Quantum Private Set Intersection Cardinality Protocol With Application to Privacy-Preserving Condition Query

Run-Hua Shi and Yi-Fei Li

***Abstract*— Private Set Intersection Cardinality (PSI-CA) is one of the most concerned issues with the protection of privacy, in which two parties jointly compute the intersection cardinality without revealing their respective private sets. There are important applications of PSI-CA in real society, e.g., strongly privacy-preserving data statistics in contact tracing for health authorities to fight the outbreaks of highly contagious diseases. In this paper, we present a novel quantum PSI-CA protocol, in which we adopt oblivious quantum key distribution, secure quantum summation and quantum counting algorithm. The proposed PSI-CA protocol not only ensures the approximatively perfect security but also achieves the linear communication complexity, i.e., $O(N)$. Furthermore, we define a new privacy protection problem, i.e., Privacy-preserving Condition Query (PCQ), and provide an efficient solution to the PCQ problem based on the proposed quantum PSI-CA protocol. Finally, we verify the correctness and the feasibility of the proposed quantum PSI-CA protocol by circuit simulations in IBM Qiskit.**

***Index Terms*— Quantum computing, quantum key distribution, secure multiparty computation, circuit simulations.**

I. Introduction

NOWADAYS, privacy protection becomes a focus of attention in the cryptography community. Accordingly, a variety of privacy-preserving issues emerge, among which Private Set Intersection (PSI) [1] is a well-known primitive protocol that enables two parties to jointly compute the intersection of their respective private sets without revealing any private information.

There are lots of practical applications of PSI, especially in both privacy-preserving and data-sharing settings [2]–[5]. For example, PSI can allow a person to determine if the data they gathered in contact tracing matches the dataset of diagnosed patients without revealing their private information. Please note that contact tracing is a powerful countermeasure that can be utilized to control the spread of infection during a global pandemic of the novel coronavirus.

Manuscript received November 6, 2021; revised January 13, 2022; accepted February 16, 2022. Date of publication March 3, 2022; date of current version May 27, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61772001. This article was recommended by Associate Editor M. Mozaffari Kermani. *(Corresponding author: Run-Hua Shi.)*

The authors are with the School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China (e-mail: rhshi@ncepu.edu.cn; 1085037470@qq.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2022.3152591.

Digital Object Identifier 10.1109/TCSI.2022.3152591

However, for specific security settings with higher privacy requirements (e.g., strongly privacy-preserving data statistics), PSI reveals information excessively. Furthermore, Private Set Intersection Cardinality (PSI-CA) is relatively advantageous as it meets higher security requirements, which outputs only the cardinality of the intersection, rather than any element.

In 2004, Freedman *et al*. [1] first considered the problem of computing the intersection of two-party private sets and its variants. Accordingly, they presented several PSI and PSI-CA protocols by using homomorphic encryption and oblivious polynomial evaluation. Due to its importance, subsequently, there appeared many PSI-CA protocols [6]–[14]. Among these PSI-CA protocols, especially, Cristofaro *et al.* [7] presented a PSI-CA protocol with linear computation and communication complexity. They employed Diffie-Hellman key exchange to blind the private information and built an efficient PSI-CA protocol based on the difficulty assumptions of the discrete logarithm problem, which achieves linear complexities in the size of two sets. Recently, Pinkas, *et al*. [15] proposed novel circuit-based protocols for computing variants of PSI with an almost linear number of comparisons via Cuckoo Hashing. In 2019, they further presented the first circuit-based PSI protocol with a true linear complexity [16], which is also concretely more efficient than all previous circuit-based PSI protocols.

However, the security of most existing PSI-CA protocols is based on the difficulty assumptions (e.g., the difficulty of factoring and finding a discrete logarithm), which are vulnerable to the attacks by quantum computers or fast quantum algorithms [17]–[19]. As a consequence, the designing of quantum-resistant PSI-CA becomes one of the hot research topics in classical cryptography. On the other hand, quantum cryptography [20]–[23] has emerged as an important complement to classical cryptography, whose security mainly depends on the fundamental laws of quantum mechanics so that it can guarantee unconditional security in theory.

In this paper, we present a quantum approach to solve the PSI-CA problem, which can be roughly divided into three stages. Firstly, the client and the server generate two auxiliary datasets associated with their respective original sets by adopting oblivious quantum key distribution. Secondly, two parties securely compute the summations of two auxiliary datasets by the help of quantum random access memory. Finally, the client rightly outputs the cardinality of their intersection by

executing quantum counting algorithm. The proposed quantum PSI-CA protocol indeed achieves the linear communication complexity, i.e., $O(N)$ qubits instead of $O(N)$ quantum messages, so it is more suitable for big data applications. The security of the proposed protocol is based on the physical principles of quantum mechanics, instead of the difficulty assumptions, and hence it has the advantage of higher security compared to classical related protocols. What's more, we focus on PSI-CA's applications. Especially, we consider a new but interesting query problem, later called Privacy-preserving Condition Query (PCQ). For example, there is a statistics table of all students' test scores, which is kept by a dean in secret. Though each student knows his/her score, he/she further wants to learn how many students' scores are higher or less than his/her score, but he/she does not like to reveal his/her identity (i.e., name) and other privacy (e.g., score). Besides, each query should not reveal others' privacy yet. Moreover, e.g., an employer privately keeps a form, which records all employees' salaries. Similarly, an employee wants to know his/her salary level by privately querying how many employees' salaries are within a region near to his/her salary. Finally, based on the proposed quantum PSI-CA protocol, we construct a novel quantum solution to the PCQ problem.

Our contributions in this paper are summarized below.

(1) We design an efficient encoding method based on oblivious quantum key distribution and further compute Private Set Intersection Cardinality by classical and quantum hybrid technologies, e.g., classical one-time pad, quantum summing and quantum counting.

(2) We present a secure and efficient quantum protocol to privately compute the summations of two datasets.

(3) We first define an interesting privacy query, i.e., Privacy-preserving Condition Query, and present its corresponding solutions based on quantum protocols proposed above.

(4) Finally, we design the simulated quantum circuits and verify the correctness and the feasibility of the proposed quantum protocols by circuit simulations in IBM Qiskit.

II. Related Works

A. Quantum PSI/PSI-CA Protocol

In 2016, Shi *et al*. [24] first presented a cheat-sensitive quantum protocol for Private Set Intersection (PSI) using phase-encoded private query. At the same time, they designed a two-party quantum protocol for Private Set Intersection Cardinality (PSI-CA) [2], which can output a good estimator of the intersection cardinality with high probability and small error. Compared with the classical relevant protocols, the proposed quantum PSI-CA protocol has the higher security and the lower communication complexity. Especially, it achieves the communication complexity of $O(1)$, which is fully independent of the size of data sets. However, this protocol requires some additional assumptions about the cardinalities of the input sets [25], which may limit its wider applications. In 2018, Shi successfully discarded these assumptions and presented a stronger quantum PSI-CA protocol without any limitation using secret splitting, quantum multiplication and other operators [25]. However, these protocols need the complicated oracle operators. Subsequently, in order to enhance the realizability, Shi *et al*. introduced a non-colluding third party to help two legitimate parties to compute the intersection cardinality with single photons [26] or EPR pairs [27]. Recently, Liu *et al*. presented an improved PSI-CA protocol with single photons using Bloom filter in Ref. [28].

In addition, Li *et al*. [19], Zhang *et al*. [29], and Wang *et al*. [30] extended two-party PSI-CA to multi-party PSI-CA and presented the corresponding multi-party quantum PSI-CA (PUI-CA) protocols.

Like quantum key distribution (QKD), the feasibility of quantum PSI/PSI-CA protocols is the focus of research. However, there are two intractable issues to implement these existing quantum protocols: one is that it is difficult to implement the complicated oracle operators and measurements in high-dimensional Hilbert space and the other is that it is hard to find a fully trusted third party in the real world.

Therefore, it has always been our goal to design the practical and feasible quantum PSI/PSI-CA protocols with the present quantum technology.

B. Oblivious Quantum Key Distribution

In 2011, to ensure the better feasibility of quantum private query, Jakobi *et al*. [31] proposed a practical quantum key distribution protocol, hereafter called oblivious quantum key distribution (OQKD), in which an oblivious key can be distributed between two legitimate parties by using SARG04 QKD [32], where the sender knows the whole key while the receiver only knows a few bits of the key. The main process of OQKD can be briefly described as follows: The sender randomly prepares a long sequence of photons which are in one of four polarized states $\{|0\rangle, |1\rangle, |+x\rangle, |-x\rangle\}$ and then sends the photon sequence to the receiver in order, where each photon carries a bit of classical information, e.g., $|0\rangle$ and $|1\rangle$ represent the bit 0, while $|+x\rangle$ and $|-x\rangle$ denote the bit 1. After receiving each photon sent from the sender, the receiver measures it randomly in $\{|0\rangle, |1\rangle\}$ basis or $\{|+x\rangle, |-x\rangle\}$ basis. For each photon that the receiver has successfully measured, the sender announces a pair of states: one that has been sent actually and the other that is selected randomly from the other basis. However, the receiver does not know which one is true. For example, if $|+x\rangle$ has been sent, the sender will announce $\{|1\rangle, |+x\rangle\}$ or $\{|0\rangle, |+x\rangle\}$ at random. This process is the same as in the SARG04 QKD [32]. According to the sender's declaration and his measurement result, the receiver can successfully identify the carried bit value with the probability of 1/4. By this way, it asymmetrically distributes a secret string between two parties, such that the string is known entirely to the sender but in a quarter to the receiver. Furthermore, in order to reduce the receiver's information on the string, two parties cut the raw string into multiple substrings of length N, and add these strings bitwise to obtain the final key with length N.

Later, Gao *et al*. generalized Jakobi's protocol and proposed a similar 4-state O-QKD protocol [33], which uses four generalized states $\{|0\rangle, |1\rangle, |0'\rangle, |1'\rangle\}$, where $|0'\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle$ and $|1'\rangle = \cos\theta\,|0\rangle - \sin\theta\,|1\rangle\}$.

TABLE I
DEFINITIONS OF NOTATIONS

Notations	Definitions
k_B	The raw key distributed by Bob
k_b	The intermediate key after checking Bob's honesty
k_b^*	The final key distributed by Bob
$k_b(i)$ / $k_b^*(i)$	The ith bit of the key k_b / k_b^*
π	A permutation
C, S	Two input sets
$\|C\|$ / $\|S\|$	The set cardinality
c_i / s_i	The ith element of the set C / S
$c(i)$ / $s(i)$	The ith element of the ancillary set encoded by the set C / S
r	A random number
$\|x\rangle_A$	The state $\|x\rangle$ in the quantum register A

Gao's protocol exhibits better database security than Jakobi's protocol. To further improve the performance, Yang *et al*. also proposed a flexible B92-based O-QKD protocol [34].

III. PROPOSED QUANTUM PSI-CA PROTOCOL

A. Security Model

In the following protocols, we only consider the honest-but-curious parties, like the semi-honesty model in the classical settings, where adversaries may try to learn as much information as possible from a given protocol execution but are not able to deviate from the protocol steps [16]. Like most classical/quantum secure multi-party protocols, we can use classical/quantum bit commitment, zero-knowledge proof, and other verifiable technologies to ensure that the parties honestly execute the protocol.

Definition 1: Private Set Intersection Cardinality (PSI-CA) protocol – There are two parties, a client with a private set C and a server with a private set S. Suppose that $|C| = n$ and $|S| = m$, and n and m are public. After running a PSI-CA protocol, the client outputs the cardinality of the intersection of their respective private sets, i.e., $|C \cap S|$, but the server gets nothing. In addition, a perfect security PSI-CA protocol should meet the following privacy requirements:

1) Server Privacy: The client learns no information about the server's set except $|C \cap S|$ and $|S|$.

2) Client Privacy: The server cannot get any private information about the client's set.

We first present a protocol (i.e., Protocol I) to distribute a special oblivious key between two parties, conventionally called Alice and Bob, such that Bob knows all bits of the key, while Alice only knows the partial bits of the key, where each known bit is just associated with a unique element of her private set. For example, suppose that Alice has a private set $\{a_1, a_2 \cdots a_t\}$, where $a_i \in \{0, 1, 2, \ldots N-1\}$ and $t < N$. Then, Alice only knows the a_1th, a_2th, ..., and a_tth bits of the key. Here, we assume that the position indexes of the key bits start from 0 to $N-1$.

Protocol I (The Special OQKD Protocol Associated With a Private Set):

Step 1: Alice and Bob jointly call Jakobi *et al*.'s Oblivious Quantum Key Distribution (OQKD) protocol [31] to share a random $(N+q)$-bit key k_B, where Bob knows the whole key k_B, and Alice only knows $t+q$ bits of k_B (Note. t is the cardinality of Alice's private set and q is a security parameter).

Step 2: Furthermore, among these $t+q$ bits, Alice randomly chooses q bits to check Bob's honesty. That is, she requests Bob to announce the values of these checked bits. If these values published by Bob aren't entirely consistent with those she has deciphered, it will show that Bob is dishonest or there is an outside eavesdropper. If Alice finds a cheat of Bob or any outside eavesdropping, she will terminate this protocol, otherwise, continue to the next step.

Step 3: Bob discards q checked bits of the raw key k_B and further gets the intermediate key k_b of the length N, such that Alice only knows t bits of the key k_b, while Bob still knows all bits. Of course, Alice knows not only t-bit values: $k_b(j_1)$, $k_b(j_2), \ldots, k_b(j_t)$, but also their respective position indexes: $j_1, j_2, \ldots, j_t$, where $k_b(j_i)$ denotes the j_ith bit of k_b. However, Bob does not know which bits Alice knows.

Step 4: Alice generates a random permutation π of an N-element sequence by the position index set $\{j_1, j_2, \ldots, j_t\}$ and her private set $a_1, a_2, \cdots a_t$, but which must meet the following condition

$$\{k(j_1), k(j_2), \ldots, k(j_t)\} = \{k^*(a_1), k^*(a_2), \ldots, k^*(a_t)\}, \tag{1}$$

where k^* is the new sequence after applying the permutation π to the N-element sequence k, i.e., $k^* = \pi(k)$. Then Alice declares the permutation π to Bob.

Step 5: Bob applies the permutation π to the key k_b to get the final oblivious key $k_b^* = \pi(k_b)$. Obviously, Alice knows its partial bits: $k_b^*(a_1), k_b^*(a_2), \ldots, k_b^*(a_t)$, where $k_b^*(a_i)$ denotes the a_ith bit of the final key k_b^* for $i = 1, 2, \ldots, t$. However, Bob does not know any secret information about the set $a_1, a_2, \cdots a_t$ without $\{j_1, j_2, \ldots, j_t\}$.

Here, we give a simple example to illustrate how to generate an oblivious key between Alice and Bob, as shown in Figure 1. In Figure 1, Alice has a private set {3,9,12} over Z_{16}, and thus finally she only knows $k_b^*(3)$, $k_b^*(9)$ and $k_b^*(12)$, while Bob knows all bits of k_b^*.

Furthermore, based on Protocol I, we present a novel quantum PSI-CA protocol (i.e., Protocol II). Suppose that the client's private set $C = \{c_1, c_2, \cdots, c_n\}$ and the server's private set $S = \{s_1, s_2, \cdots s_m\}$, and all elements of the sets C and S lie in Z_N, where $Z_N = \{0, 1, 2, \ldots, N-1\}$. In addition, we assume that n and m are public and $n, m < N$. The proposed protocol consists of 9 steps, which are described in detail as follows.

Protocol II (The Quantum PSI-CA Protocol):

//The following protocols from Step 1 to Step 3 are to privately generate two ancillary sets $\{c(0), c(1), \ldots, c(N-1)\}$ and $\{s(0), s(1), \ldots, s(N-1)\}$ based on Protocol I.

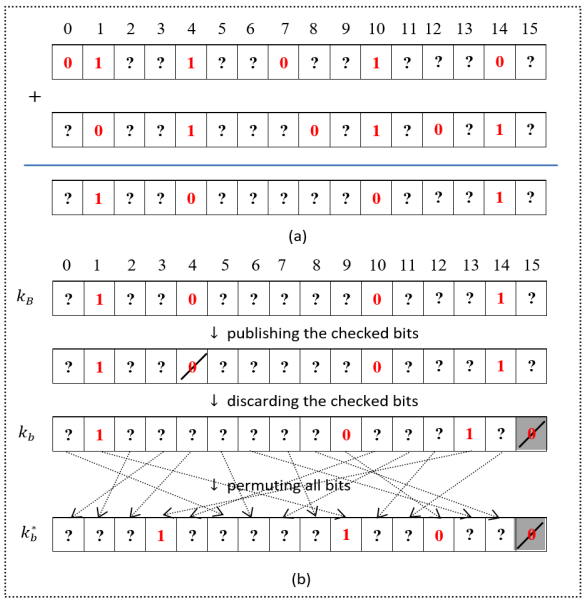


Fig. 1. Illustration of generating the oblivious key. (a) How to reduce Alice's information on the key. (b) How to process the raw key k_B to get the final key k_b^*.

Step 1: The client with the private set $\{c_1, c_2, \cdots, c_n\}$ requests the server to jointly execute Protocol I to generate a random N-bit key k_s^*, where the server knows the whole key k_s^*, while the client only knows its n bits: $k_s^*(c_1)$, $k_s^*(c_2)$, ..., $k_s^*(c_n)$. Here $k_s^*(c_i)$ denotes the c_i-th bit of the key k_s^* (please refer to the red digits in Figure 3). Furthermore, based on these known bits, the client computes $k_{c_s}^*(i)$s for $i = 0, 1, \ldots, N-1$, which are defined by,

$$k_{c\_s}^*(i) = \begin{cases} k_s^*(i), & if\ i \in \{c_1, c_2, \cdots, c_n\} \\ -2, & if\ i \notin \{c_1, c_2, \cdots, c_n\} \end{cases} \tag{2}$$

From Eq.(2), we can easily see that $k_s^*(i) - k_{c\_s}^*(i) = 0$ if $i \in \{c_1, c_2, \cdots, c_n\}$ (i.e., $i \in C$); Otherwise $k_s^*(i) - k_{c\_s}^*(i) = 2$ or 3 because $k_s^*(i) \in \{0, 1\}$ and $k_{c\_s}^*(i) = -2$ (i.e., $i \notin C$).

Step 2: The server with the private set $s_1, s_2, \cdots s_m$ asks the client to jointly run Protocol I to generate another random N-bit key k_c^*, where the client knows the whole key k_c^*, while the server only knows m bits of the key k_c^*: $k_c^*(s_1)$, $k_c^*(s_2)$, ..., $k_c^*(s_m)$ (please see the red digits in Figure 3). Furthermore, based on these known bits, the server computes $k_{s_c}^*(i)$s for $i = 0, 1, \ldots, N-1$, which are defined by,

$$k_{s\_c}^*(i) = \begin{cases} k_c^*(i), & if\ i \in \{s_1, s_2, \cdots, s_m\} \\ -2, & if\ i \notin \{s_1, s_2, \cdots, s_m\} \end{cases} \tag{3}$$

From Eq. (3), we can easily see that $k_c^*(i) - k_{s\_c}^*(i) = 0$ if $i \in S$; Otherwise $k_c^*(i) - k_{s\_c}^*(i) = 2$ or 3 because $k_c^*(i) \in \{0, 1\}$ and $k_{s\_c}^*(i) = -2$ (i.e., $i \notin S$).

Step 3: Furthermore, the client and the server respectively compute $c(i)$s and $s(i)$s for $i = 0, 1, \ldots, N-1$, which are defined by the following equations:

$$c(i) = [k_c^*(i) - k_{c\_s}^*(i)] mod N, \quad \text{for } i = 0, 1, \ldots, N-1, \tag{4}$$

$$s(i) = [k_s^*(i) - k_{s\_c}^*(i)] mod N, \quad \text{for } i = 0, 1, \ldots, N-1. \tag{5}$$

Please note that $c(i) + s(i) = 0$ if $i \in C \cap S$, which will be proven later in Theorem 1.

//The following protocols from Step 4 to Step 8 are to compute the summations of the sequences $c(i)$s and $s(i)$s for $i = 0, 1, \ldots, N-1$.

Step 4: The client prepares a quantum random access memory (QRAM) [35] with N data registers, where the address register of the QRAM contain a superposition state $\frac{1}{\sqrt{N}}\sum_{i=0}^{N}|i\rangle$, and each data register stores a basis state $|c(i)\rangle$ in M dimension Hilbert space (for simplicity, later set $M = N$). That is, the QRAM outputs the following state $|\varphi\rangle$,

$$|\varphi\rangle = \frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a|c(i)\rangle_d. \tag{6}$$

Step 5: The client randomly selects an integer $r \in \{0, 1, \ldots, N-1\}$ and further performs an Add operator U_{add_r} on the state $|\varphi\rangle$, which implements $\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c(i)\rangle_d \rightarrow \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c(i)+r\rangle_d$. Let $c^*(i) = c(i) + r$ and

$$\left|\varphi^*\right\rangle = \frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d. \tag{7}$$

Step 6: The client prepares another ancillary register A in $\log N$-qubit basis state $|0\rangle_A$ and further sends two registers $|i\rangle_a$ and $|0\rangle_A$ through $\log N$ $CNOT$ operators, where each qubit of $|i\rangle_a$ is the control qubit and the corresponding qubit of $|0\rangle_A$ is the target qubit. Then, the client will get the following state $|\phi\rangle$,

$$\begin{aligned} |\phi\rangle &= CNOT^{\otimes \log N}[\left|\varphi^*\right\rangle \otimes |0\rangle_A] \\ &= CNOT^{\otimes \log N}[\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d \otimes |0\rangle_A] \\ &= \frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d|i\rangle_A. \end{aligned} \tag{8}$$

Finally, the client holds the ancillary register A in secret, and sends the remaining two registers a and d to the server through the quantum channels.

Step 7: After successfully receiving the two registers a and d, the server performs another Add operator U_{add_S} on them, where the Add operator U_{add_S} is defined by (see Figure 2),

$$U_{add\_S}|i\rangle\left|c^*(i)\right\rangle = |i\rangle\left|c^*(i) + s(i)\right\rangle. \tag{9}$$

That is,

$$\begin{aligned} U_{add\_S}|\phi\rangle &= U_s\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d|i\rangle_A \\ &= \frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a|c^*(i)+s(i)\rangle_d|i\rangle_A. \end{aligned} \tag{10}$$

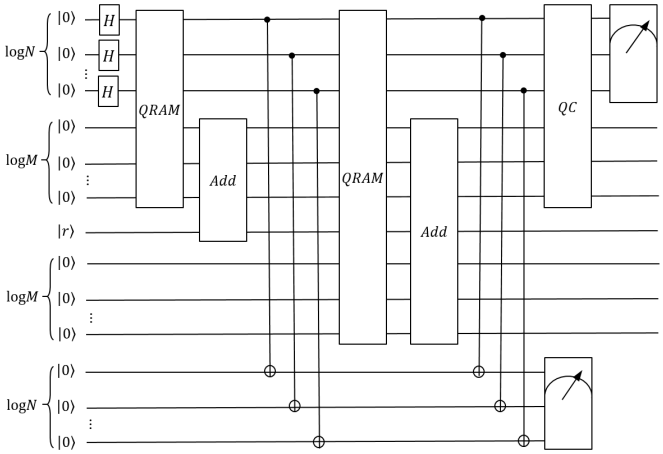


Fig. 2. Circuit diagrams of the main quantum process of Protocol II.

After finishing the Add operator U_{add_S}, the server sends two registers a and d back to the client through the quantum channels.

Step 8: After successfully receiving the registers from the server, the client carries out an honest test: he again performs $\log N$ $CNOT$ operators on two registers $|i\rangle_a$ and $|i\rangle_A$ to get the following state,

$$CNOT^{\otimes \log N}[\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)+s(i)\rangle_d|i\rangle_A] = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)+s(i)\rangle_d|0_A. \quad (11)$$

Furthermore, the client measures the state of the ancillary register A in the computational basis. If the measured result is $|0\rangle_A$, then he will continue to execute the next step; Otherwise, he will believe that the server is dishonest or there is outside eavesdropping and terminate this protocol. In the next step, we will denote the state of the remaining system as $|\psi\rangle$, i.e.,

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)+s(i)\rangle_d. \quad (12)$$

Step 9: The client further counts the number of components satisfying $c^*(i)+s(i)=r$ (i.e., $c(i)+s(i)=0$) in the state $|\psi\rangle$ by calling quantum counting algorithm [36], as the final output of the client, i.e., the cardinality of the intersection, $|\{c_1, c_2, \cdots, c_n\} \cap \{s_1, s_2, \cdots s_m\}|$.

The corresponding circuits of the main quantum process of Protocol II are shown in Figure. 2

IV. Analysis

A. Correctness

Theorem 1: The proposed quantum PSI-CA protocol (i.e., Protocol II) is correct.

Proof: Let $C = \{c_1, c_2, \cdots, c_n\}$ and $S = \{s_1, s_2, \cdots s_m\}$. By Protocol II, we will get that,

$$\begin{aligned} i \in C \cap S &\Longleftrightarrow i \in C \wedge i \in S \\ &\Longleftrightarrow k_s^*(i) - k_{c\_s}^*(i) = 0 \wedge k_c^*(i) - k_{s\_c}^*(i) = 0 \\ &\quad \text{(by Eqs. (2) and (3))} \\ &\Longleftrightarrow k_s^*(i) - k_{c\_s}^*(i) + k_c^*(i) - k_{s\_c}^*(i) = 0 \\ &\quad \text{(by } k_s^*(i) - k_{c\_s}^*(i), k_c^*(i) - k_{s\_c}^*(i) \in \{0, 2, 3\}) \\ &\Longleftrightarrow k_s^*(i) - k_{s\_c}^*(i) + k_c^*(i) - k_{c\_s}^*(i) = 0 \\ &\Longleftrightarrow c(i) + s(i) = 0 \text{ (by Eqs. (4) and (5))} \\ &\Longleftrightarrow c^*(i) + s(i) = r \text{ (by } c^*(i) = c(i) + r) \end{aligned}$$

So, the number of $c^*(i)+s(i)=r$ in all components of the state $|\psi\rangle$ is equal to the cardinality of the intersection of their respective private sets, i.e., $|C \cap S|$. Therefore, the proposed protocols are correct.

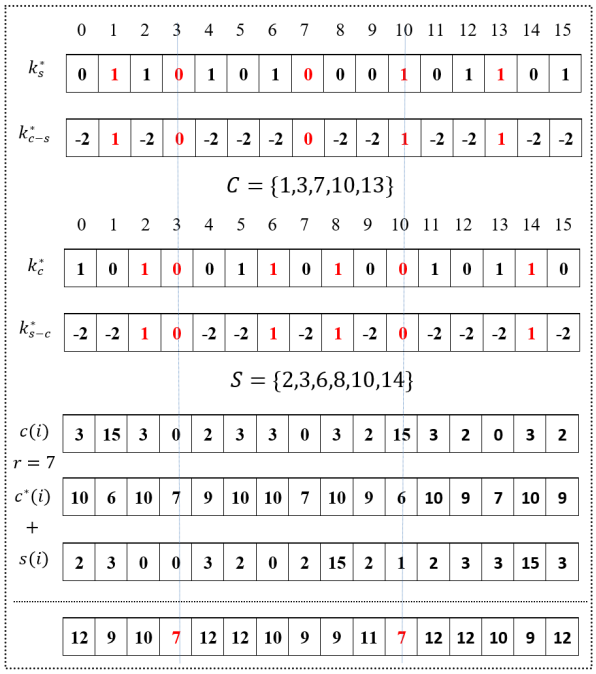


Fig. 3. An example of privately computing $|C \cap S|$.

Furthermore, we take an example to clearly illustrate Protocol II, as shown in Figure 3. In our example, the client and the server have a private set $C = \{1, 3, 7, 10, 13\}$ and $S = \{2, 3, 6, 8, 10, 14\}$ over Z_{16}, respectively. After calling Protocol I, the client secretly gets k_{c-s}^* by his private set C, where the position index set of the red digits in k_{c-s}^* is just equal to his private set C. Similarly, the server secretly gets k_{s-c}^* by his private set S, where the position index set of the red digits in k_{s-c}^* is also just equal to his private set S. From Figure 3, obviously, $k_s^*(i) - k_{c-s}^*(i) = 0$ if $i \in C$, and $k_c^*(i) - k_{s-c}^*(i) = 0$ if $i \in S$ (please see the red digits). So, $c(i)+s(i) = 0$, if $i \in C \cap S$. Therefore, the number of zero in the $c(i)+s(i)$ sequence is just equal to $|C \cap S|$. We assume $r = 7$. Then $c(i)+s(i) = 0$ implies $c^*(i)+s(i) = 7$. That is, the number of 7 in the $c^*(i)+s(i)$ sequence is equal to $|C \cap S|$. In our example, $|C \cap S| = 2$, which is exactly equal to the number of the red digits of the last sequence in Figure 3.

B. Security

Furthermore, we will analyze the security of proposed quantum PSI-CA protocols, which mainly includes Server Privacy and Client Privacy.

1) Server Privacy: Theorem 2: Protocol I can ensure the privacy of the set of the server. Specifically, in Protocol I, if Bob is dishonest, i.e., he wants to eavesdrop on Alice's private set, then the probability that his dishonesty will be detected later by Alice is at least $1-\frac{1}{2^q}$, where q is the secure parameter.

Proof: In Step 2 of Protocol II, the server asks the client to call Protocol I to generate an oblivious key k_c^*, which is associated with the private set of the server. Here, the server and the client play the roles of Alice and Bob in Protocol I, respectively. Furthermore, we will analyze that if Bob is dishonest in Protocol I, the probability that his dishonesty will be detected later by Alice is at least $1-\frac{1}{2^q}$, where q is the secure parameter.

The security of Step 1 of Protocol I is guaranteed by Jakobi *et al.*'s OQKD protocol [31]. By the analysis of Ref. [31], a dishonest Bob will introduce bit errors. That is, if Bob gains information on the conclusiveness of Alice's bits, he will lose information on the bit values Alice has recorded. In fact, it is impossible for Bob to have both the correct bit value and the conclusiveness information of Alice's measurement [31] (i.e., the index of the correct basis). Therefore, Bob cannot simultaneously obtain the bit value, $k_b(j)$, which is the correct result deciphered by Alice, and its corresponding index j.

Furthermore, in Step 2 of Protocol I, Alice randomly compares q bits rightly deciphered by herself with the corresponding bits announced by Bob to decide whether there are bit errors introduced by Bob's dishonesty. Please note that Bob cannot know which bits in the raw key will be taken as the checked bits before Alice declaring them. Moreover, for each checked bit, if Bob does not honestly execute the protocols and gets its position information beforehand by some cheating, he will lead to an error probability of $\frac{1}{2}$ later in the honest test, which is similar to the detection technologies of decoy states. So, for a dishonest Bob, the successful probability to completely pass the honest test in Step 2 of Protocol I is not more than $\frac{1}{2^q}$. That is, the probability that his dishonesty will be detected later by Alice is at least $1-\frac{1}{2^q}$. In addition, by this honest test, Alice can also easily find the attack that Bob prepares and sends all the same quantum states (e.g., all $|0\rangle$s, or all $|1\rangle$s).

Finally, in Step 4 of Protocol I, Alice declares the permutation π to Bob, which is defined by two sets $\{j_1, j_2, \ldots, j_t\}$ and $\{a_1, a_2, \cdots a_t\}$. Then we will analyze the conditional probability $p(\{j_1, j_2, \ldots, j_t\}, \{a_1, a_2, \cdots a_t\}|\pi)$. Here, the permutation π is randomly selected by Alice, but it must satisfy the equation of Eq. (1). That is, Alice declares a random permutation π with t fixed points, where fixed points are private but the permutation are public. Accordingly, the number of the permutations satisfying the condition is $t!(N-t)!$, instead of $N!$.

For simplicity, let JA denotes two sets $\{j_1, j_2, \ldots, j_t\}$ and $\{a_1, a_2, \cdots a_t\}$. Then we can deduce the following results (Note. $p(|)$ and $\mathrm{I}(;)$ denote the conditional probability and the mutual information, respectively):

$$p(\pi) = \frac{1}{N!} \tag{13}$$

$$p(\pi|JA) = \frac{1}{t!(N-t)!} \tag{14}$$

$$p(JA) = \frac{1}{C_N^t \cdot C_N^t} \tag{15}$$

$$\begin{aligned}\mathrm{I}(\pi; JA) &= \log\frac{p(\pi|ja)}{p(\pi)} \\ &= \log\frac{\frac{1}{t!(N-t)!}}{\frac{1}{N!}} = \log\frac{N!}{t!(N-t)!}.\end{aligned} \tag{16}$$

$$\begin{aligned}\mathrm{I}(JA) &= -\log p(JA) = -\log\frac{1}{C_N^t \cdot C_N^t} \\ &= 2\log C_N^t = 2\log\frac{N!}{t!(N-t)!}.\end{aligned} \tag{17}$$

$$\begin{aligned}\mathrm{I}(JA|\pi) &= \mathrm{I}(JA) - \mathrm{I}(JA;\pi) \\ &= 2\log\frac{N!}{t!(N-t)!} - \log\frac{N!}{t!(N-t)!} \\ &= \log\frac{N!}{t!(N-t)!}.\end{aligned} \tag{18}$$

$$\mathrm{I}(JA|\pi) = -\log p(JA|\pi). \tag{19}$$

$$p(JA|\pi) = \frac{1}{\frac{N!}{t!(N-t)!}} = \frac{1}{C_N^t}. \tag{20}$$

That is, though it is not unconditionally secure, i.e., information-theoretically secure, the probability of successfully guessing two sets $\{j_1, j_2, \ldots, j_t\}$ and $\{a_1, a_2, \cdots a_t\}$ by the public permutation π is small enough, i.e., $\frac{1}{C_N^t}$, which is negligible.

Furthermore, we know that $p(A) = 1/C_N^t$. So, $p(JA|\pi) = p(A)$. That is, the probability of successfully guessing two sets $\{j_1, j_2, \ldots, j_t\}$ and $\{a_1, a_2, \cdots a_t\}$ with the public permutation π is equal to that of directly guessing $\{a_1, a_2, \cdots a_t\}$ without the permutation π. Please note that we have previously proved that the set $\{j_1, j_2, \ldots, j_t\}$ is private to Bob. So, it is not easier for Bob to get the private set $\{a_1, a_2, \cdots a_t\}$ even if Alice opens the permutation π.

In a word, error detection mechanisms (i.e., q checked bits) guarantee the honesty of Bob and the probability of successfully guessing the private sets by the public permutation is sufficiently small, i.e., $1/C_N^t$. Therefore, Protocol I can ensure the privacy of the set of the server well.

Theorem 3: A dishonest client in Protocol II can get at most one $s(i)$, but he also loses the chance to count the intersection cardinality.

Proof: On the one hand, by calling Protocol I, the privacy information of the set S is finally hidden in the dataset $\{s(i)|i = 0, 1, \ldots, N-1\}$. Furthermore, in Step 7 of Protocol II, the server helps the client to compute the summation of $c^*(i) + s(i)$ by applying the Add operator U_{add_S} (see Eqs. (9) and (10)). Accordingly, the client gets the state $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i) + s(i)\rangle_d$ after the honest test, where the reduced density matrixes of subsystems stored in two registers a and d are as follows:

$$\rho_a = tr_d(\rho_{ad}) = \frac{1}{N}\sum_{i=0}^{N-1}|i\rangle\langle i|, \tag{21}$$

$$\begin{aligned}\rho_d = tr_a(\rho_{ad}) = \frac{1}{N}\sum_{i=0}^{N-1}|c^*(i) \\ + s(i)\rangle\langle c^*(i) + s(i)|.\end{aligned} \tag{22}$$

Though all $s(i)$s for $i = 0, 1, \ldots, N-1$ have been inserted into the quantum state $|\psi\rangle$, the client can get at most one $s(i)$ by the physical laws of quantum mechanics, because the maximum Von Neumann entropy of the log$N$ qubits is equal to log$N$, which is an upper bound on the accessible information from the log$N$ qubits. That is,

$$S(\rho_d) \leq S\left(\frac{I}{N}\right) = -\sum \frac{1}{N}\log\frac{1}{N} = \log N. \quad (23)$$

On the other hand, if the client wants to get one $s(i)$ by measuring the quantum state $|\psi\rangle$ in the computational basis, obviously he will lose the chance to count the cardinality of the intersection.

In addition, the client might perform a cheating strategy as follows: To get private information about the set S, the client makes all $k_c^*(i) = 0$ and $c(i) = 0$ for $i = 0, 1, \ldots, N-1$. Accordingly, he can later get the number of $s(i) = 0$ (or 1) after running Protocol II. That is, the client will finally output $|S|$, not $|C \cap S|$. However, $|S|$ is public in PSI-CA protocol. So, this cheating strategy is infeasible.

By the no-go theorem, unconditionally secure two-party computations cannot be implemented theoretically without rigorous space-time constraints. The proposed protocol still complies with the no-go theorem. In short, a dishonest client in Protocol II may get at most one $s(i)$ and further decide whether i belongs to the set S by the value of $s(i)$, but he also loses the chance to count the intersection cardinality. Besides, the number of $s(i)$s is N in total, so the possible information leakage rate is at most $1/N$. When N is large enough, e.g., $N \approx 2^{160}$, $1/N$ is negligible. So, our proposed protocol is more suitable for big datasets in Z_N.

2) Client Privacy: Theorem 4: Protocol I can ensure the privacy of the set of the client.

Similarly, Protocol I can ensure the privacy of the set of the client when generating the oblivious key k_s^* in Step 1 of Protocol II, where the server knows the whole key, while the client only gets its partial bits. Here the proof is abbreviated (see the proof of **Theorem 2**).

Theorem 5: After successfully executing Protocol I in Protocol II, the server cannot get any private information about the client's set. So, the privacy of the client in Protocol II is guaranteed by Protocol I.

Proof: In Step 5, the client sends the server two registers a and d, which contain two subsystems of the whole quantum system $|\phi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d|i\rangle_A$. After receiving the two registers a and d, the server can perform the following two attacks:

The first attack is to directly measure the subsystem in the register d to get some information about $c^*(i)$. Similarly, in the general case, measurement of the state in the register d would give the only $c^*(i)$ for a definite i, that is, the server could get at most one $c^*(i)$ by the fundamental principles of quantum mechanics.

The second attack is an entangle-measure attack [37] that the server first prepares an ancillary quantum system and further entangles his ancillary quantum system and the encoded system sent from the client by a local unitary operator, and afterwards, he can measure the ancillary quantum system to get the partial information about the client's private inputs. The server's dishonest action can be described by a local unitary operator $\tilde{U}$, which is defined by,

$$\tilde{U}\left|c^*(i)\right\rangle_d|0\rangle_{\tilde{a}} = \sqrt{\eta_i}\left|c^*(i)\right\rangle_d|\xi(c^*(i))\rangle_{\tilde{a}} + \sqrt{1-\eta_i}|V(c^*(i))\rangle_{d\tilde{a}}, \quad (24)$$

where the subscript of $\tilde{a}$ denotes the ancillary register prepared by the server, and $|V(c^*(i))\rangle_{d\tilde{a}}$ is a vector orthogonal to $|c^*(i)\rangle_d|\xi(c^*(i))\rangle_{\tilde{a}}$, i.e.,

$${}_d\langle c^*(i)|_{\tilde{a}}\langle\xi(c^*(i))|V(c^*(i))\rangle_{d\tilde{a}} = 0. \quad (25)$$

In order to completely pass the honest test (see Step 8), we can easily deduce $\eta_j = 1$. That is, the whole quantum systems should be in the following state after performing the operator $\tilde{U}$:

$$\tilde{U}\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d|i\rangle_A|0\rangle_{\tilde{a}} = \frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a\left|c^*(i)\right\rangle_d|i\rangle_A|\xi(c^*(i))\rangle_{\tilde{a}}. \quad (26)$$

Then the server performs his Add operator U_{add_S} and sends two registers a and d back to the client, while he keeps $|\xi(c^*(i))\rangle_{\tilde{a}}$ in his hands. After successfully passing the honest test, the state of the remaining quantum systems becomes,

$$\frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}|i\rangle_a\left|c^*(i)+s(i)\right\rangle_d|\xi(c^*(i))\rangle_{\tilde{a}}. \quad (27)$$

To simplify the analysis, we omit the register a and only consider the following state,

$$\frac{1}{\sqrt{N}}\sum\nolimits_{i=0}^{N-1}\left|c^*(i)+s(i)\right\rangle_d|\xi(c^*(i))\rangle_{\tilde{a}}. \quad (28)$$

By the above pure state, we can further get the reduced density matrix of the register d and the corresponding Von Neumann entropy,

$$\rho_d = \frac{1}{N}\sum\nolimits_{i=0}^{N-1}\left|c^*(i)+s(i)\right\rangle_d\langle c^*(i)+s(i)|, \quad (29)$$

$$S(\rho_{\tilde{a}}) = S(\rho_d) \leq \log N. \quad (30)$$

That is, if the server measures his ancillary state $|\xi(c^*(i))\rangle_{\tilde{a}}$, he will not get more information about $c^*(i)$ than he would if he had measured the register d directly. What's more, if the server measures his ancillary state this moment, the quantum state held by the client will become $|i\rangle_a|c^*(i)+s(i)\rangle_d$. Of course, accordingly, the client will not right get the cardinality of the intersection.

Furthermore, we assume that the client executes quantum counting algorithm [36]. That is, he first prepares a register in the initial state $\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1}|j\rangle$ and implements

$$\frac{1}{\sqrt{M}}\sum\nolimits_{j=0}^{M-1}|j\rangle\otimes|\phi\rangle \rightarrow \frac{1}{\sqrt{M}}\sum\nolimits_{j=0}^{M-1}|j\rangle G^j|\phi\rangle, \quad (31)$$

where G is Grover iteration or Grover operator [36], [38], and $|\phi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|c^*(i)+s(i)\rangle_d|\xi(c^*(i))\rangle_{\tilde{a}}$. The client further applies an inverse Quantum Fourier Transform to the

first register. Then, the whole quantum system will be in the following state [36]:

$$\frac{e^{i\pi\omega}}{\sqrt{2}}|\phi_+\rangle + \frac{e^{-i\pi\omega}}{\sqrt{2}}|\tilde{x}_-\rangle|\phi_-\rangle, \tag{32}$$

where

$$|\tilde{x}_+\rangle = \sum_{x=0}^{M-1}\{\frac{1}{M}\sum_{j=0}^{M-1} e^{i2\pi j(\omega-\frac{x}{M})}\}|x\rangle, \tag{33}$$

$$|\tilde{x}_-\rangle = \sum_{x=0}^{M-1}\{\frac{1}{M}\sum_{j=0}^{M-1} e^{i2\pi j[(1-\omega)-\frac{x}{M}]}\}|x\rangle, \tag{34}$$

$$|\phi_+\rangle = \frac{1}{\sqrt{2}}(|\beta\rangle - i|\alpha\rangle), \tag{35}$$

$$|\phi_-\rangle = \frac{1}{\sqrt{2}}(|\beta\rangle + i|\alpha\rangle), \tag{36}$$

$$|\alpha\rangle = \frac{1}{\sqrt{N}}\sum_{i:c^*(i)+s(i)=r} \left|c^*(i)+s(i)\right\rangle_d |\xi(c^*(i))\rangle_{\tilde{a}}, \tag{37}$$

$$|\beta\rangle = \frac{1}{\sqrt{N}}\sum_{i:c^*(i)+s(i)\neq r} \left|c^*(i)+s(i)\right\rangle_d |\xi(c^*(i))\rangle_{\tilde{a}}. \tag{38}$$

Finally, the client measures the first register in the computational basis. The remaining quantum system will be collapsed into $|\phi_+\rangle$ or $|\phi_-\rangle$. Then, we can also get that,

$$S(\rho_{\tilde{a}}) = S(\rho_d) \leq \log N. \tag{39}$$

So, if the server measures his ancillary state $|\xi^*(c^*(i))\rangle_{\tilde{a}}$ this moment, he will get less than $\log N$ bits of information.

By analyzing the above two attacks, we can see that the server can get less than or equal to $\log N$ bits of private information. In fact, we can deduce that the server can get at most $\log N$ bits of private information (i.e., $c^*(i)$) for any possible attack. Since the whole system is in $|\phi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_a|c^*(i)\rangle_d|i\rangle_A$, where all private information of the client is embedded in the register d, the reduced density matrix of the register d and the corresponding Von Neumann entropy satisfy

$$\rho_d = \frac{1}{N}\sum_{i=0}^{N-1}\left|c^*(i)\right\rangle_d\langle c^*(i)|, \tag{40}$$

$$S(\rho_d) \leq \log N. \tag{41}$$

Furthermore, the quantum subsystem in the register d can be characterized by the quantum ensemble $\gamma \equiv p_i, \rho_d(i)$, where $p_i = \frac{1}{N}$ is the server's probability of getting the secret $c^*(i)$. So, whenever the server performs any attack, he can get at most one $c^*(i)$ by Holevo's theorem. However, $c^*(i) = c(i) + r$, where r is random and private. It is equivalent to one-time pad. So, the server cannot learn any information about $c(i)$ without knowing r. Therefore, after successfully executing Protocol I in Protocol II, the server cannot get any private information about the client's set due to the random number r. That is, the privacy of the client in Protocol II is guaranteed by Protocol I, where the security of Protocol I was previously analyzed in Theorem 2.

To sum up, private information of two sets is first hidden in respective private sequences $c(i)$s and $s(i)$s by using Protocol I, where the probabilities of not finding the dishonesty of the party and successfully guessing private information by public information are small enough and negligible. Furthermore, the procedures of summing $c^*(i)+s(i)$ in Protocol II can ensure the perfect security (i.e., information-theoretical security) of the client. In addition, the dishonest client may get at most one $s(i)$, but he also loses the chance to count the intersection cardinality. So, to ensure that it finally outputs the correct result, our proposed protocol achieves the approximatively perfect security in the semi-honesty model.

C. Performance

In Protocol I, it takes 4-state polarized photons as quantum resources. There is not any other quantum operator except the projective measurements of single photons. Thus, it is easy to implement this protocol due to its required quantum resources and measurements.

In Protocol II, the most complicated quantum transformation is to compute the summation of each component of two states based on Quantum Random Access Memory. Besides, it still needs other ordinary operators, e.g., *CNOT* operator, Grover operator and inverse Quantum Fourier Transform (i.e., to implement quantum counting), where most of these quantum operators have been implemented by the newest reports [39]–[42]. Furthermore, we give detailed performance comparisons of our proposed PSI-CA protocol with other related protocols in terms of the main quantum resources, the required operators and measurements, the transmitted qubits, feasibility, and security features, respectively, which are listed in Table II.

From Table II, we can see that two previously proposed protocols in Refs. [2] and [25] has better communication complexity, but they require complicated oracle operators. At present, it is still difficult to implement these complicated oracle operators. Furthermore, other related protocols in Refs. [26]–[28] have better feasibility, but they ask a trusted third party (TP) to help two parties to prepare quantum resources and count the final cardinality. However, it is hard to find a completely trusted third party in the real world. Therefore, under consideration of both feasibility and security, our protocol has better performance: On the one hand, our proposed protocol does not need a trusted TP. On the other hand, it is relatively feasible to implement our protocol (please see the later simulation experiments).

In addition, quantum communication complexity of our proposed PSI-CA protocol is $O(N)$, which is independent of the size of sets. As previously stated, the best classical method needs the linear communication complexity. It implies that the method needs to exchange $O(n)$ classical messages or conduct $O(n)$ circuit-based comparisons [15], [16], and each classical message is about $O(\log N)$ bits, where n is the size of the set ($n \leq N$) and each element of the set lies in Z_N. So, the communication complexity of these methods should be $O(n\log N)$ bits. However, our protocol needs to transmit $O(N)$ qubits, i.e., the communication complexity of our proposed protocol is $O(N)$ qubits. Therefore, our proposed PSI-CA protocol is more suitable for sets with big sizes.

Finally, we simulate the proposed quantum PSI-CA protocol in Qiskit of IBM (Qiskit-0.23.2; Python-3.8.6; OS-Linux). The whole circuit diagram of this simulated experiment is shown in Figure 4, mainly including the circuits of Input, QRAM,

TABLE II
THE PERFORMANCE COMPARISONS

Protocols	Quantum resources	Quantum operators	Quantum measurements	Transmitted qubits	Feasibility	Trusted TP
Shi *et al*. [2]	2log$N$-qubit entangled states	Complicated oracle operators	log$N$-qubit projective measurements	$O(\log N)$	Difficult	No
Shi *et al*. [25]	4log$N$-qubit entangled states	Complicated oracle operators	log$N$-qubit projective measurements	$O(m\log N)$	Difficult	No
Shi *et al*. [26]	Single Photons	Single-photon Pauli operators	Single-photon measurements	$O(N)$	Better	Yes
Shi *et al*. [27]	Bell states	Single-photon Pauli operators	Bell measurements	$O(N)$	Better	Yes
Liu *et al*. [28]	Single Photons	Single-photon Pauli operators	Single-photon measurements	$O(N)$	Better	Yes
Our protocol	Single Photons + 3log$N$-qubit entangled states	Quantum summing+ quantum counting	single-photon measurements+ log$N$-qubit projective measurements	$O(N)$	Good	No

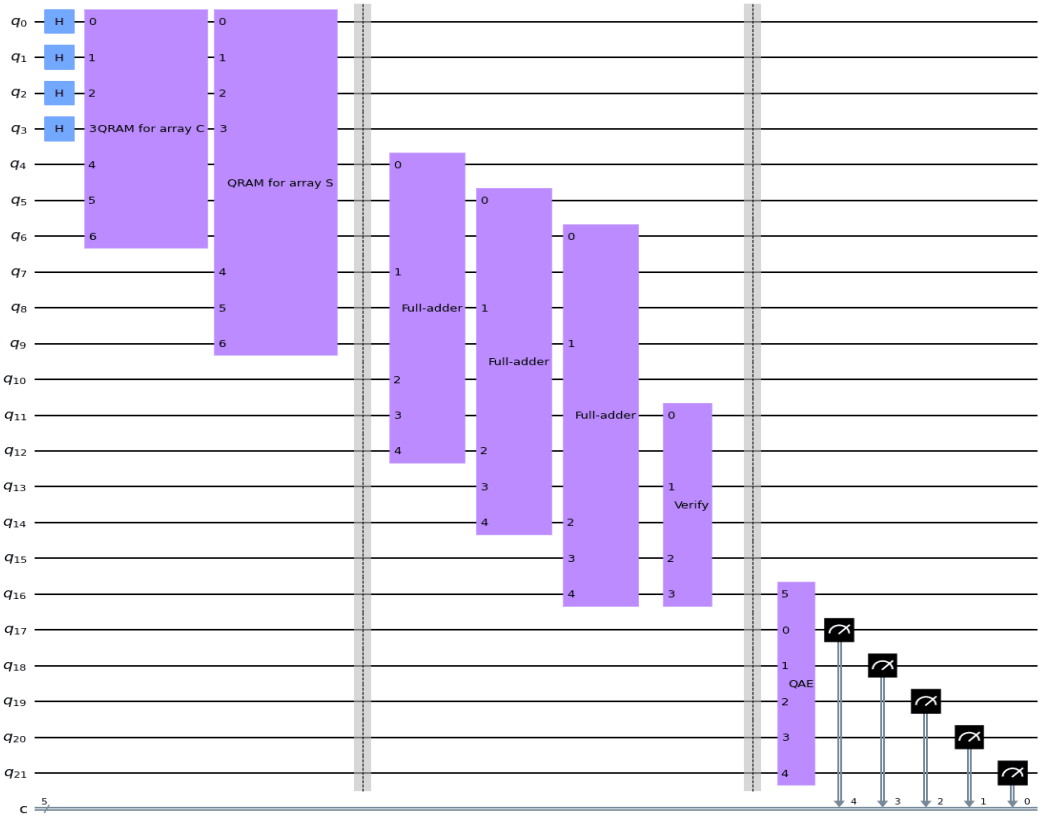


Fig. 4. Simulated Circuits of proposed PSI-CA.

Full-adder, Verify and quantum amplitude estimation (QAE), respectively.

For simplicity, we assume that $c(i), s(i) \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ and $i \in Z_{16}$ in our simulated experiments. Accordingly, we represent $c(i)$ and $s(i)$ by using 3 qubits, respectively. For example, in Figure 4, $c(i)$ and $s(i)$ are encoded into $\{q_4, q_5, q_6\}$ and $\{q_7, q_8, q_9\}$, respectively, while 4 qubits of $\{q_0, q_1, q_2, q_3\}$ denote the address index, i.e., i.

Figure 5 shows quantum circuits of an instance of QRAM (i.e., to generate $\sum |i\rangle|d(i)\rangle$), where $\{q_0, q_1, q_2, q_3\}$ denote the address index and $\{q_4, q_5, q_6\}$ stores all classical $d(i)s$, e.g., [0,7,2,3,0,1,2,2,3,1,7,1,3,1,7,2].

Figure 6 denotes quantum circuits of Full-adder, which computes the summation of two input bits, encoded into q_0 and q_1. In addition, q_2, q_3 and q_4 denote the qubits corresponding to the carry bit of input, the final summation, and the carry bit of output, respectively. Therefore, three Full-adders implement $c(i) + s(i)$ by the bitwise summations in Figure 4.

After the bitwise summations, we combine "Verify" with "quantum amplitude estimation" instead of quantum counting

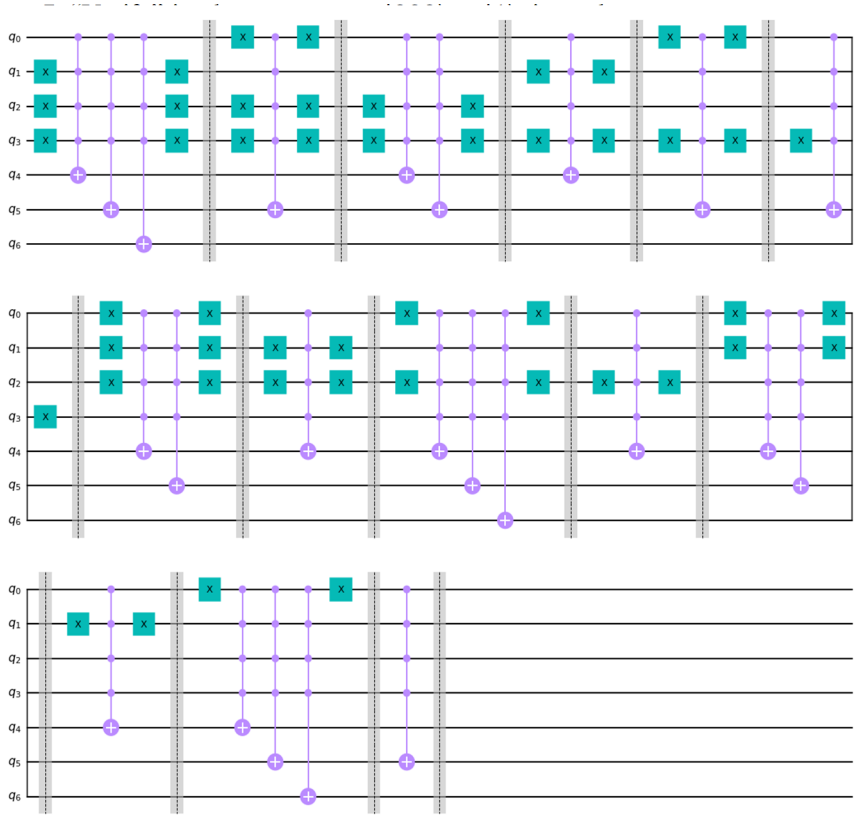


Fig. 5. Quantum circuits of an instance of QRAM.

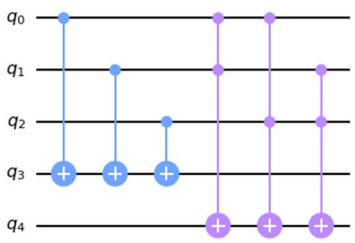


Fig. 6. Quantum circuits of full-adder.

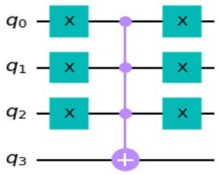


Fig. 7. Quantum circuits of verify.

to estimate the number of $c(i) + s(i) = 000$ (i.e., $r = 0$). In Figure 7, "Verify" implements a map: $|000\rangle \rightarrow |1\rangle$, i.e., when $q_0q_1q_2$ input $|000\rangle$, q_3 will output $|1\rangle$, and $|0\rangle$ otherwise. Accordingly, if $c(i) + s(i) = 000$ in Figure 4, then q_{16} will output $|1\rangle$, i.e., $|1\rangle_{q_{16}}$. Furthermore, we estimate the probability of $|1\rangle_{q_{16}}$ by quantum amplitude estimation algorithm. Please note that the qubit q_5 in Figure 8 as workspace is corresponding to the qubit q_{16} in Figure 4. In addition, we use 5 qubits as counting space, i.e., $\{q_0, q_1, q_2, q_3, q_4\}$.

Finally, we list estimated values of various instances in Figure 9. Clearly, the average value approximates the true value well by Figure 9. Therefore, our simulation experiments verify the correctness and the feasibility of the proposed quantum PSI-CA protocol.

V. Application

Here, we first give a definition of Privacy-preserving Condition Query.

Definition 2: Privacy-preserving Condition Query (for short PCQ). Suppose that there are two parties, a user (Alice) and a data owner (Bob). Bob owns a private data set with many repeated elements, which can be defined by a two-column table: the first column records different elements of Bob's data set, and the second column shows the number of their respective repetitions accordingly, e.g., a statistics table about students' test scores. Furthermore, Alice wants to privately query the number of the element x in Bob's data set, which matches with a specific query condition, e.g., $x \geq a$, $x = a$, $x < b$ or $a \leq x \leq b$. In addition, PCQ should meet the following private requirements:

Alice's Privacy: The query condition and the query result are both private. That is, Bob cannot learn any private information about the query condition and the query result.

Bob's Privacy: Alice cannot get any private information about Bob's data set except the query result.

In the above definition, please note that Alice's query result is only the number of the elements matching with a specific query condition, not any element of Bob's private data set,

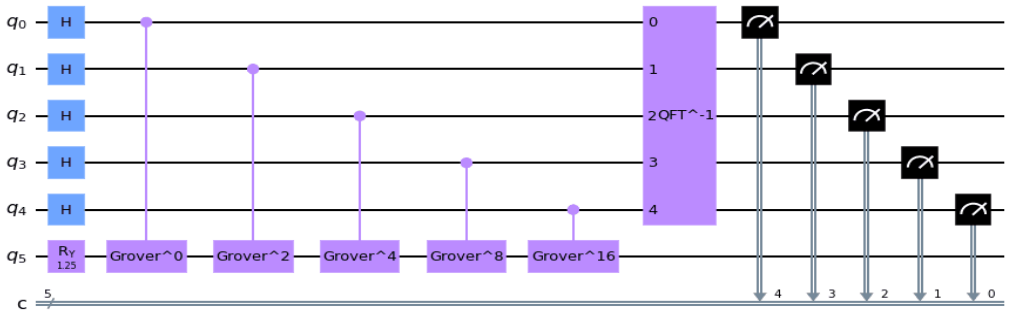


Fig. 8. Quantum circuits of quantum amplitude estimation.

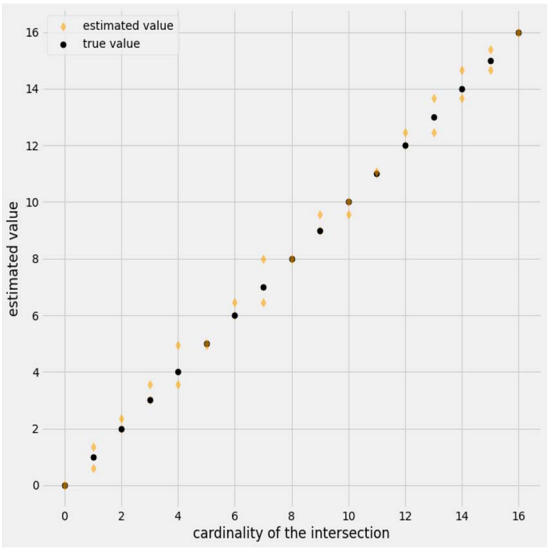


Fig. 9. The comparison results of various simulated instances.

TABLE III
BOB'S PRIVATE DATA SET

The different elements	The number of repetitions
2	2
4	3
5	1
8	1
9	3
12	2
15	1

(d_i)	(0, 0, 2, 0, 3, 1, 0, 0, 1, 3, 0, 0, 2, 0, 0, 1)
$(x_{1,i})$	(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0)
$(x_{2,i})$	(0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$(x_{3,i})$	(0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
$(x_{4,i})$	(0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1)
$(x_{5,i})$	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0)

Fig. 10. How to split Bob's private vector.

and the query condition is diversified, not fixed (i.e., more flexible). The traditional private query does not satisfy PCQ's higher privacy requirements, because the query result of the traditional private query reveals too much information (i.e., at least one element of the data set). In addition, to protect sensitive data from the cloud or the third party's server, the data owner could publish an encrypted version of the original data. Later, the data owner wants to obtain all the data matching with a query condition or within a query region (i.e., private range query), while keeping the query private to the service provider. Similarly, these private queries on encrypted databases cannot yet meet the purpose of PCQ because of revealing too much information.

Furthermore, based on the quantum PSI-CA protocol proposed above, we construct a novel scheme to solve the PCQ problem. Suppose there are two parties, a user (Alice) and a data owner (Bob), where Bob owns a private data set with duplicate elements. Without loss of generality, we assume that all elements of Bob's data set belong to $0, 1, 2, \ldots, N-1\}$.

PCQ Scheme

Step 1: (Encoding). Bob encodes his private data set (i.e., a private two-column table) over Z_N into a private N-component vector: $(d_0, d_1, \ldots, d_{N-1})$, where $d_i = 0$ if i does not belong to Bob's data set (i.e., i is not in the first column of his table), and otherwise d_i is equal to the number of repetitions of i. For example, if Bob's private data set over Z_{16} is listed as Table III, then the encoded 16-component vector is (0, 0, 2, 0, 3, 1, 0, 0, 1, 3, 0, 0, 2, 0, 0, 1).

Step 2: (Splitting). Bob splits his encoded vector into m private vectors by secret splitting ideas as follows: Bob randomly generates m N-component vectors: $(x_{j,0}, x_{j,1}, \ldots, x_{j,N-1})$ for $j = 1, 2, \ldots, m$, such that $d_i = \sum_{j=1}^{m} x_{j,i}$ for each i, where $x_{j,i} \in_R \{0, 1\}$. Please note that $m \geq d_i$ for any i, which is a security parameter determined by Bob. For the above example, how to further split Bob's private vector, please see Figure 10.

Step 3: Alice generates an N-component vector: $(t_0, t_1, \ldots, t_{N-1})$ by her query condition, where $t_i = 1$ if i satisfies the query condition, and $t_i = 0$ otherwise. For example, if Alice wants to query the number of the element x in Bob's private data set above, such that $4 \leq x \leq 9$, then

she will generate the following condition vector,

$$(0,\ 0,\ 0,\ 0,\ \mathbf{1},\ \mathbf{1},\ \mathbf{1},\ \mathbf{1},\ \mathbf{1},\ \mathbf{1},\ 0,\ 0,\ 0,\ 0,\ 0,\ 0). \quad (42)$$

Step 4: Alice asks Bob to run quantum PSI-CA protocol to compute $r_j = \left|T \bigcap X_j\right|$ for $j = 1, 2, \ldots, m$, where Alice serves as the client who inputs her private set, T, and accordingly Bob acts as the server who inputs his private set, X_j. Here the sets, T and X_j, are defined by their respective private vectors as follows:

$$\begin{aligned} T &= \{i | t_i = 1 \wedge i \in Z_N\}, \\ X_j &= \{i | x_{j,i} = 1 \wedge i \in Z_N\}. \end{aligned} \quad (43)$$

Finally, Alice computes $r = \sum_{j=1}^{m} r_j$ as her query result, i.e., the number of the elements satisfying her query condition in Bob's private data set.

A. Correctness

By the encoding methods, we can easily see that r_j is just equal to the number of $x_{j,i} \cdot t_i = 1$ (i.e., satisfying the query condition) for $i = 0, 1, \ldots, N-1$. Furthermore, since $d_i = \sum_{j=1}^{m} x_{j,i}$ for each i, we can get,

$$\begin{aligned} r &= \sum_{i=0}^{N-1} d_i \cdot t_i = \sum_{i=0}^{N-1} \left(\sum_{j=1}^{m} x_{j,i} \cdot t_i\right) \\ &= \sum_{j=1}^{m} \left(\sum_{i=0}^{N-1} x_{j,i} \cdot t_i\right) = \sum\nolimits_{j=1}^{m} r_j. \end{aligned} \quad (44)$$

Therefore, the proposed QCP scheme is correct.

B. Security

The security of the proposed QCP scheme is based on the quantum PSI-CA protocol obviously since two parties exchange messages only by the quantum PSI-CA protocol. And yet, we have proved the security of the quantum PSI-CA protocol. That is, the quantum PSI-CA protocol ensures the security of the proposed QCP scheme.

C. Performance

The main complexity of the proposed QCP scheme is to execute m quantum PSI-CA protocols, while other computations of encoding and secret splitting are lightweight and negligible. According to the previous analysis, the communication complexity of proposed quantum PSI-CA protocol achieves $O(N)$ qubits. Therefore, the communication complexity of proposed QCP scheme is $O(mN)$ qubits, which are mainly used for the costs of transmitting single photons.

VI. Conclusion

In this paper, we presented a novel quantum PSI-CA protocol. Compared with classical related protocols, our proposed quantum PSI-CA protocol obtains higher security and is more suitable for big data applications. Furthermore, we defined a new and flexible private query, i.e., Privacy-preserving Condition Query (QCP). Based on the proposed quantum PSI-CA protocol, we presented an efficient quantum scheme to solve the QCP problem. Finally, we verify the correctness and the feasibility of the proposed quantum protocols by circuit simulations in IBM Qiskit. These proposed protocols have a good application prospect, e.g., in privacy-preserving string statistics and hamming weight calculation.

Our work further shows that quantum cryptography can not only guarantee data security, but also ensure data privacy, and it can also design sophisticated and flexible cryptographic protocols based on quantum mechanics as mathematical cryptography.

References

[1] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Proc. Adv. Cryptol.-Eurocrypt*, in Lecture Notes in Computer Science, vol. 3027. Berlin, Germany: Springer, 2004, pp. 1–19.

[2] R.-H. Shi, Y. Mu, H. Zhong, S. Zhang, and J. Cui, "Quantum private set intersection cardinality and its application to anonymous authentication," *Inf. Sci.*, vols. 370–371, pp. 147–158, Nov. 2016.

[3] M. E. Wu, S. Y. Chang, C. J. Lu, and H. M. Sun, "A communication-efficient private matching scheme in client–server model," *Inf. Sci.*, vol. 275, pp. 348–359, Aug. 2014.

[4] F. Buccafurri, L. Fotia, G. Lax, and V. Saraswat, "Analysis-preserving protection of user privacy against information leakage of social-network likes," *Inf. Sci.*, vol. 328, pp. 340–358, Jan. 2016.

[5] M. Kantarcioglu, R. Nix, and J. Vaidya, "An efficient approximate protocol for privacy-preserving association rule mining," in *Proc. Adv. Knowl. Discovery Data Mining (KDD)*, in Lecture Notes in Computer Science, vol. 5476. Berlin, Germany: Springer, 2009, pp. 515–524.

[6] J. Camenisch and G. M. Zaverucha, "Private intersection of certified sets," in *Proc. Financial Cryptogr. Data Secur. (FC)*, in Lecture Notes in Computer Science, vol. 5628. Berlin, Germany: Springer, 2009, pp. 108–127.

[7] E. D. Cristofaro, P. Gasti, and G. Tsudik, "Fast and private computation of cardinality of set intersection and union," in *Proc. Cryptol. Netw. Secur. (CANC)*, in Lecture Notes in Computer Science, vol. 7712. Berlin, Germany: Springer, 2012, pp. 218–231.

[8] S. K. Debnath and R. Dutta, "Secure and efficient private set intersection cardinality using Bloom filter," in *Proc. Inf. Secur. (ISC)*, in Lecture Notes in Computer Science, vol. 9290. Cham, Switzerland: Springer, 2015, pp. 209–226.

[9] S. Hohenberger and S. Weis, "Honest-verifier private disjointness testing without random oracles," in *Proc. Privacy Enhancing Technol. (PET)*, in Lecture Notes in Computer Science, vol. 4258. Berlin, Germany: Springer, 2006, pp. 277–294.

[10] L. Kissner and D. Song, "Privacy-preserving set operations," in *Proc. Adv. Cryptol.-Crypto*, in Lecture Notes in Computer Science, vol. 3621. Berlin, Germany: Springer, 2005, pp. 241–257.

[11] J. Vaidya and C. Clifton, "Secure set intersection cardinality with application to association rule mining," *J. Comput. Secur.*, vol. 13, no. 4, pp. 593–622, Oct. 2005.

[12] P. Rindal and M. Rosulek, "Malicious-secure private set intersection via dual execution," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1229–1242.

[13] C. Dong and G. Loukides, "Approximating private set union/intersection cardinality with logarithmic complexity," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2792–2806, Nov. 2017.

[14] A. Tajima, H. Sato, and H. Yamana, "Outsourced private set intersection cardinality with fully homomorphic encryption," in *Proc. 6th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, May 2018, pp. 1–8.

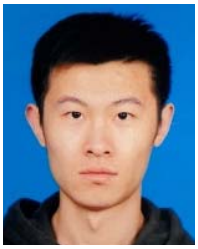
[15] B. Pinkas, T. Schneider, C. Weinert, and U. Wieder, "Efficient circuit-based PSI via Cuckoo hashing," in *Proc. Adv. Cryptol.-Eurocrypt*, in Lecture Notes in Computer Science, vol. 10822. Cham, Switzerland: Springer, 2018, pp. 125–157.

[16] B. Pinkas, T. Schneider, O. Tkachenko, and A. Yanai, "Efficient circuit-based PSI with linear communication," in *Proc. Adv. Cryptol.-Eurocrypt*, in Lecture Notes in Computer Science, vol. 11478. Cham, Switzerland: Springer, 2019, pp. 122–153.

[17] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 212–219.

[18] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Nov. 1994, pp. 124–134.
[19] H.-S. Li, P. Fan, H.-Y. Xia, H. Peng, and S. Song, "Quantum implementation circuits of quantum signal representation and type conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 341–354, Jan. 2019.
[20] K. Cho and T. Miyano, "Chaotic cryptography using augmented Lorenz equations aided by quantum key distribution," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 2, pp. 478–487, Feb. 2015.
[21] R.-H. Shi, "Quantum multiparty privacy set intersection cardinality," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 4, pp. 1203–1207, Apr. 2021.
[22] R.-H. Shi, "Anonymous quantum sealed-bid auction," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 414–418, Feb. 2022.
[23] R.-H. Shi, "Quantum sealed-bid auction without a trusted third party," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 10, pp. 4221–4231, Oct. 2021.
[24] R.-H. Shi, Y. Mu, H. Zhong, J. Cui, and S. Zhang, "An efficient quantum scheme for private set intersection," *Quantum Inf. Process.*, vol. 15, no. 1, pp. 363–371, 2016.
[25] R. H. Shi, "Efficient quantum protocol for private set intersection cardinality," *IEEE Access*, vol. 6, pp. 73102–73109, 2018.
[26] R.-H. Shi, "Quantum private computation of cardinality of set intersection and union," *Eur. Phys. J. D*, vol. 72, p. 221, Dec. 2018.
[27] R.-H. Shi and M. Zhang, "A feasible quantum protocol for private set intersection cardinality," *IEEE Access*, vol. 7, pp. 72105–72112, 2019.
[28] B. Liu, O. Ruan, R. Shi, and M. Zhang, "Quantum private set intersection cardinality based on Bloom filter," *Sci. Rep.*, vol. 11, no. 1, p. 17332, Aug. 2021.
[29] C. Zhang, Y. Long, Z. Sun, Q. Li, and Q. Huang, "Three-party quantum private computation of cardinalities of set intersection and union based on GHZ states," *Sci. Rep.*, vol. 10, no. 1, p. 22246, Dec. 2020.
[30] Y. Wang, P. Hu, and Q. Xu, "Quantum protocols for private set intersection cardinality and union cardinality based on entanglement swapping," *Int. J. Theor. Phys.*, vol. 60, no. 9, pp. 3514–3528, Aug. 2021.
[31] M. Jakobi *et al.*, "Practical private database queries based on a quantum-key-distribution protocol," *Phys. Rev. A, Gen. Phys.*, vol. 83, no. 2, Feb. 2011, Art. no. 022301.
[32] V. Scarani *et al.*, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," *Phys. Rev. Lett.*, vol. 92, no. 5, 2004, Art. no. 057901.
[33] F. Gao, B. Liu, Q.-Y. Wen, and H. Chen, "Flexible quantum private queries based on quantum key distribution," *Opt. Exp.*, vol. 20, no. 16, pp. 17411–17420, 2012.
[34] Y.-G. Yang, S.-J. Sun, P. Xu, and J. Tian, "Flexible protocol for quantum private query based on B92 protocol," *Quantum Inf. Process.*, vol. 13, no. 3, pp. 805–813, 2014.
[35] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Phys. Rev. Lett.*, vol. 100, no. 16, Apr. 2008, Art. no. 160501.
[36] Z. Diao, C. Huang, and K. Wang, "Quantum counting: Algorithm and error distribution," *Acta Applicandae Math.*, vol. 118, no. 1, pp. 147–159, Apr. 2012.
[37] R.-H. Shi, Y. Mu, H. Zhong, and S. Zhang, "Quantum oblivious set-member decision protocol," *Phys. Rev. A, Gen. Phys.*, vol. 92, no. 2, Aug. 2015, Art. no. 022309.
[38] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[39] S. Weimann *et al.*, "Implementation of quantum and classical discrete fractional Fourier transforms," *Nature Commun.*, vol. 7, no. 1, p. 11027, Mar. 2016.
[40] S. Dogra, A. Dorai, and K. Dorai, "Implementation of the quantum Fourier transform on a hybrid qubit–qutrit NMR quantum emulator," *Int. J. Quantum Inf.*, vol. 13, Mar. 2015, Art. no. 1550059.
[41] M. Smania, A. M. Elhassan, A. Tavakoli, and M. Bourennane, "Experimental quantum multiparty communication protocols," *npj Quantum Inf.*, vol. 2, no. 1, p. 16010, Jun. 2016.
[42] S. Weimann *et al.*, "Implementation of quantum and classical discrete fractional Fourier transforms," *Nature Commun.*, vol. 7, no. 1, p. 11027, Mar. 2016.

Run-Hua Shi received the Ph.D. degree in information security from the University of Science and Technology of China in 2011. He is currently a Professor with North China Electric Power University. His current research interests include classical/quantum cryptographic algorithms/protocols and their applications.



Yi-Fei Li received the bachelor's degree in information security from North China Electric Power University in 2019, where he is currently pursuing the master's degree. His main works are quantum computing and quantum circuits.