A private set intersection protocol based on multi-party quantum computation for greatest common divisor

Muhammad Imran
Institute of Mathematics, Department of Algebra,
Budapest University of Technology and Economics,
Műegyetem rkp. 3., Budapest, H-1111, Hungary.
E-mail: mimran@math.bme.hu

July 4, 2023

Abstract

Private set intersection (PSI) is a cryptographic primitive that allows two or more parties to learn the intersection of their input sets and nothing else. In this paper, we introduce a new secure multi-party quantum computing for the greatest common divisor (GCD) and a private set intersection protocol based on it. The protocol is primarily influenced by the recently published LCM-based quantum private set union protocol by Liu, Yang, and Li. Performance analysis verifies the correctness and demonstrates the entire security of the suggested protocols in the semi-honest model. Additionally, it has been demonstrated that the complexity is efficient in terms of input set size.

Keywords: Multi-party quantum computation, Greatest common divisor, Quantum private set intersection, privacy-preserving matching.

1 Introduction

Private set intersection (PSI) is a crucial cryptographic building block for carrying out joint set operations while protecting privacy. In particular, PSI protocols allow two or more parties to collaboratively compute the intersection of their secret sets without disclosing one another's privacy. The PSI problem is an important problem in secure multi-party computation (MPC) and has many real-world applications, including testing human genomes [1], contact discovery [2], remote diagnosis [3], record linkage [4], privacy-preserving data mining [5], comparing data from cloud storage services [6], determining the distance between two parties [7], etc.

Cryptography is one of the most impacted areas as the subject of quantum computing advances. The majority of extant PSI protocols (and multiparty computing in general) are based on traditional classical cryptosystems that have been shown to be vulnerable in the quantum realm. This necessitates the use of quantum computer resistant PSI. The use of quantum cryptography in the architecture of PSI is an excellent way to overcome these difficulties. Quantum cryptography, which is an integration of quantum physics and classical cryptography, has been extensively researched in a variety of fields, including quantum key distribution [8–12], quantum secret sharing [13–15], and quantum key agreement [16–18]. However, only a few quantum protocols exist for multiparty quantum computation (MPQC), particularly for the private set intersection problem. Shi et al. [19] presented the first quantum protocol for two-party (client and server) PSI in 2016. Later that year, Cheng et al. [20] demonstrated that a dishonest server can manipulate a client's query in the proposed protocol [19], and thus the protocol does not preserve fairness. Maitra [21] presented a novel technique for PSI in 2018 by expanding the oblivious set member decision process of [22]. Recently, Liu et al. [23] suggested a novel quantum PSI based on the quantum Fourier transform in 2021, and Debnath et al. [24] demonstrated a realistic and achievable quantum protocol PSI with single photons and simple single-particle projective measurements at the same time.

In this paper, We provide a novel method for performing private set intersection in the quantum setting. Our PSI protocol is based on another specific purpose multiparty quantum computation protocol, the protocol for finding the greatest common divisor (GCD). The proposed PSI protocol is heavily influenced by Liu et al.'s recent protocol [25] for another set operation, the quantum multiparty private set union (PSU). The PSU protocol is based on the least common multiple (LCM) multiparty quantum computing [26]. Liu's primary idea [25] is transforming the private set union problem into the problem of computing the least common multiple. Specifically, each element of the input sets is encoded to a unique prime number, and hence the input set itself is encoded to a product of primes. Therefore, computing the prime factors of the least common multiple of the encoded secret sets gives a way to obtain the union of all the input sets.

A secure multiparty computation for GCD is necessary to develop quantum multiparty PSI using a similar technique to PSU. However, there is no known protocol for GCD, even in the classical setting, to the best of our knowledge. It was unclear how to build an MPQC for GCD. Using both the multiplication and LCM procedures, the formula $\gcd(x, y) = \frac{xy}{\operatorname{lcm}(x,y)}$ yields the greatest common divisor. However, the formula is only applicable to two integers, and it is obvious that this is not secure in the two-party case because the two-party multiplication protocol always reveals each other's inputs. Furthermore, the recursive generalization of the formula, i.e., $\gcd(a, b, c) = \gcd(a, \gcd(b, c))$, does not give any help to build secure protocol. A simple observation also shows that computing GCD cannot be done using the approach of [26] for LCM which is based on quantum period-finding algorithm [27]. Fortunately, the extension of LCM protocol to the private set union [25] appears to be a promising technique for constructing a secure protocol for GCD. Specifically, we can transform the GCD problem to the private set union problem by working iteratively on the set of prime factors of the secret inputs.

1.1 Our contributions

The first MPQC for computing the greatest common divisor is proposed in this work. The protocol mainly relies on Liu, Yang, and Li's quantum multiparty PSU [25]. Furthermore, using the same idea of the PSU protocol, we construct a quantum multiparty private set intersection (PSI) by transforming the PSI problem into the problem of computing GCD.

1.2 Outline

The rest of the paper is organized as follows: In Section 2, we briefly recall all the necessary tools and protocols for our results: Shor's factoring algorithm, Li-Liu's protocol for LCM, and the quantum multiparty private set union. Section 3 contains all the proposed MPQC protocols: the GCD protocol and the private set intersection protocol. Finally, we present the performance analysis (security and complexity) of the proposed protocols in Section 4.

2 Preliminary

In this section, we give high level descriptions of Shor's factoring algorithm [27], Li-Liu's MPQC protocol for least common multiple [26], and the quantum multiparty private union by Liu et al. [25].

2.1 Shor's factoring algorithm

The well-known Shor's factoring algorithm is able to factor any large integer N efficiently. Shor's factoring algorithm is based on a reduction of factoring to period-finding problem (observed by Miller in the 1970s). The main tool of Shor's factoring (to factor a large integer N) is the quantum period-finding algorithm (QPA) to find the period of the function $f : \mathbb{Z} \to \mathbb{Z}_N$ defined by $f(x) = a^x \bmod N$ (where a is chosen at random), i.e., the smallest positive integer r such that $f(x + r) = f(x)$. Quantum period-finding algorithm in modulo N requires $\mathcal{O}((\log n)n^3)$ quantum operations, with $\mathcal{O}(\log n)$ uses of modular exponentiation where $n = \log N$. The main subroutines of Shor's period-finding algorithm are modular exponentiation and quantum Fourier transform. Modular exponentiation needs $\mathcal{O}(n)$ multiplications [28] and the Quantum Fourier Transform circuit is quadratic in n [27]. Hence, the main steps to find a factor of an odd number N, given quantum period-finding algorithm, is as follows: choose a random $x \bmod N$ and find its period r using the QPA. Finally, compute $\gcd(x^{r/2} - 1, N)$. Since $(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1 = 0 \bmod N$, thus the $\gcd(x^{r/2} - 1, N)$ fails to be a non trivial divisor of N only for r is odd. Hence, the procedure yields a non trivial divisor of N with probability at least $1 - 1/2^{k-1}$, where k is the number of distinct odd prime factors of N. The factoring process will be iterated over the obtained non trivial factors, then all prime factors of N can be found.

We summarize the key steps of the quantum period finding algorithm (note that we skip most of the analysis of the exact parameters for simplicity) as follows:

(1) Prepare two m-registers ($m = \log N$) initialized as $|0\rangle|0\rangle$ and apply QFT over $\mathbb{Z}_N$ to the first register:

$$|0\rangle|0\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |0\rangle.$$

(2) Apply the oracle function f on the second register:

$$\frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |0\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |f(x)\rangle$$

(3) Measures the second register and discarding the measurement outcome. The first register becomes

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |x_0 + jr\rangle,$$

where $n = \lfloor N/r \rfloor$ and x_0 is uniformly random of probability n/N.

(4) Apply another QFT over $\mathbb{Z}_N$ to get:

$$\frac{1}{\sqrt{Nn}} \sum_{j=0}^{n-1} \sum_{k\in\mathbb{Z}_N} \omega_N^{k(x_0+jr)} |k\rangle$$

(5) Measure the register to obtain $k = jN/r$ with probability $\frac{4}{\pi^2 r^2} \geq \frac{1}{3r^2} = \phi(r)/r = \mathcal{O}(\frac{1}{\log\log r})$ and use the continued fraction method to recover r from $k/N = j/r$.

2.2 Li-Liu's MPQC for least common multiple

Multiparty least common multiple problem: Assume that there are n parties: $P_0, \ldots, P_{n-1}$, where each party P_i has a secret integer $r_i \in \{0, 1, \ldots, 2^m - 1\}$. All n parties want to jointly compute the $\mathrm{lcm}(r_0, \ldots, r_{n-1})$ without revealing their respective secrets.

The key idea of Li-Liu's protocol is based on the observation that given functions $f_0, \ldots, f_{n-1}$ with period $r_0, \ldots, r_{n-1}$ respectively, then the function $f(x) = (f_0(x), \ldots, f_{n-1}(x))$ has period $r = \mathrm{lcm}(r_0, \ldots, r_{n-1})$. Thus, each party P_i is equipped with the oracle of the secret function f_i ($|x\rangle|0\rangle \mapsto |x\rangle|f_i(x)\rangle$) and hence together they compute the superposition:

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle|f_0(x)\rangle \ldots |f_{n-1}(x)\rangle$$

where $N = 2^m$. Therefore, the period $r = \mathrm{lcm}(r_0, \ldots, r_{n-1})$ can be found by applying the quantum period-finding algorithm. However, because of the probabilistic nature of the QPA (the probability of the correct output is $\mathcal{O}(1/\log\log r)$), an additional voting procedure is required to check the correctness of the QPA's output. Namely, each party votes whether the output divides their secret input. If the output divides all the secret inputs, then the output passes the verification. The voting procedure is based on the multiparty quantum summation by Shi *et al.* in [29].

We summarize the MPQC protocol for computing LCM as follows:

(1) For each P_i, let $f_i(x) = x \bmod r_i$.

(2) For P_0:

  (a) prepares two m-qubit quantum registers h, t initialized as $|0\rangle_h|0\rangle_t$;

  (b) applies $H^{\otimes m}$ on h:

$$|0\rangle_h|0\rangle_t \mapsto \frac{1}{\sqrt{2^m}} \sum_{j \in [2^m]} |j\rangle_h|0\rangle_t;$$

  (c) applies $CNOT^{\otimes m}$ on h, t, where h controls t:

$$\frac{1}{\sqrt{2^m}} \sum_{j \in [2^m]} |j\rangle_h|0\rangle_t \mapsto \frac{1}{\sqrt{2^m}} \sum_{j \in [2^m]} \sum_{j \in [2^m]} |j\rangle_h|j\rangle_t;$$

  (d) prepares an m-qubit quantum register e_0 initialized as $|0\rangle_{e_0}$;

  (e) applies $U_{f_0} : |j\rangle_t|0\rangle_{e_0} \mapsto |j\rangle_t|f_0(j)\rangle_{e_0}$ on t, e_0 :

$$\frac{1}{\sqrt{2^m}} \sum_{j \in [2^m]} |j\rangle_h|j\rangle_t|0\rangle_{e_0} \mapsto \frac{1}{\sqrt{2^m}} \sum_{j \in [2^m]} |j\rangle_h|j\rangle_t|f_0(j)\rangle_{e_0};$$

  (f) sends t to P_1.

(5) For P_i, $1 \le i \le n-1$:

  (a) prepares an m-qubit registers e_i initialized as $|0\rangle_{e_i}$;

  (b) applies $U_{f_i} : |j\rangle_t|0\rangle_{e_i} \mapsto |j\rangle_t|f_i(j)\rangle_{e_i}$ on t, e_i :

$$\frac{1}{\sqrt{k}} \sum_{j \in [k]} |j\rangle_h|j\rangle_t|f_0(j)\rangle_{e_0}|f_(j)\rangle_{e_1} \ldots |f_{i-1}(j)\rangle_{e_{i-1}}|0\rangle_{e_i}$$

$$\mapsto \frac{1}{\sqrt{k}} \sum_{j\in[k]} |j\rangle_h |j\rangle_t |f_0(j)\rangle_{e_0} |f(j)\rangle_{e_1} \dots |f_{i-1}(j)\rangle_{e_{i-1}} |f_i(j)\rangle_{e_i};$$

(c) sends t to P_{i+1}.

(6) For P_0:

(1) applies $CNOT^{\otimes m}$ on h, t, where h controls t:

$$\frac{1}{\sqrt{k}} \sum_{j\in[k]} |j\rangle_h |j\rangle_t |f(j)\rangle_e \mapsto \frac{1}{\sqrt{k}} \sum_{j\in[k]} |j\rangle_h |0\rangle_t |f(j)\rangle_e,$$

where $f(j) = f_0(j)|| \dots ||f_{n-1}(j)$, $e = (e_0, \dots, e_{n-1})$;

(2) measures t, if t is not $|0\rangle$, then rejects, otherwise continues;

(3) Applies QPA to find the period r of f;

(4) Broadcasts r to all other parties.

The total computation and communication complexity of Li-Liu's protocol is $\mathcal{O}(n^3m^2)$ and $\mathcal{O}(n^2m)$ respectively. However, considering the success probability of the standard QPA, Li-Liu's protocol needs $\mathcal{O}(\log\log r) \leq \mathcal{O}(\log(nm))$ repetitions. A simple observation can show that the repetition itself can lead to some possible attacks. Specifically, the parties can learn a factor of other's inputs in each repetition from the incorrect outputs and their own secrets. The information about factors of the secret inputs is a nontrivial information that can be derived from the output which is the least common multiple, which is not good. Moreover, the risk increases as the repetition grows (the size m of the inputs and the number of participants n grow), especially in the malicious model.

Remark 1. *Liu et al. [25] proposed an improved QPA based on an extended Knill's technique [30] (which is a trade off between classical and quantum computations) to increase the probability to $\mathcal{O}(1)$ to avoid the required repetition (but still probabilistic, there is still a small probability that the output of the protocol is incorrect). However, as mentioned in [30], this increases the complexity of the original QPA by a factor $(c + 1)^2$ of classical computations if the QPA runs twice in parallel (in [25], the QPA runs $s > 2$ times in parallel, which gives worse bound). Furthermore, it is also noted in [30] that the trade off is worth in some specific cases.*

Remark 2. *Imran [31] proposed an exact quantum period finding algorithm (EQPA). Using EQPA instead of the standard QPA, the voting procedure in the LCM protocol [26] can disregarded. However, as EQPA requires a multiple of the period r, a few modification of Li-Liu's LCM protocol is needed. Specifically, we need to perform the multiparty quantum multiplication in [29] to compute $k = \prod_{i=0}^{n-1} r_i$ (a multiple of the period) as the first step and use EQPA instead of QPA in step (6.c). Since the EQPA's complexity is $\mathcal{O}(\log^4 k)$, then the modified version of Li-Liu's protocol [26] has also computation complexity $\mathcal{O}(\log^4 k) = \mathcal{O}(n^4m^4)$ while the communication complexity remains $\mathcal{O}(n^2m)$.*

A simpler zero-error LCM protocol: There is a simpler alternative to fix the drawbacks of Li-Liu's LCM protocol. The key idea is instead of using voting procedure, P_0 can test whether the output r of the standard QPA is the correct period by comparing the values of $f(x)$ and $f(x + r)$ by the exact swap test in [32]. If the values are equal, then $c = r$ is the desired period which is the least common multiple of all the secret inputs. Finally, P_0 broadcasts r to all other parties.

As the exact swap test for $f(x)$ and $f(x+r)$ has asymptotically the same complexity as the cost of the quantum query of the function f, then the above modified LCM protocol has the same complexity as the original Li-Liu's protocol.

2.3 Quantum multiparty private set union

Private set union problem: Assume that there are n parties: $P_1, \dots, P_n$, where each party P_i has a secret set $S_i \subseteq U = \{1, 2, 3, \dots, N\}$: $2^{m-1} < N \leq 2^m$. All n parties want to jointly compute the $\bigcup S_i$ without revealing their respective secret.

The key idea of the quantum multiparty private set union proposed by Li, Yang, and Liu consists of three main steps: encoding procedure, an improved quantum multiparty computation for LCM, and decoding procedure. The encoding procedure transforms all elements of the secret set S_i (for all $1 \leq i \leq n$) to prime numbers and hence encode the set S_i as the product of prime numbers image of all its elements. After the encoding procedure, the MPQC protocol for LCM (based on an improved QPA) is performed to find the LCM of all the encoded S_i. Finally, decoding procedure is done by (an improved) Shor's algorithm to get the union from the prime factors of the LCM obtained in the previous procedure.

We briefly summarize the quantum protocol for private set union as follows:

(1) *Encoding phase:* Let p_j denotes the jth prime. Each party P_i encode the elements a of the respective set S_i by $a \mapsto p_a$ and hence encode $S_i \mapsto \prod_{a \in S_i} p_a$.

(2) *LCM Protocol:* All parties jointly perform the LCM (only until the step 6.3) of their encoded sets such that P_1 gets:

$$M = \operatorname{lcm}\left(\prod_{a_1 \in S_1} p_{a_1}, \prod_{a_2 \in S_2} p_{a_2}, \dots, \prod_{a_n \in S_n} p_{a_n} \right).$$

(3) *Decoding phase:* P_1 computes the set of the prime factor of M using Shor's algorithm. Decode the prime factors: $p_a \mapsto a$. The set of the decoded prime factors is exactly the union $\bigcup_{i=1}^{n} S_i$. Broadcasts the union to all other parties.

The computation and communication complexity of the protocol are claimed to be $\mathcal{O}(n^3 m^3 k^3 \log(nmk))$ and $\mathcal{O}(n^2 mk)$ respectively where k is the upper bound of the cardinalities of the secret inputs S_i.

Remark 3. *We can use the proposed zero-error LCM protocol in the subsection 2.2 instead of the improved multiparty quantum computation for LCM in remark 1 (which is still probabilistic) as the subroutine of the PSU protocol to unconditionally guarantee its correctness.*

3 Proposed MPQC protocols

3.1 Multiparty quantum computation for GCD.

Multiparty greatest common divisor problem: Assume that there are n parties: $P_0, \dots, P_{n-1}$, where each party P_k has a secret integer $r_k \in \{0, 1, \dots, 2^m - 1\}$. All n parties want to jointly compute the $\gcd(r_1, \dots, r_n)$ without revealing their respective secret.

The key idea of our proposed protocol is by transforming the greatest common divisor problem into the private set union problem of all sets of prime factors of each secret inputs and then finally, apply voting procedure to obtained the greatest prime power of each prime factors in the union set obtained.

We summarize the protocol for computing greatest common divisor as follows.

(1) Each party P_i ($0 \leq i \leq n-1$): Apply Shor's factoring algorithm on the respective secret input r_i to obtain the set R_i of all prime factors of r_i.

(2) All parties jointly perform the private set union protocol to get the set $R = \bigcup_{i=0}^{n-1} R_i$.

(3) For each prime $p \in R$, do the following iteration: using the voting procedure as in [26], all parties jointly vote whether $p, p^2, \ldots$ divide their secret inputs in order to get the largest power p^k that simultaneously divides all their secret inputs. Finally, the GCD can be obtained by the product of all the largest prime power of all elements of R.

Correctness proof. In the first step, each party performs Shor's factoring on their inputs to get the set of all prime factors of r_i. Therefore, each party can easily verify that they hold a correct set of prime factors of their inputs before applying private set union protocol in the next step. Since the correctness of the PSU protocol in the second step follows directly from the remark 3. Then it is left to show that the last step indeed gives the GCD of the secret inputs r_i's. The last step indeed gives a correct output according to the definition of greatest common divisor

$$\gcd(p_1^{a_1} \cdots p_m^{a_m}, p_1^{b_1} \cdots p_m^{b_m}) = p_1^{\max\{a_1,b_1\}} \cdots p_m^{\max\{a_m,b_m\}}$$

which is true for computing GCD for any n numbers through the prime factorization.

3.2 Multiparty quantum private set intersection

Private set intersection problem: Assume that there are n parties: $P_1, \ldots, P_n$, where each party P_i has a secret set $S_i \subseteq U$ where U is the complete set of cardinality N: $2^{m-1} < N \leq 2^m$. All n parties want to jointly compute the $\bigcup S_i$ without revealing their respective secret.

The protocol for private set intersection straightforwardly follows the protocol for private set union. We give the key steps of the protocol as follows:

(1) *Encoding phase:* Let p_j denotes the jth prime. Each party P_i encode the elements a of the respective set S_i by $a \mapsto p_a$ and hence encode $S_i \mapsto \prod_{a \in S_i} p_a$.

(2) *GCD Protocol:* All parties jointly perform the GCD (only until the step 6.3) of their encoded sets such that P_1 gets:

$$M = \gcd\left(\prod_{a_1 \in S_1} p_{a_1}, \prod_{a_2 \in S_2} p_{a_2}, \ldots, \prod_{a_n \in S_n} p_{a_n}\right).$$

(3) *Decoding phase:* P_1 computes the set of the prime factor of M using Shor's algorithm. Decode the prime factors back: $p_a \mapsto a$. The set of the decoded prime factors is exactly the union $\bigcap_{i=1}^{n} S_i$. Broadcasts the union to all other parties.

Correctness proof. Since the correctness of the GCD protocol has been proven, then it is left to show that the prime factors of the GCD indeed gives the intersection of all input sets S_i. Let $Enc(S_i)$ denotes the encoding image of S_i which is the product of prime images p_a of all elements a of S_i. For any element $u \in U$ such that p_u divides $Enc(S_i)$ for all $1 \leq i \leq n$, p_u divides the GCD M. Conversely, if there exists a set S_j that does not contain an element $u \in U$, then p_u is not a factor of $Enc(S_j)$ and hence p_u is not a factor of the gcd M. Therefore, an element $u \in \bigcap_{i=1}^{n} S_i$ must correspond to a factor of the GCD M.

4 Performance analysis

4.1 Security analysis

The private set intersection protocol is based on the multiparty quantum computation for GCD. Therefore, the security of PSI protocol is mainly follows from the security of the GCD protocol. However, as the GCD protocol is based on the private set union [25] in which its security based on the LCM protocol [26], let us first briefly recall the security analysis of the LCM protocol. Li et al. have proved the security against the following semi-honest attacks.

(1) Direct measurement attack: Before the QPA process is completed, P_i measures the register h, t or e_i to obtain any useful information.

(2) Pre-period-finding attack: Before the QPA process is completed, P_i applies the invers QFT to his own registers h, t to obtain the LCM of the parties who have completed their operations.

(3) Post-period-finding attack: P_i copies the register $|j\rangle_t$ using the $CNOT^{\otimes m}$ and wait until the QPA process is completed. Then, P_i applies QFT to his own copy $|j\rangle$ to obtain any useful information.

All three attacks cannot leak any useful information from the other secrets mainly because of the entanglement of the registers and the honest test in step 6.2 of the protocol. However, because of the probabilistic nature of the standard QPA, the protocol should be repeated ($\mathcal{O}(\log nm)$) followed by a verification procedure to check whether the QPA's output is correct. The repetition itself leads to some possible attacks. Specifically, the parties can learn information about some factors of other parties from the incorrect outputs and their own secrets.

In order to resolve the drawback of Li-Liu's protocol, we proposed 2.2 to use the exact quantum period-finding algorithm [31] by adding some additional steps to fulfill the requirement of the EQPA, namely a multiple of the period. In the first step, each P_i sends $y_i = r_i q$ to P_0 so that P_0 can compute a multiple of the period. However, P_0 cannot gain any useful information as y_i is a multiplication of the secret input r_i with a random element q. The modified version needs no repetition as the EQPA always gives a correct output with certainty, and hence the modified protocol is completely secure in the semi-honest model.

As a conclusion, the proposed protocol for GCD and PSI are completely secure in the semi-honest model following the security of the modified LCM protocol and the PSU [25].

4.2 Complexity analysis

Firstly, we analyze the complexity of the GCD protocol 3.1. The use of Shor's factoring in the first step of the protocol costs $\mathcal{O}(nm^2 \log m)$ computational complexity. As for the second step, the computational and communication complexity of the private set union are $\mathcal{O}(n^3m^3k^3 \log(nmk))$ and $\mathcal{O}(n^2mk)$ respectively where k is the upper bound of the cardinality of the sets R_i's. Thus, the second step has $\mathcal{O}(n^3m^6 \log(nm^2))$ computational complexity and communication complexity $\mathcal{O}(n^2m^2)$. Finally, in the last step, there are at most m iterations of voting procedure, thus the computational and communication complexity of the last step are $\mathcal{O}(nm^3)$ and $\mathcal{O}(nm^2)$ respectively following the complexity of the voting procedure in [26]. Hence, the total computational and communication complexity are $\mathcal{O}(n^3m^6 \log(nm^2))$ and $\mathcal{O}(n^2m^2)$ respectively. On the other hand, using the EQPA to get a deterministic output in the subroutine of the PSU protocol gives extra computational complexity with total computational complexity $\mathcal{O}(n^4m^6 \log(nm^2))$ instead of $\mathcal{O}(n^3m^6 \log(nm^2))$.

For the private set intersection protocol, it is easily seen that the most expensive computational cost comes from the use of GCD protocol in the second step. Therefore, the total computation and communication complexity of the PSI protocol coincide the complexity of the GCD protocol.

References

[1] Liyan Shen, Xiaojun Chen, Dakui Wang, Binxing Fang, and Ye Dong. Efficient and private set intersection of human genomes. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 761–764. IEEE, 2018.

[2] Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. Pir-psi: scaling private contact discovery. *Cryptology ePrint Archive*, 2018.

[3] Justin Brickell, Donald E Porter, Vitaly Shmatikov, and Emmett Witchel. Privacy-preserving remote diagnostics. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 498–507, 2007.

[4] Xi He, Ashwin Machanavajjhala, Cheryl Flynn, and Divesh Srivastava. Composing differential privacy and secure computation: A case study on scaling private record linkage. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1389–1406, 2017.

[5] Ji Young Chun, Dowon Hong, Ik Rae Jeong, and Dong Hoon Lee. Privacy-preserving disjunctive normal form operations on distributed sets. *Information Sciences*, 231:113–122, 2013.

[6] Zeeshan Pervez, Ammar Ahmad Awan, Asad Masood Khattak, Sungyoung Lee, and Eui-Nam Huh. Privacy-aware searching with oblivious term matching for cloud storage. *The Journal of Supercomputing*, 63:538–560, 2013.

[7] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, Dan Boneh, et al. Location privacy via private proximity testing. In *NDSS*, volume 11, 2011.

[8] CH BENNET. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, Dec. 1984*, pages 175–179, 1984.

[9] Artur K Ekert. Quantum cryptography based on bell's theorem. *Physical review letters*, 67(6):661, 1991.

[10] Charles H Bennett. Quantum cryptography using any two nonorthogonal states. *Physical review letters*, 68(21):3121, 1992.

[11] Adán Cabello. Quantum key distribution in the holevo limit. *Physical Review Letters*, 85(26):5635, 2000.

[12] Han-Cheng Shih, Kuo-Chang Lee, and Tzonelih Hwang. New efficient three-party quantum key distribution protocols. *IEEE Journal of Selected Topics in Quantum Electronics*, 15(6):1602–1606, 2009.

[13] Mark Hillery, Vladimír Bužek, and André Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829, 1999.

[14] Anders Karlsson, Masato Koashi, and Nobuyuki Imoto. Quantum entanglement for secret sharing and secret splitting. *Physical Review A*, 59(1):162, 1999.

[15] Li Xiao, Gui Lu Long, Fu-Guo Deng, and Jian-Wei Pan. Efficient multiparty quantum-secret-sharing schemes. *Physical Review A*, 69(5):052307, 2004.

[16] Nanrun Zhou, Guihua Zeng, and Jin Xiong. Quantum key agreement protocol. *Electronics Letters*, 40(18):1, 2004.

[17] Song-Kong Chong, Chia-Wei Tsai, and Tzonelih Hwang. Improvement on "quantum key agreement protocol with maximally entangled states". *International Journal of Theoretical Physics*, 50:1793–1802, 2011.

[18] Song-Kong Chong and Tzonelih Hwang. Quantum key agreement protocol based on bb84. *Optics Communications*, 283(6):1192–1195, 2010.

[19] Run-hua Shi, Yi Mu, Hong Zhong, Jie Cui, and Shun Zhang. An efficient quantum scheme for private set intersection. *Quantum Information Processing*, 15:363–371, 2016.

[20] Xiaogang Cheng, Ren Guo, and Yonghong Chen. Cryptanalysis and improvement of a quantum private set intersection protocol. *Quantum Information Processing*, 16:1–8, 2017.

[21] Arpita Maitra. Quantum secure two-party computation for set intersection with rational players. *Quantum Information Processing*, 17:1–21, 2018.

[22] Run-hua Shi, Yi Mu, Hong Zhong, and Shun Zhang. Quantum oblivious set-member decision protocol. *Physical Review A*, 92(2):022309, 2015.

[23] Wen Liu and Han-Wen Yin. A novel quantum protocol for private set intersection. *International Journal of Theoretical Physics*, 60(6):2074–2083, 2021.

[24] Sumit Kumar Debnath, Kunal Dey, Nibedita Kundu, and Tanmay Choudhury. Feasible private set intersection in quantum domain. *Quantum Information Processing*, 20:1–11, 2021.

[25] Wenjie Liu, Qi Yang, and Zixian Li. Quantum multi-party private set union protocol based on least common multiple and shor's algorithm. *International Journal of Quantum Information*, 2023.

[26] Zixian Li and Wenjie Liu. a quantum secure multiparty computation protocol for least common multiple. *arXiv preprint arXiv:2210.08165*, 2022.

[27] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[28] Donald E Knuth. *The art of computer programming: Volume 3: Sorting and Searching*. Addison-Wesley Professional, 1998.

[29] Run-hua Shi, Yi Mu, Hong Zhong, Jie Cui, and Shun Zhang. Secure multiparty quantum computation for summation and multiplication. *Scientific reports*, 6(1):1–9, 2016.

[30] E Knill. On shor's quantum factor finding algorithm: Increasing the probability of success and tradeoffs involving the fourier transform modulus. Technical report, Citeseer, 1995.

[31] Muhammad Imran. An exact version of shor's order finding algorithm. *arXiv preprint arXiv:2205.04240*, 2022.

[32] M Imran and Gábor Ivanyos. An exact quantum hidden subgroup algorithm and applications to solvable groups. *Quantum Inf. Comput.*, 22(9-10):770–789, 2022.