CHIN. PHYS. LETT. Vol. 37, No. 5 (2020) 050303

Quantum Secure Multiparty Computation with Symmetric Boolean Functions *

Hao Cao(曹浩)[1,2], Wenping Ma(马文平)[3**], Ge Liu(刘鸽)[3],
Liangdong Lü(吕良东)[3,4], Zheng-Yuan Xue(薛正远)[5,6**]

[1]*Anhui Province Key Laboratory of Animal Nutritional Regulation and Health, School of Information and Network Engineering, Anhui Science and Technology University, Fengyang 233100*
[2]*School of Mathematical Science, Huaibei Normal University, Huaibei 235000*
[3]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071*
[4]*Department of Basic Sciences, Air Force Engineering University, Xi'an 710071*
[5]*Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, and School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006*
[6]*Frontier Research Institute for Physics, South China Normal University, Guangzhou 510006*

(Received 28 February 2020)

We propose a class of n-variable Boolean functions which can be used to implement quantum secure multiparty computation. We also give an implementation of a special quantum secure multiparty computation protocol. An advantage of our protocol is that only 1 qubit is needed to compute the n-tuple pairwise AND *function, which is more efficient comparing with previous protocols. We demonstrate our protocol on the IBM quantum cloud platform, with a probability of correct output as high as 94.63%. Therefore, our protocol presents a promising generalization in realization of various secure multipartite quantum tasks.*

PACS: 03.67.Ac, 03.67.Dd, 42.50.Dv DOI: 10.1088/0256-307X/37/5/050303

Currently, the tremendous development of cloud computing and network technology makes it possible for multiple people with limited resources to complete complex computing tasks with the help of a cloud server. In the cloud computing scenario, to protect the privacy of all the involved clients, a special cryptographic model for secure multiparty computation (SMPC) can be used. The SMPC problem originates from Yao's millionaire problem,[1] and many classical solutions[1−3] have been proposed so far. Recently, quantum solutions to the SMPC problem (i.e., quantum secure multiparty computation (QSMPC) protocols)[4−17] have attracted a great deal of attention. Unfortunately, due to the restriction of no-go theorem,[16,17] most QSMPC protocols are insecure unless the participants are restricted to individual measurements.

The QSMPC protocol states that some clients jointly compute a nonlinear Boolean function without revealing their private inputs. Recently, a QSMPC protocol has been proposed[8] based on a particular 2-degree n-variable symmetric Boolean function (SBF) $f = \oplus_{1\le i<j\le n} x_i x_j$ $(i, j \in \{1, 2, \ldots, n\})$. However, since only one function is available, function special attacks (i.e., some potential special attack on a specific function) may be developed. Therefore, more functions that can be used as subroutines for larger computation protocols are highly desired. Hence, these authors 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 give an affirmative solution and find a class of such functions, leading to the choice of the functions for QSMPC to be arbitrary.

Here, we define a class of n-variable SBFs $f_n^k (k = 1, 2, \ldots, n)$ as follows: $f_n^k(x_1, x_2, \ldots, x_n) = 1$ if and only if $wt(x_1, x_2, \ldots, x_n) = ak + b (0 \le b < k)$ with a odd, where $wt(x_1, x_2, \ldots, x_n) = \sum_i x_i$ be the weight of the input vector $(x_1, x_2, \ldots, x_n)$. We will show that the class of SBFs, f_n^k $(2 \le k \le n)$, can be used to accomplish the QSMPC task. In our protocol, the choice of SBFs can be arbitrary, and thus our scheme may be potentially more secure against function special attacks. Meanwhile, we present quantum implementation of all $f_n^l (l = 0, 1, \ldots, n)$ functions, which form a complete basis for n-variable SBFs, and thus an arbitrary SBF can be implemented. In addition, we only need 1 qubit to compute the n-tuple pairwise AND function, which is more efficient compared with previous schemes.

Our scheme is based on SBFs. A Boolean function $f(x)$ will be symmetric if $f(x) = f(y)$ for any $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^n$ with $wt(x) = wt(y)$, that is, the outputs of $f(x)$ only depend on the weight of the input vectors. Hence, an SBF $f(x)$ can be seen as an $(n+1)$-dimensional vector $(f_0, f_1, \ldots, f_n) \in \{0, 1\}^{n+1}$, where $f_i (i = 0, 1, \ldots, n)$ is the function value of f and its input vector of weight i, and the set of SBFs can be regarded as an $n + 1$-dimensional linear space. Given $n+1$ linearly independent SBFs, they can form a complete set of bases for any n-variable SBFs.

Before presenting our protocol, we first consider the quantum implementation of SBFs. Let $V = R_y(\pi) = -\mathrm{i}Y$ be the π rotation around the y axis of

*Supported in part by the National Key R&D Program of China (Grant No. 2017YFB0802400), the National Natural Science Foundation of China (Grant Nos. 61373171 and 11801564), the Program for Excellent Young Talents in University of Anhui Province, China (Grant No. gxyqZD2019060), the Basic Research Project of Natural Science of Shaanxi Province, China (Grant Nos. 2017JM6037 and 2017JQ1032), and the Key Project of Science Research of Anhui Province, China (Grant No. KJ2017A519).
**Corresponding authors. Email: wp_ma@mail.xidian.edu.cn; zyxue83@163.com
© 2020 Chinese Physical Society and IOP Publishing Ltd

the Bloch sphere, $U_{(k)} = R_y(\pi/k)$ $(k \in \{1, 2, \ldots, n\})$ be the π/k rotation around the y axis, and $U_{(0)} = R_y(0) = I$ be the identity operation, we can obtain

$$U_{(k)}U_{(h)} = U_{(h)}U_{(k)}; \quad U^{\dagger}_{(k)}U_{(h)} = U_{(h)}U^{\dagger}_{(k)}. \tag{1}$$

Theorem 1. Let $k(1 \leq k \leq n)$ be a positive integer, and $x_i \in \{0, 1\}(i = 1, 2, \ldots, n)$. Then, the following propositions hold.

(i) Denote $U(n, k, x) = (U^{\dagger}_{(k)})^{(\sum_i x_i) \bmod k} U^{x_n}_{(k)} \cdots U^{x_2}_{(k)} U^{x_1}_{(k)}$, it can implement an n-variable SBF f^k_n, i.e.,

$$U(n, k, x)|0\rangle = |f^k_n\rangle. \tag{2}$$

(ii) Then $\deg(f^k_n) \geq k$ and there is no monomial with degree less than k in the algebraic norm form (ANF) of f^k_n.

(iii) The ANF of f^k_n contains all monomials with degree k.

(iv) Let $f^0_n = 1$, then each n-variable SBF can be represented by the linear combination of $f^k_n (k = 0, 1, \ldots, n)$.

Proof. (i) First, we show that $U(n, k, x)$ can be regarded as the quantum implementation of a special n-variable Boolean function f, i.e., $U(n, k, x)|0\rangle$ is either $|0\rangle$ or $|1\rangle$. Let $wt(x_1, x_2, \ldots, x_n) = \sum_i x_i = ak+b$ with a and b being nonnegative integers, and $0 \leq b < k$, then

$$\begin{aligned} U(n, k, x)|0\rangle =& (U^{\dagger}_{(k)})^{(\sum_i x_i) \bmod k} U^{\sum_i x_i}_{(k)}|0\rangle \\ =& (U^{\dagger}_{(k)})^{b} U^{ak+b}_{(k)}|0\rangle \\ =& (U_{(k)})^{ak}|0\rangle \\ =& \left[R_y(\frac{\pi}{k})\right]^{ak}|0\rangle \\ =& R_y(a\pi)|0\rangle \in \{|0\rangle, |1\rangle\}. \end{aligned} \tag{3}$$

Hence, $U(n, k, x)$ implements a special Boolean function f.

Second, we show that f is an SBF. From Eq. (3), $f = 1$ if and only if $R_y(a\pi)|0\rangle = |1\rangle$ for odd a. Let $a' = \lfloor a/2 \rfloor$ be a nonnegative integer, then $a = 2a'$ if a is even and $a = 2a' + 1$ if a is odd. We have

$$f = \begin{cases} 0, & wt(x) = 2a'k + b, \ (0 \leq b < k) \\ 1, & wt(x) = (2a' + 1)k + b, \ (0 \leq b < k) \end{cases} \tag{4}$$

thus f is an SBF. Denote the special n-variable Boolean function f as f^k_n, we can reach Eq. (2).

(ii) For each vector $x \in \{0, 1\}^n$ with $wt(x) = k$, from Eq. (4), we have $wt(x_1, x_2, \ldots, x_n) = k$, which implies $f^k_n(x_1, x_2, \ldots, x_n) = 1$. Hence, f^k_n is a nonzero Boolean function. Similarly, for each vector $x \in \{0, 1\}^n$ with $wt(x) = b < k$, we have $wt(x_1, x_2, \ldots, x_n) = b$, which implies $f^k_n(x_1, x_2, \ldots, x_n) = 0$. Because $\deg(f^k_n) \geq k$, there is no monomial with degree less than k in the ANF of f^k_n.

(iii) Assuming that a k-degree monomial $x_{i_1}x_{i_2}\cdots x_{i_k}$ with $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ does not appear in the ANF of f^k_n. Let us select a vector β from $\{0, 1\}^n$ such that the elements of β are all 0 except for the i_jth $(j = 1, 2, \ldots, k)$ elements, which are 1. Then, from (ii), it is obvious that $f^k_n(\beta) = 0$. On the other hand, we can obtain $f^k_n(\beta) = 1$ from Eq. (4) as $wt(\beta) = k$, which is in a contradiction. Hence, the ANF of f^k_n contains all monomials with degree k.

(iv) To prove that each n-variable SBF can be represented by the linear combination of $f^k_n (k = 0, 1, \ldots, n)$, we only need to show that they are linearly independent; i.e.,

$$a_0 f^0_n \oplus a_1 f^1_n \oplus \cdots \oplus a_n f^n_n = 0, \tag{5}$$

only has zero solution. We first consider the coefficient a_0. Note that $f^k_n (k = 1, 2, \ldots, n)$ does not contain a monomial with degree less than k and $f^0_n = 1$, thus we obtain $a_0 = 0$. Then, Eq. (5) reduces to

$$a_1 f^1_n \oplus \cdots \oplus a_n f^n_n = 0. \tag{6}$$

Next, we consider the coefficient a_1. Note that $f^k_n (k = 2, 3, \ldots, n)$ does not contain a monomial with degree less than k, we obtain that the coefficient of any monomial with degree 1 on the left side of Eq. (6) is a_1, which is equal to 0, thus $a_1 = 0$. Similarly, we can deduce that $a_2 = \cdots = a_n = 0$. Therefore, the $n + 1$ SBFs f^k_n with $k = 0, 1, \ldots, n$ are linearly independent, and they form a complete set of basis for n-variable SBFs; that is, an arbitrary n-variable SBF can be represented by the linear combination of f^k_n.

From Eq. (1) and theorem 1, we obtain

$$VU(n, k, x, r)|0\rangle = |\bar{r} \oplus f^k_n\rangle, \tag{7}$$

where

$$VU(n,k,x,r) = (U^{\dagger}_{(k)})^{(\sum_i x_i) \bmod k} V^{r_n} U^{x_n}_{(k)} \cdots V^{r_2} U^{x_2}_{(k)} V^{r_1} U^{x_1}_{(k)},$$

$r_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, and $\bar{r} = \oplus^n_{i=1} r_i$.

Theorem 2. Let k and h be two nonnegative integers, then

$$|f^k_n \oplus f^h_n\rangle = \quad U(n, k, x)U(n, h, x)|0\rangle. \tag{8}$$

Proof. First, let $r_n = r_{n-1} = \cdots r_2 = 0$ and $r_1 = 1$ in Eq. (7), we get

$$U(n, k, x)|1\rangle = |f^k_n \oplus 1\rangle. \tag{9}$$

Next, we consider the right hand of Eq. (8),

$$\begin{aligned} &U(n, k, x)U(n, h, x)|0\rangle = U(n, k, x)|f^h_n\rangle \\ &= \begin{cases} |f^k_n\rangle & \text{if} \quad f^h_n = 0 \\ |f^k_n \oplus 1\rangle & \text{if} \quad f^h_n = 1 \end{cases} \\ &= |f^k_n \oplus f^h_n\rangle. \end{aligned} \tag{10}$$

We now proceed to present our QSMPC protocol.

Let $f(x_1, x_2, \ldots, x_n)$ be an n-variable SBF which can be presented as $f = \bigoplus_{k=0}^{n} a_k f^k_n$, then $|f\rangle = \prod_{k=0}^{n} [U(n, k, x)]^{a_k}|0\rangle$. Let $C_i (i = 1, 2, \ldots, n)$ be n clients and each client C_i possesses a private input x_i

and select a random bit r_i. We 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 $x_{i,1}, x_{i,2}, \ldots x_{i,n}$ with $x_{i,j} \in \{0, 1, \ldots, k-1\}, j = 1, 2, \ldots, n$, and the random bit r_i into n elements $r_{i,1}, r_{i,2}, \ldots r_{i,n}$ with $r_{i,j} \in \{0, 1\}, j = 1, 2, \ldots, n$, such that $\sum_{j=1}^{n} x_{i,j} \equiv x_i \bmod k$ 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 through a secure authenticated channel. Third, each client C_i computes $\tilde{x}_i = \sum_{j=1}^{n} x_{j,i} \bmod k$ and $\tilde{r}_i = \oplus_{j=1}^{n} r_{j,i}$.

Step 2. First, the server S prepares a single particle in the state of $|0\rangle$ and sends it to the client C_1. Second, C_1 performs the quantum operation $V^{r_1} U_{(k)}^{x_1}$ on the received particle according to his private input x_1 and random bit r_i, and sends the particle to the client C_2 who will then perform the operation $V^{r_2} U_{(k)}^{x_2}$ on the particle according to his private input x_2 and random bit r_2. This process continues until all the clients have applied their operations to the particle. After this process, the 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 a secure authenticated channel. Second, C_n calculates $(\sum_{i=1}^{n} \tilde{x}_i) \bmod k$. Third, C_n applies $(U_{(k)}^{\dagger})^{(\sum_{i=1}^{n} \tilde{x}_i) \bmod k}$ on the particle, which leads the state of the particle to $|f_n^k \oplus \bar{r}\rangle$, due to the fact that $(\sum_{i=1}^{n} \tilde{x}_i) \bmod k = (\sum_{i=1}^{n} x_i) \bmod k$. Fourthly, C_n sends the particle back to the server S. Then, the server can get the state $|f_n^k \oplus \bar{r}\rangle$ by measuring the received particle, and announce $f_n^k \oplus \bar{r}$ to all the 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 a secure authenticated channel. Second, each client C_i calculates $\oplus_{j=1}^{n} \tilde{r}_j = \oplus_{j=1}^{n} r_j = \bar{r}$. Finally, each client can extract the value of f_n^k by performing the XOR operation $f_n^k = (f_n^k \oplus \bar{r}) \oplus \bar{r}$.

We now turn to analyze the security of our scheme and we only focus on the internal attack from the clients or the server, as internal attacks are usually more effective than external attacks.[18] We assume that all the clients and the server will execute the QSMPC protocol faithfully. However, they also try to get the private inputs or the function output f_n^k of others.

(1) Considering the security against the attack from the server S. First, if the server S wants to extract the private input of a client, say C_1, he must intercept the particle sent from C_1 to C_2, and then will measure it in the correct basis $\{|0\rangle, |1\rangle\}$ or $\{U_{(k)}|0\rangle, U_{(k)}|1\rangle\}$. However, he cannot choose the right measurement basis as he does not know the unitary operation $V^{r_1} U^{x_1}$ and the state information of the particle. Second, the server S cannot also extract the function output f_n^k because he knows nothing about the random classical bits $r_i (i = 1, 2, \ldots, n)$, $\tilde{r}_i (i = 1, 2, \ldots, n)$ and $\bar{r}$.

(2) Considering the attack from the clients. Let us assume a very unfavorable situation, i.e., the dishonest clients consist of C_n who can get the information of $(\sum_{i=1}^{n} \tilde{x}_i) \bmod k$ as well as other $n-t-1$ clients $C_{t+1}, \ldots, C_{n-1}$, and they collaborate to extracts the private inputs of $C_1, \ldots, C_t$. Apparently, the dishonest clients could easily access the $(\sum_{i=1}^{t} \tilde{x}_i) \bmod k$ from $(\sum_{i=1}^{n} \tilde{x}_i) \bmod k$ and $x_{t+1}, \ldots, x_{n-1}, x_n$. If $t = 1$ (i.e., only one client is honest, which is almost impossible), then they can extract the private input x_1; otherwise, they cannot get any unknown information about the private inputs of the honest clients. Therefore, our protocol will be secure whenever two or more clients are honest, which can be met in general.

In addition, the no-go theorem shows that all quantum bit commitment and protocols based on it are insecure. However, QSMPC protocols restricted by the no-go theorem contain only participants, not the third party. Owing to the existence of the server S, which can be considered as a third party, our protocol can overcome the no-go theorem.

We now turn to discuss the efficiency of related implementations. In classical case, SMPC is usually achieved by garbled circuits.[11] Similarly, quantum circuits, consisting of a number of quantum gates, can be used to perform QSMPC. Previous QSMPC implementations involve the usage of classical NAND gate, which are based on either an entangled GHZ state or a single-qubit state.[6] In the process of performing NAND gate based on single qubit, one qubit, two single-qubit quantum gates and several rounds of classical communication are involved. For the NAND gate implementation based on GHZ entangled state, three qubits, four single-qubit quantum gates, and several rounds of classical communication are involved. Furthermore, these protocols guarantee no security for the inputs.[8]

Quantum implementation of Boolean function and its application in QSMPC were explored in Ref. [8], where one qubit, $2n+1$ single-qubit quantum gates and one round classical communication are needed. Hence, the protocol in Ref. [8] is more efficient.

Our protocol also needs one qubit, $2n+1$ single-qubit quantum gates and one round classical communication when n clients are involved. Hence, the efficiency of these two protocols is the same. Meanwhile, Ref. [8] requires $n-1$ qubits to compute the n-tuple pairwise AND function. In comparison, only 1 qubit is needed to do this in our protocol. For example, to calculate the n-tuple pairwise AND function $x_1 x_2 \cdots x_n$, the server and n participants only need to perform unitary operations on the qubit $|0\rangle$, i.e., $(U_{(n)}^{\dagger})^{(\sum_{i=1}^{n} \tilde{x}_i) \bmod n} V^{r_n} U_{(n)}^{x_n} \cdots V^{r_2} U_{(n)}^{x_2} V^{r_1} U_{(n)}^{x_1} = |x_1 x_2 \cdots x_n \oplus \bar{r}\rangle (\bar{r} = \oplus_{j=1}^{n} r_j)$. Therefore, our protocol is more efficient than theirs in this task.

Finally, we simulate our QSMPC protocol on the IBM quantum cloud (i.e., ibmqx4 quantum computer), and only employ the available quantum gates of

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda} \sin\frac{\theta}{2} \\ e^{i\phi} \sin\frac{\theta}{2} & e^{i(\lambda+\phi)} \cos\frac{\theta}{2} \end{pmatrix}, \quad (11)$$

where θ, ϕ and λ represent the Y axis rotation an-

gle, the pre-phase rotation angle and the post-phase rotation angle of a quantum state, respectively.

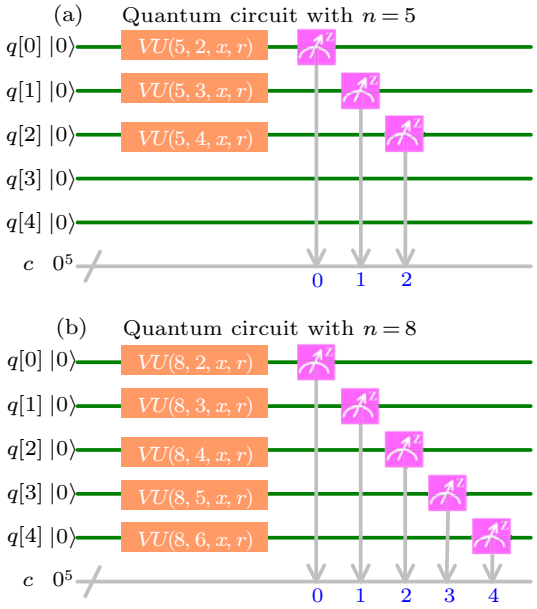

Fig. 1. The proposed quantum circuits for QSMPC. (a) The case of $n=5$, the outputs f_5^4, f_5^3 and f_5^2 are obtained by measuring $q[2]$, $q[1]$ and $q[0]$. (b) For the $n=8$ case, the outputs f_8^6, f_8^5, f_8^4, f_8^3, and f_8^2 are obtained by measuring $q[4]$, $q[3]$, $q[2]$, $q[1]$ and $q[0]$.

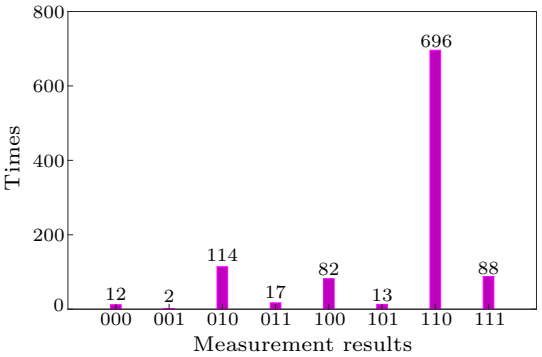

Fig. 2. Measurement results of (f_5^4, f_5^3, f_5^2) for a total of 1024 times of experiments.

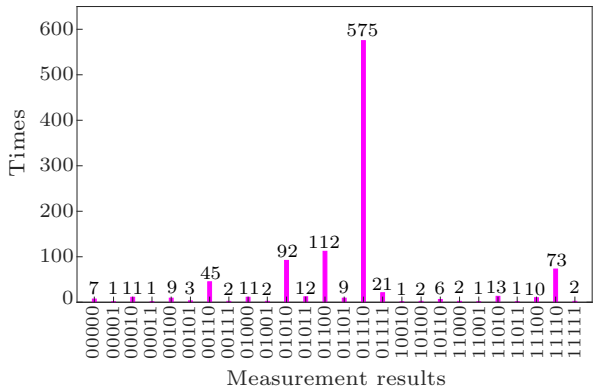

Fig. 3. Measurement results $(f_8^6\oplus 1, f_8^5\oplus 1, f_8^4\oplus 1, f_8^3\oplus 1, f_8^2\oplus 1)$ in a total of 1024 times of experiments, where the results $(1,0,0,0,1)$, $(1,0,0,1,1)$, $(1,0,1,0,1)$, $(1,0,1,1,1)$ and $(1,1,1,0,1)$ do not appear.

Suppose that the preparation and measurement of the qubits are operated by the server (i.e., $q[0]$, $q[1]$, $q[2]$, $q[3]$, and $q[4]$). The private inputs of n clients and the random classical bits selected can be represented as $x=(x_1,x_2,\ldots,x_n)$ and $r=(r_1,r_2,\ldots,r_n)$. Figure 1 shows the quantum circuits of our protocol with $n=5$ and $n=8$, where $U_{(k)}$ and V are realized by the quantum gates of $U_3(\frac{\pi}{k},0,0)$ and $U_3(\pi,0,0)$, respectively; and the identity operation I can be simply realized by leaving the particle to be idle. Due to the limitation of the quantum computing resource, we only select a few input vectors randomly to verify the protocol. Through verification, we find that the probability of the correct output of the quantum computer remains above 80%, and the highest correct probability is as high as 94.63% for the f_8^2 case.

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_5^2, f_5^3, and f_5^4 with $x=(x_1,x_2,\ldots,x_5)=(1,1,0,0,1)$, $r=(r_1,r_2,\ldots,r_5)=(1,0,0,1,0)$, $(\sum_i^5 x_i) \bmod 2 = 1$, $(\sum_i^5 x_i) \bmod 3 = 0$, $(\sum_i^5 x_i) \bmod 4 = 3$ and $\bar{r}=\oplus_{i=1}^5 r_i = 0$. The quantum implementation of f_5^k with $k=2,3,4$ is illustrated in Fig. 1(a). Here,

$$
\begin{aligned}
&VU(5,2,x,r)\\
&=(U_{(2)}^{\dagger})^{(\sum_i x_i) \bmod 2} V^{r_5} U_{(2)}^{x_5} V^{r_4} U_{(2)}^{x_4} V^{r_3} U_{(2)}^{x_3} V^{r_2} U_{(2)}^{x_2} V^{r_1} U_{(2)}^{x_1}\\
&=U_{(2)}^{\dagger} V^0 U_{(2)}^1 V^1 U_{(2)}^0 V^0 U_{(2)}^0 V^0 U_{(2)}^1 V^1 U_{(2)}^1\\
&=U_{(2)}^{\dagger} I U_{(2)} V I I I I U_{(2)} V U_{(2)}.
\end{aligned}
$$

Similarly,

$$VU(5,3,x,r)=(U_{(3)}^{\dagger})^0 I U_{(3)} V I I I I U_{(3)} V U_{(3)},$$

$$VU(5,4,x,r)=(U_{(4)}^{\dagger})^3 I U_{(4)} V I I I I U_{(4)} V U_{(4)}.$$

The measurement results are presented in Fig. 2, and the probabilities of correct output f_5^k are 85.84%, 89.36%, and 88.28% for $k=2$, 3, and 4 cases, respectively.

Case 2: $n=8$. Let $C_1, C_2, \ldots,$ and C_8 be the eight clients. They will collaborate to compute the function $f_8^2, f_8^3, f_8^4, f_8^5$, and f_8^6 with $x=(x_1,x_2,\ldots,x_8)=(1,0,1,1,1,0,1,0)$, $r=(r_1,r_2,\ldots,r_8)=(1,1,1,0,0,1,0,0)$, $(\sum_i^8 x_i) \bmod 2 = 1$, $(\sum_i^8 x_i) \bmod 3 = 2$, $(\sum_i^8 x_i) \bmod 4 = 1$, $(\sum_i^8 x_i) \bmod 5 = 0$, $(\sum_i^8 x_i) \bmod 6 = 5$ and $\bar{r}=\oplus_{i=1}^8 r_i = 0$. The quantum implementation of $f_8^k (k=2,3,4,6,6)$ is shown in Fig. 1(b). Here,

$$
\begin{aligned}
VU(8,2,x,r)&=U_{(2)}^{\dagger} III U_{(2)} VIIU_{(2)} IU_{(2)} VU_{(2)} VIVU_{(2)},\\
VU(8,3,x,r)&=(U_{(3)}^{\dagger})^2 III U_{(3)} VIIU_{(3)} IU_{(3)} VU_{(3)} VIVU_{(3)},\\
VU(8,4,x,r)&=U_{(4)}^{\dagger} III U_{(4)} VIIU_{(4)} IU_{(4)} VU_{(4)} VIVU_{(4)},\\
VU(8,5,x,r)&=(U_{(5)}^{\dagger})^0 III U_{(5)} VIIU_{(5)} IU_{(5)} VU_{(5)} VIVU_{(5)},\\
VU(8,6,x,r)&=(U_{(6)}^{\dagger})^5 III U_{(6)} VIIU_{(6)} IU_{(6)} VU_{(6)} VIVU_{(6)}.
\end{aligned}
$$

The measurement results $(f_8^2\oplus\bar{r}, f_8^3\oplus\bar{r}, f_8^4\oplus\bar{r}, f_8^5\oplus\bar{r}, f_8^6\oplus\bar{r}) = (f_8^2, f_8^3, f_8^4, f_8^5, f_8^6)$ are presented in Fig. 3, and the probabilities of correct output f_8^k are 94.63%, 83.50%, 84.86%, 91.41%, and 89.16% for $k=2$, 3, 4, 5, and 6 cases, respectively.

In conclusion, we have explored the quantum realization of SBFs and demonstrated that a class of n-variable SBFs f_n^k can be used as subroutines for QSMPC protocols. In our protocol, a single qubit can be used to compute arbitrary symmetric Boolean functions. In addition, the choice of the BSFs used for QSMPC can be arbitrary in our protocol, thus it may be potentially more secure. Therefore, our protocol is of practical importance for the implementation of QSMPC. Another advantage of our scheme is that only 1 qubit is needed when computing the n-tuple pairwise AND function, which is more efficient than others in this task. Moreover, we demonstrate our protocol on IBM quantum cloud platform with the probability of the correct output as high as 94.63% for the case of f_8^2. Therefore, our protocol represents a promising generalization in realization of multipartite quantum tasks.

References

[1] Yao A C 1982 *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science* (Chicago, USA 3–5 November 1982) p 160
[2] Yao A C 1986 *Proceedings of the 27rd Annual Symposium on Foundations of Computer Science* (Toronto, Canada 27–29 October 1982) p 162
[3] Goldreich O, Micali S and Wigderson A 1987 *Proceedings of the 19th Annual ACM Symposium on Theory of Computing* (New York, USA 25–27 May 1987) p 218
[4] Anders J and Browne D E 2009 *Phys. Rev. Lett.* **102** 050502
[5] Loukopoulos K and Browne D E 2010 *Phys. Rev. A* **81** 062336
[6] Dunjko V, Kapourniotis T and Kashefi E 2016 *Quantum Inf. Comput.* **16** 0061
[7] Barz S, Dunjko V, Schlederer F, Moore M, Kashefi E and Walmsley I A 2016 *Phys. Rev. A* **93** 032339
[8] Clementi M, Pappa A, Eckstein A, Walmsley I A, Kashefi E and Barz S 2017 *Phys. Rev. A* **96** 062317
[9] Cao H and Ma W 2018 *Laser Phys. Lett.* **15** 095201
[10] He G P 2018 *Phys. Scr.* **93** 095001
[11] Huang Y, Evans D, Katz J and Malka L 2011 *Proceedings of the 20th USENIX Security Symposium* (San Francisco, USA 10–12 August 2011) p 331
[12] Wei C Y, Cai X Q, Wang T Y, Qin S J, Gao F and Wen Q Y 2020 *IEEE J. Sel. Area. Commun.* (in press)
[13] Wei C Y, Cai X Q, Liu B, Wang T Y and Gao F 2018 *IEEE Trans. Comput.* **67** 2
[14] Gao F, Qin S J, Huang W and Wen Q Y 2019 *Sci. Chin.-Phys. Mech. Astron.* **62** 070301
[15] Yang Y G, Yang R, Cao W F, Chen X B, Zhou Y H and Shi W M 2017 *Int. J. Theor. Phys.* **56** 1286
[16] Lo H K 1997 *Phys. Rev. A* **56** 1154
[17] Lo H K and Chau H F 1997 *Phys. Rev. Lett.* **78** 3410
[18] Yang Y G, Liu Z C, Li J, Chen X B, Zuo H J, Zhou Y H and Shi W M 2016 *Phys. Lett. A* **380** 4033