cryptology Conference*, pages 145–161. Springer Berlin Heidelberg, 2003.

[22] Yuval Ishai, Ranjit Kumaresan, Eyal Kushilevitz, and Anat Paskin-Cherniavsky. Secure computation with minimal interaction, revisited. In *Advances in Cryptology – CRYPTO 2015*, pages 359–378. Springer-Verlag, 2022.

[23] Daniel Kales, Christian Rechberger, Thomas Schneider, Matthias Senker, and Christian Weinert. Mobile private contact discovery at scale. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1447–1464, 2019.

[24] Kiran S. Kedlaya and Christopher Umans. Fast polynomial factorization and modular composition. *SIAM Journal on Computing*, 40(6):1767–1802, 2011.

[25] Florian Kerschbaum. Outsourced private set intersection using homomorphic encryption. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 85–86. ACM, 2012.

[26] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set intersection. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, pages 818–829. ACM, 2016.

[27] Vladimir Kolesnikov, Naor Matania, Benny Pinkas, Mike Rosulek, and Ni Trieu. Practical multi-party private set intersection from symmetric-key techniques. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1257–1272. ACM, 2017.

[28] Phi Hung Le, Samuel Ranellucci, and S. Dov Gordon. Two-party private set intersection with an untrusted third party. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*, pages 2403–2420. ACM, 2019.

[29] Jack P. K. Ma and Sherman S. M. Chow. Secure-computation-friendly private set intersection from oblivious compact graph evaluation. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 1086–1097. ACM, 2022.

[30] Catherine Meadows. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, pages 134–134. IEEE, 1986.

[31] Peihan Miao, Sarvar Patel, Mariana Raykova, Karn Seth, and Moti Yung. Two-sided malicious security for private intersection-sum with cardinality. In *Advances in Cryptology – CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020*, pages 3–33. Springer International Publishing, 2020.

[32] Shishir Nagaraja, Prateek Mittal, Chi-Yao Hong, Matthew Caesar, and Nikita Borisov. BotGrep: Finding P2P bots with structured graph analysis. In *19th USENIX Security Symposium (USENIX Security 10)*, pages 95–110, 2010.

[33] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, and Dan Boneh. Location privacy via private proximity testing. In *Network and Distributed Security Symposium (NDSS'11)*. The Internet Society, 2011.

[34] Ofri Nevo, Ni Trieu, and Avishay Yanai. Simple, fast malicious multiparty private set intersection. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1151–1165. ACM, 2021.

[35] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — EUROCRYPT '99*, pages 223–238. Springer Berlin Heidelberg, 1999.

[36] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay 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*, pages 401–431. Springer International Publishing, 2019.

[37] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. Phasing: Private set intersection using permutation-based hashing. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 515–530, 2015.

[38] Benny Pinkas, Thomas Schneider, Oleksandr Tkachenko, and Avishay Yanai. Efficient circuit-based PSI with linear communication. In *Advances in Cryptology – EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 122–153. Springer International Publishing, 2019.

[39] Benny Pinkas, Thomas Schneider, and Michael Zohner. Faster private set intersection based on OT extension. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 797–812, 2014.

[40] Benny Pinkas, Thomas Schneider, and Michael Zohner. Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS)*, 21(2):1–35, 2018.

[41] Srinivasan Raghuraman and Peter Rindal. Blazing fast PSI from improved OKVS and subfield VOLE. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2505–2517. ACM, 2022.

[42] Peter Rindal and Mike Rosulek. Malicious-secure private set intersection via dual execution. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1229–1242. ACM, 2017.

[43] Peter Rindal and Phillipp Schoppmann. VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 901–930. Springer International Publishing, 2021.

[44] Mike Rosulek and Ni Trieu. Compact and malicious private set intersection for small sets. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS'21)*, pages 1166–1181. ACM, 2021.

[45] Mingli Wu and Tsz Hon Yuen. Efficient unbalanced private set intersection cardinality and user-friendly privacy-preserving contact tracing. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 283–300, 2023.

[46] Yaxi Yang, Jian Weng, Yufeng Yi, Changyu Dong, Leo Yu Zhang, and Jianying Zhou. Predicate private set intersection with linear complexity. In *International Conference on Applied Cryptography and Network Security*, pages 143–166. Springer Nature Switzerland, 2023.

[47] Foo Yee Yeo and Jason H. M. Ying. Third-party private set intersection. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 1633–1638. IEEE, 2023.

[48] Yongjun Zhao and Sherman S. M. Chow. Are you the one to share? Secret transfer with access structure. *Proceedings on Privacy Enhancing Technologies*, 2017(1):149–169, 2017.

[49] Yongjun Zhao and Sherman S. M. Chow. Can you find the one for me? In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, pages 54–65, 2018.

Foo Yee Yeo, Seagate Technology, Singapore
Email address: fooyee.yeo@seagate.com

Jason H. M. Ying, Seagate Technology, Singapore
Email address: jasonhweiming.ying@seagate.com