Further Investigation on Classical Multiparty computation using Quantum Resources

Hao Cao,1,2 Wenping Ma,1 Ge Liu,1 and Liangdong Lyu1,3 ∗

1State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an, China 710071
2School of Information and Network Engineering, Anhui Science and Technology University, Chuzhou, China 233100
3Department of Basic Sciences, Air Force Engineering University, Xi’an, 710071, China

(Dated: December 18, 2018)

The tremendous development of cloud computing and network technology makes it possible for multiple people with limited resources to complete a large-scale computing with the help of cloud servers. In order to protect the privacy of clients, secure multiparty computation (SMPC) plays an important role in the process of computing. Recently, Clementi et al [Phys. Rev. A 96, 062317(2017)] proposed a secure multiparty computation protocol using quantum resources. In their protocol, utilizing only linear classical computing and limited manipulation of quantum information, a method of computing \( n - \text{variable} \) symmetric Boolean function \( f(x_1,x_2,\cdots,x_n) \) with degree 2 is proposed, and all clients can jointly compute \( f(x_1,x_2,\cdots,x_n) \) without revealing their private inputs with the help of a sever. They proposed an open problem: are there more simple nonlinear functions like the one presented by them that can be used as subroutines for larger computation protocols? We will give the answer to this question in this paper. Inspired by Clementi et al’s work, we continue to explore the quantum realization of Boolean functions. First, we demonstrate a way to compute a class of \( n - \text{variable} \) symmetric Boolean function \( f_n^k \) by using single-particle quantum state \( |0\rangle \) and single-particle unitary operations \( U_k \). Second, we show that each \( n - \text{variable} \) symmetric Boolean function can be represented by the linear combination of \( f_n^k (k = 0, 1, \cdots, n) \) and each function \( f_n^k (2 \leq k \leq n) \) can be used to perform secure multiparty computation. Third, we propose an universal quantum implementation method for arbitrary \( n - \text{variable} \) symmetric Boolean function \( f(x_1,x_2,\cdots,x_n) \). Finally, we demonstrate our secure multiparty computation protocol on IBM quantum cloud platform.

I. INTRODUCTION

In the cloud environment, it is very common for a number of clients with limited resource to delegate the server to compute a function of their inputs. If each client would rather not reveal his input information to server and other clients, a special cryptographic model called Secure multiparty computation (SMPC) will be considered for use. The SMPC problem is originated from the Yao’s millionaire problem[1], and many classical solutions[1–3] to it have been proposed.

In 1984, a quantum key distribution protocol known as BB84[4], which is completely different from classical cryptography, came into people’s vision and attracted wide attention. Since then, various types of quantum cryptographic protocols, such as quantum secret sharing(QSS)[5, 6], quantum secure direct communication(QSDC)[7], quantum key agreement(QKA)[8], quantum privacy comparison(QPC)[9], and so on, have been proposed. Especially, quantum solutions to SMPC problem, i.e., quantum SMPC(QSMPC) [10–14], attracts much attention because of its widely application in electronic voting, online auction, and multiparty data processing.

Recently, Clementi et al[14] proposed a QSMPC protocol in which a number of clients can collaborate to compute a function without revealing their inputs. The function in their protocol is a \( n - \text{variable} \) symmetric Boolean function with degree 2. They proposed an open problem: are there more simple nonlinear functions like the one presented by them that can be used as subroutines for larger computation protocols? We will give the answer to this question in this paper. Inspired by Clementi et al’s work, we focus on the case of arbitrary symmetric Boolean functions. First, we give a quantum implementation of a class of symmetric Boolean function \( f_n^k \). Second, we show that each \( n - \text{variable} \) symmetric Boolean function can be represented by linear combination of \( f_n^k (k = 0, 1, \cdots, n) \) and each function \( f_n^k (2 \leq k \leq n) \) can be used to perform secure multiparty computation. Third, we explore an universal quantum implementation method for arbitrary \( n - \text{variable} \) symmetric Boolean function \( f(x_1,x_2,\cdots,x_n) \).

The remainder of our work is organized as follows. Section II introduces a method to compute a class of \( n - \text{variable} \) symmetric Boolean function \( f_n^k \) by using single-particle quantum state \( |0\rangle \) and single-particle unitary operations \( U_k \). Besides, the quantum implementations of arbitrary symmetric Boolean functions are explored in this section. In section III, we give the description of our QSMPC protocol, i.e., each function \( f_n^k (2 \leq k \leq n) \) can be used to perform secure multiparty computation. Section IV analyzes the security of our protocol. In section V, a simulation of our protocol on IBM quantum cloud platform. At last, a conclusion of this paper is given in section VI.

II. THEORY

Our work is a further investigation of Clementi et al’s work[14]. In their paper, they focus on computing the \( n - \text{variable} \) Boolean function \( f \) by using the following equations:

\[
(U^1)^\otimes x_1 U^2 x_2 \cdots U^n x_n |0\rangle = |f\rangle,
\]

(1)

\[ \text{ccaho2000854@163.com} \]
\[ \text{wpma@mail.xidian.edu.cn} \]
\[ \text{446552402@qq.com} \]
\[ \text{kelinglv@163.com} \]
where $U = R_y(\frac{\pi}{2}) = \cos \frac{\pi}{2} I - \sin \frac{\pi}{2} Y$ be the $\frac{\pi}{2}$ rotation around the $y$ axis of the Bloch sphere, $V = R_y(\pi) = \cos \frac{\pi}{2} I - \sin \frac{\pi}{2} Y$ be the $\pi$ rotation around the $y$ axis, $f$ be the $2 - \text{degree}$ $n - \text{variable}$ symmetric Boolean function $f_2^n \equiv \sum_{j=1}^{n-1} [x_j + 1 \times (\hat{\bigwedge}_{i=1}^{j} x_i)] = \sum_{1 \leq i < j \leq n} x_i x_j$, $i \in \{0, 1\}$ $(i = 1, 2, \ldots, n)$ and $\tilde{r} = \oplus_{i=1}^{n} r_i$.

Define $U_k = R_y(\frac{\pi}{2k})$ $(k = 1, 2, \ldots, n)$ be the $\frac{\pi}{2k}$ rotation around the $y$ axis of the Bloch sphere, and $U_0 = I = R_y(0)$ be the identity operation. Then we have the following important arguments about $U_k$ $(k = 1, 2, \ldots, n)$.

**Theorem 1** Let $k$ and $h$ be two nonnegative integers, then the following equation holds:

$$U_k U_h = U_h U_k \quad U_k^* U_h = U_h^* U_k^*.$$  \hspace{1cm} (3)

**Proof.** First, $U_k U_h = R_y(\frac{\pi}{2k}) R_y(\frac{\pi}{2h}) = R_y(\frac{\pi}{2} + \frac{\pi}{2h}) = R_y(\frac{\pi}{2} + \frac{\pi}{2k}) = U_h U_k$.

Next, $U_h \approx IU_k (U_k^* U_h) U_h \Rightarrow U_h = U_k^* U_h U_k \Rightarrow U_h U_k^* = U_k^* U_h$. 

**Theorem 2** Let $k$ be a nonnegative integer, $x_i \in \{0, 1\}$ $(i = 1, 2, \ldots, n)$.

(1) The single-particle quantum state in the form of $(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle$ can be regarded as the quantum implementation of a $n - \text{variable}$ Boolean function $f_n^k$, i.e.,

$$(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle = |f_n^k\rangle.$$  \hspace{1cm} (4)

(2) $f_n^k$ is a symmetric Boolean function.

(3) $\text{deg}(f_n^k) \geq k$.

(4) The algebraic normal form of $f_n^k$ contains all monomials with degree $k$.

(5) Each $n - \text{variable}$ symmetric Boolean function can be represented by the linear combination of $f_n^k (k = 0, 1, \ldots, n)$.

**Proof.** (1) In order to prove the proposition (1), we only need to show that the quantum state in the form of $(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle$ is either $|0\rangle$ or $|1\rangle$. Let $wt(x_1, x_2, \ldots, x_n) = \sum_i x_i = ak + b$, where $wt(x_1, x_2, \ldots, x_n)$ be the weight of $(x_1, x_2, \ldots, x_n)$ (i.e., the number of ones in $x_1, x_2, \ldots, x_n$), $a$ and $b$ be two nonnegative integers, and $0 \leq b < k$, then

$$(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle = (U_k^\dagger x_i) \mod k U_k^\dagger x_i |0\rangle = (U_k^\dagger x_i)^k |0\rangle = [R_y(\frac{\pi}{2})]^k |0\rangle = R_y(a\pi) |0\rangle \in \{0\}, \{1\},$$

Hence, $(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle$ can be regarded as the quantum implementation of a $n - \text{variable}$ Boolean function $f_n^k$.

(2) From equation(3), we can easily get that $f_n^k = 1$ if and only if $R_y(a\pi) |0\rangle = |1\rangle$, if and only if $a$ be odd. Let $a'$ be a nonnegative integer, we have

$$f_n^k = \begin{cases} 0 & \text{wt}(x_1, x_2, \ldots, x_n) = 2a'k + b \\ 1 & \text{wt}(x_1, x_2, \ldots, x_n) = (2a' + 1)k + b. \end{cases}$$

which implies that $f_n^k$ is a symmetric Boolean function.

(3) It is easy to verify the fact that $f_n^k (x_1, x_2, \ldots, x_n) = 0$ for each vector $(x_1, x_2, \ldots, x_n)$ with $wt(x_1, x_2, \ldots, x_n) < k$, which implies $\text{deg}(f_n^k) \geq k$.

(4) Owing to the fact that $f_n^k (x_1, x_2, \ldots, x_n) = 1$ for each vector $(x_1, x_2, \ldots, x_n)$ with $wt(x_1, x_2, \ldots, x_n) = k$, we get that the algebraic normal form of $f_n^k$ contains all monomials with degree $k$.

(5) From (3) and (4), we can draw that $f_0^k, f_1^k, f_2^k, \ldots,$ and $f_n^k$ are linearly independent, and they form a basis from the $n - \text{variable}$ symmetric Boolean functions. Hence, each $n - \text{variable}$ symmetric Boolean function can be represented by the linear combination of $f_n^k (k = 0, 1, \ldots, n)$.

From Theorem 1 and Theorem 2, we can easily get the following equation which is the generalization of equation(2).

$$(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle = |\tilde{r} \oplus f_n^k\rangle,$$  \hspace{1cm} (6)

where $r_i \in \{0, 1\}$ $(i = 1, 2, \ldots, n)$ and $\tilde{r} = \oplus_{i=1}^{n} r_i$.

For convenience of description, we denote the vector $(x_1, x_2, \ldots, x_1)$ as $x$, the vector $(r_1, r_2, \ldots, r_1)$ as $r$, the unitary operation $(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i = U(n, k, x)$, and the unitary operation $(U_k^\dagger \sum x_i) \mod k V^r U_k^\dagger \cdots V^r U_k^\dagger U_k x_i = VU(n, k, x, r)$ separately, then equation(4) and equation(5) can be rewritten as follows:

$$U(n, k, x) |0\rangle = |f_n^k\rangle$$  \hspace{1cm} (7)

$$VU(n, k, x, r) |0\rangle = |\tilde{r} \oplus f_n^k\rangle$$  \hspace{1cm} (8)

**Theorem 3** Let $k$ and $h$ be two nonnegative integers, then

$$|f_n^k \oplus f_0^h| = \tilde{U}(n, k, x) U(n, h, x) |0\rangle$$  \hspace{1cm} (9)

**Proof.** First, let $r_n = r_{n-1} = \cdots = r_2 = 0$ and $r_1 = 1$ in equation (6), we get

$$(U_k^\dagger \sum x_i) \mod k U_k^\dagger \cdots U_k^\dagger U_k x_i |0\rangle = |f_n^k \oplus 1\rangle$$  \hspace{1cm} (10)

i.e.,

$$\tilde{U}(n, k, x) |1\rangle = |f_n^k \oplus 1\rangle$$  \hspace{1cm} (11)
Next, consider the right hand of the equation (9).

\[
\overline{U}(n, k, x)\overline{U}(n, h, x)|0\rangle = \overline{U}(n, k, x)|f_n^h\rangle
\]

\[
= \begin{cases} 
|f_n^h\rangle & \text{if } f_n^h = 0 \\
|f_n^h \oplus 1\rangle & \text{if } f_n^h = 1 
\end{cases} \quad (\text{From equation (11)}) \tag{12}
\]

From Theorem 2(5) and Theorem 3, we can easily get the quantum implementation of an arbitrary symmetric Boolean function.

**Theorem 4** Let \( f(x_1, x_2, \cdots, x_n) \) be a \( n \)-variable symmetric Boolean function which can be presented as

\[
f = \oplus_{k=0}^{n-1} a_k f_n^k \tag{13}
\]

then

\[
|f\rangle = \prod_{k=0}^{n-1} (\overline{U}(n, k, x)^{a_k}|0\rangle \tag{14}
\]

### III. QUANTUM SECURE MULTIPARTY COMPUTATION PROTOCOL

In this section, we will show the QSMPC protocol by using equation (5) or equation (9) in the ideal case environment. Let \( C_i \) (\( i = 1, 2, \cdots, n \)) be \( n \) clients and each client \( C_i \) possesses a private input \( x_i \) and selects a random bit \( r_i \). They want to jointly compute the function \( f_n^k(2 \leq k \leq n) \) without revealing their inputs with the help of a server \( S \).

**Step 1** First, each client \( C_i \) divides his private input \( x_i \) into \( n \) elements \( x_{i,1}, x_{i,2}, \cdots \), and \( x_{i,n} (x_{i,j} \in \{0, 1 \}, j = 1, 2, \cdots, n) \), and the random bit \( r_i \) into \( n \) elements \( r_{i,1}, r_{i,2}, \cdots, \) and \( r_{i,n} (r_{i,j} \in \{0, 1 \}, j = 1, 2, \cdots, n) \), such that \( \sum_{j=1}^{n} x_{i,j} \equiv x_i \mod k \) and \( \oplus_{j=1}^{n} r_{i,j} = r_i \). Second, each client \( C_i \) sends \( x_{i,j} \) and \( r_{i,j} \) to client \( C_j \). Third, each client \( C_i \) computes \( \tilde{x}_i = \sum_{j=1}^{n} x_{i,j} \mod k \) and \( \tilde{r}_i = \oplus_{j=1}^{n} r_{i,j} \).

**Step 2** First, the server \( S \) prepares a single particle in the state \(|0\rangle \) and sends it to the client \( C_1 \). Second, \( C_1 \) performs the unitary operation \( V^r U_x \) on the received single particle according to his private input \( x_1 \) and random bit \( r_1 \), and sends the resulted single particle to the client \( C_2 \) who will perform the unitary operation \( V^{r_2} U_x \) on the received particle according to his private input \( x_2 \) and random bit \( r_2 \). This procession continues until all the clients have applied their unitary operations to the single particle. At this point, the single particle is in the hands of \( C_n \).

**Step 3** First, each client \( C_i \) sends \( \tilde{x}_i \) to the client \( C_n \) through an secure authentication channel. Second, \( C_n \) calculates \( (\sum_{i=1}^{n} \tilde{x}_i) \mod k \). Third, \( C_n \) perform the unitary operation \( (U^{\dagger}) (\sum_{i=1}^{n} \tilde{x}_i) \mod k \) on the single particle, and the resulted particle will be in the state \(|f_n^k \oplus \tilde{r}\rangle\) owing to the fact that \( (\sum_{i=1}^{n} \tilde{x}_i) \mod k = (\sum_{i=1}^{n} x_i) \mod k \). Fourth, \( C_n \) sends the resulted particle back to the server \( S \). At last, the server \( S \) will get the state \(|f_n^k \oplus \tilde{r}\rangle\) by measuring the received particle, and announce \( f_n^k \oplus \tilde{r} \) to all clients.

**Step 4** First, each client \( C_i \) transmits the classical bit \( \tilde{r}_i = \oplus_{j=1}^{n} r_{j,i} \) to all other clients through an secure authentication channel. Second, each client \( C_i \) calculates \( \oplus_{j=1}^{n} r_{j,i} = \oplus_{j=1}^{n} \tilde{r}_j = \tilde{r} \). At last, every client will extract the value of \( f_n^k \) by performing the XOR operation \( f_n^k = (f_n^k \oplus \tilde{r}) \oplus \tilde{r} \).

### IV. SECURITY ANALYSIS AND EFFICIENCY COMPARISON OF THE PROPOSED QSMPC PROTOCOL

**A. Security Analysis**

In this section, we only focus on the internal attack from the clients or the server because internal attacks are usually more effective than external attacks. We assume that both the clients and the server will execute the QSMPC protocol. However, the server \( S \) will try to get the private inputs of the clients or the function output \( f_n^k \), and each client will also try to get the private inputs of other clients.

(1) Consider the security against the attack from the server \( S \). First, if the server \( S \) wants to extract the private input of some one client, say \( C_1 \), he must intercept the particle sent from \( C_1 \) to \( C_2 \), and measures it in the correct measurement basis \( \{|0\rangle, |1\rangle\} \) or \( \{|U_k|, |U_k\rangle\} \). However, he could not choose the right measurement base because he knows nothing about the unitary operation \( V^{r_1} U_x \) and the state information of the single particle. Second, if the server \( S \) wants to extract the function output \( f_n^k \). However, he will also fail because he knows nothing about the random classical bits \( r_i(i = 1, 2, \cdots, n) \), \( \tilde{r}_i(i = 1, 2, \cdots, n) \) and \( \tilde{r} \).

(2) Consider the security against the attack from the clients. Let us discuss a very unfavorable situation, i.e., the dishonest clients consist of \( C_n \) who will get the information of \( (\sum_{i=1}^{n} \tilde{x}_i) \mod k \), and some other \( n - t - 1 \) clients \( C_{t+1}, \cdots, C_{n-1} \). They will collaborate to extracts the private inputs of \( C_1, \cdots, C_t \). Apparently, they could easily access the \( (\sum_{i=1}^{n} \tilde{x}_i) \mod k \) from \( (\sum_{i=1}^{n} \tilde{x}_i) \mod k \) and \( x_{t+1}, \cdots, x_{n-1}, x_n \). If \( t = 1 \) (i.e., only one client is honest), they could extract the private input \( x_1 \); or else, they cannot get any useful information about the private inputs of the honest clients.

**B. Efficiency Comparison**

In classical case, secure multi-party computation is usually achieved by garbled circuits[13]. Similarly, quantum circuits are usually used to perform quantum secure multi-party computation and a quantum circuit consists of a number of quantum gates. The efficiency evaluation usually includes three indicators: the quantum resource, the size and the number of quantum gates, and classical communication cost involved in the quantum circuit.

Several quantum implementations of classical NAND gate, which are based on either entangled GHZ state or single qubit,
are presented in the previous work[12]. In the process of performing NAND gate based on single qubit, one qubit, two single-particle unitary operations and several rounds of classical communication are involved. In the process of performing NAND gate based on single entangled GHZ state, three qubit, Four single-particle unitary operations, and several rounds of classical communication are involved. Besides, their protocols guarantees no security for the inputs of the parties as stated in Clementi et al’s work[14]. Quantum implementation of Boolean function and its application in QSMPC are explored in Clementi et al’s work[14]. In their protocol, one qubit, $n + 1$ single-particle unitary operations and one round classical communication are needed if there are $n$ clients are involved. Hence, their protocol is more efficient.

Our QSMPC protocol is a generalization of the work by Clementi et al’s protocol and ours, we can know that the efficiency of the two protocols is the same.

V. QUANTUM SIMULATION ON IBM QUANTUM CLOUD PLATFORM

We simulates our protocol on IBM quantum cloud platform. Here, we will design the quantum circuits to fit the ibmqx4 quantum computer. Suppose the preparation and measurement of the qubit are operated by the server(i.e., q0, q1, q[2], q[3] and q[4]). The private inputs of $n$ clients and the random classical bits selected by them can be represented as $x = (x_1, x_2, \ldots, x_n)$ and $r = (r_1, r_2, \ldots, r_n)$. Fig.1 shows the quantum circuits of our protocol with $n = 5, n = 8$, and $n = 10$. The definition of the operation $VU(n, k, x, r)$ can be seen in equation (9). The unitary operation $U_k$ is realized by the quantum gate $U_3(\frac{\pi}{2}, 0, 0)$, the unitary operation $V$ is realized by the quantum gate $U_3(\pi, 0, 0)$ and the unitary operation $I$ is realized by the quantum gate $Ie$ instead of $U_3(0, 0, 0)$ in the IBM quantum cloud platform.

Case 1: $n = 5$. let $C_1$, $C_2$, $\ldots$, and $C_5$ be the five clients. They will collaborate to compute the function $f_5^1$, $f_5^2$, and $f_5^3$ with $x = (x_1, x_2, \ldots, x_5) = (1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$. In the process of performing the NAND gate based on single qubit, one qubit, two single-particle unitary operations and several rounds of classical communication are involved. In the process of performing the NAND gate based on single entangled GHZ state, three qubit, Four single-particle unitary operations, and several rounds of classical communication are involved. Besides, their protocols guarantee no security for the inputs of the parties as stated in Clementi et al’s work[14]. Quantum implementation of Boolean function and its application in QSMPC are explored in Clementi et al’s work[14]. In their protocol, one qubit, $n + 1$ single-particle unitary operations and one round classical communication are needed if there are $n$ clients are involved. Hence, their protocol is more efficient. Our QSMPC protocol is a generalization of the work by Clementi et al’s protocol and ours, we can know that the efficiency of the two protocols is the same.

Case 1: $n = 5$. let $C_1$, $C_2$, $\ldots$, and $C_5$ be the five clients. They will collaborate to compute the function $f_5^1$, $f_5^2$, and $f_5^3$ with $x = (x_1, x_2, \ldots, x_5) = (1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$. In the process of performing the NAND gate based on single qubit, one qubit, two single-particle unitary operations and several rounds of classical communication are involved. In the process of performing the NAND gate based on single entangled GHZ state, three qubit, Four single-particle unitary operations, and several rounds of classical communication are involved. Besides, their protocols guarantee no security for the inputs of the parties as stated in Clementi et al’s work[14]. Quantum implementation of Boolean function and its application in QSMPC are explored in Clementi et al’s work[14]. In their protocol, one qubit, $n + 1$ single-particle unitary operations and one round classical communication are needed if there are $n$ clients are involved. Hence, their protocol is more efficient. Our QSMPC protocol is a generalization of the work by Clementi et al’s protocol and ours, we can know that the efficiency of the two protocols is the same.

Case 2: $n = 8$. let $C_1$, $C_2$, $\ldots$, and $C_8$ be the eight clients. They will collaborate to compute the function $f_8^1$, $f_8^2$, and $f_8^3$ with $x = (x_1, x_2, \ldots, x_8) = (1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$. In the process of performing the NAND gate based on single qubit, one qubit, two single-particle unitary operations and several rounds of classical communication are involved. In the process of performing the NAND gate based on single entangled GHZ state, three qubit, Four single-particle unitary operations, and several rounds of classical communication are involved. Besides, their protocols guarantee no security for the inputs of the parties as stated in Clementi et al’s work[14]. Quantum implementation of Boolean function and its application in QSMPC are explored in Clementi et al’s work[14]. In their protocol, one qubit, $n + 1$ single-particle unitary operations and one round classical communication are needed if there are $n$ clients are involved. Hence, their protocol is more efficient. Our QSMPC protocol is a generalization of the work by Clementi et al’s protocol and ours, we can know that the efficiency of the two protocols is the same.

Case 2: $n = 8$. let $C_1$, $C_2$, $\ldots$, and $C_8$ be the eight clients. They will collaborate to compute the function $f_8^1$, $f_8^2$, and $f_8^3$ with $x = (x_1, x_2, \ldots, x_8) = (1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$. In the process of performing the NAND gate based on single qubit, one qubit, two single-particle unitary operations and several rounds of classical communication are involved. In the process of performing the NAND gate based on single entangled GHZ state, three qubit, Four single-particle unitary operations, and several rounds of classical communication are involved. Besides, their protocols guarantee no security for the inputs of the parties as stated in Clementi et al’s work[14]. Quantum implementation of Boolean function and its application in QSMPC are explored in Clementi et al’s work[14]. In their protocol, one qubit, $n + 1$ single-particle unitary operations and one round classical communication are needed if there are $n$ clients are involved. Hence, their protocol is more efficient. Our QSMPC protocol is a generalization of the work by Clementi et al’s protocol and ours, we can know that the efficiency of the two protocols is the same.
The quantum implementation of $f^k_8(k = 2, 3, 4, 6, 6)$ can be seen in Fig.1(b). Here, $VU(8, 2, x, r) = U^1_2 II IU_2 VI IU_2 II U^1_2 VI U^1_2 VI$ $IU_3 VI U^1_3 IU_3 VI IU_3$ $VI VU_3, VU(8, 4, x, r) = U^1_4 II IU_4 VI IU_4 IV$ $IU_4 VI VU_4, VU(8, 5, x, r) = (U^1_5)^0 II IU_5 VI IU_5 IV$ $IU_5 VI VU_5, and VU(8, 6, x, r) = (U^1_6)^0 II IU_6 VI IU_6 IV$ $IU_6 VI VU_6$. The measurement results $(f^k_8 \oplus \overline{r}, f^k_8 \oplus \overline{r}, f^k_8 \oplus \overline{r}, f^k_8 \oplus \overline{r}) = (f^k_8, f^k_8, f^k_8, f^k_8)$ can be seen in Fig.3, and the probabilities of output $f^k_8(k = 2, 3, 4, 5, 6)$ correctly can be seen in Tab.II.

Case 3: $n = 10$. Let $C_1, C_2, \ldots, C_{10}$ be the eight clients. They will collaborate to compute the function $f^2_{10}, f^3_{10}, f^1_{10}, f^0_{10}$, and $f^0_{10}$ with $x = (x_1, x_2, \ldots, x_{10}) = (1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$, $r = (r_1, r_2, \ldots, r_{10}) = (1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$, $(\sum_{0}^{10} x_i) \mod 2 = 1$, $(\sum_{0}^{10} x_i) \mod 3 = 2$, $(\sum_{0}^{10} x_i) \mod 4 = 1$, $(\sum_{0}^{10} x_i) \mod 5 = 0$, $(\sum_{0}^{10} x_i) \mod 6 = 5$ and $\overline{r} = \oplus_{0}^{10} r_i, r_i = 1$. The quantum implementation of $f^k_{10}(k = 2, 3, 4, 5, 6)$ can be seen in Fig.1(c). Here, $VU(10, 2, x, r) = U^1_2 II IU_2 UI IU_2$ $VI VU_2 II VU_2, VU(10, 3, x, r) = (U^1_4)^0 II IU_4 VI IU_4 IV$ $IU_4 VI VU_4, VU(10, 4, x, r) = (U^1_5)^0 II IU_5 VI IU_5 IV$ $IU_5 VI VU_5, VU(10, 5, x, r) = (U^1_6)^0 II IU_6 VI IU_6 IV$ $IU_6 VI VU_6, and VU(10, 6, x, r) = (U^1_7)^0 II IU_7 VI IU_7 IV$ $IU_7 VI VU_7. The measurement results $(f^0_{10} \oplus \overline{r}, f^0_{10} \oplus \overline{r}, f^0_{10} \oplus \overline{r}, f^0_{10} \oplus \overline{r}) = (f^0_{10}, f^0_{10}, f^0_{10}, f^0_{10})$ can be seen in Fig.4, and the probabilities of output $f^k_{10}(k = 2, 3, 4, 5, 6)$ correctly can be seen in Tab.III.

### Tab.II. Statistics of output $f^k_8$ correctly

| Boolean function | Correct value times | Probability |
|------------------|---------------------|-------------|
| $f^k_8$          | 0                   | 969         |
|                  | 1                   | 855         |
|                  | 1                   | 869         |
|                  | 1                   | 936         |
|                  | 0                   | 913         |

### Tab.III. Statistics of output $f^k_{10}$ correctly

| Boolean function | Correct value times | Probability |
|------------------|---------------------|-------------|
| $f^0_{10}$       | 0                   | 880         |
| $f^1_{10}$       | 1                   | 923         |
| $f^2_{10}$       | 1                   | 897         |
| $f^3_{10}$       | 1                   | 843         |

**VI. CONCLUSION**

In this paper, we solve an open problem proposed by Clementi et al. Inspired by Clementi et al’s work, we explore the quantum realization of symmetric Boolean functions and demonstrate that a class of $n - variable$ symmetric Boolean functions $f^k_n$ can be implemented by quantum circuits. Besides, each function $f^k_n(2 \leq k \leq n)$ can be used to...
perform secure multiparty computation and each \( n \)-variable symmetric Boolean function can be represented by the linear combination of \( f^*_k(n) \) for \( k = 0, 1, \cdots, n \). Also, we propose an universal quantum implementation method for arbitrary \( n \)-variable symmetric Boolean function \( f(x_1, x_2, \cdots, x_n) \). At last, we demonstrate our secure multiparty computation protocol on IBM quantum cloud platform.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0802400, the National Science Foundation of China under Grant 61373171, 61702007 and 11801564, the 111 Project under Grant B08038, and the Key Project of Science Research of Anhui Province under Grant KJ2017A519, the National Science Foundation of Shaanxi Province under Grant No.2017JQ1032, the Basic Research Project of Natural Science of Shaanxi Province under Grant 2017JM6037.

[1] A. C. Yao, in proceedings of the 23rd Annual Symposium on Foundations of Computer Science (IEEE Computer Society, Washington, DC, 1982), pp.160-164.
[2] A. C. Yao, in proceedings of the 27rd Annual Symposium on Foundations of Computer Science (IEEE Computer Society, Toronto, 1986), pp.162-167.
[3] O. Goldreich, S. Micali, and A. Wigderson, in Annual ACM Symposium on Theory of Computing (ACM, New York, 1987), pp.218.
[4] C. H. Bennett and G. Brassard, in proceedings of International Conference on Computer System and Signal Processing (Bangalore, India, 1984), pp.175-179.
[5] M. Hillery, V. Buzek and A. Berthiaume, Phys. Rev. A 59, 1829(1999).
[6] H. Cao and W. Ma, IEEE. Photonics. J 9, 7600207(2017).
[7] W. Zhang, D. S. Ding, Y. B. Sheng, et al, Phys. Rev. Lett 118, 220501(2017).
[8] H. Cao and W. Ma, Laser. Phys. Lett 15, 095201(2018).
[9] H. Y. Tseng, J. Lin and T. Hwang, Quantum. Inf. Process 11, 373(2012).
[10] J. Anders and D. E. Browne, Phys. Rev. Lett 102, 050502(2009).
[11] K. Loukopoulos and D. E. Browne, Phys. Rev. A 81, 062336(2010).
[12] V. Dunjko, T. Kapourniotis, and E. Kashefi, J. Quantum. Inform. Comput 16, 0061 (2016).
[13] S. Barz, V. Dunjko, and F. Schlederer, et al, Phys. Rev. A 93, 032339 (2016).
[14] M. Clementi, A. Pappa, and A. Eckstein, et al, Phys. Rev. A 96, 062317(2017).
[15] Y. Huang, D. Evans, and J. Katz, et al, in Proceedings of the 20th USENIX Security Symposium, (USENIX, San Francisco, CA, 2011), pp. 331-335.