Quantum protocols for secure multi-party summation

Preprint *in* Quantum Information Processing · March 2019

DOI: 10.1007/s11128-018-2141-1

6 authors, including:

Zhaoxu Ji
Wuhan University
5 PUBLICATIONS **29** CITATIONS

SEE PROFILE

Huanguo Zhang
Wuhan University
62 PUBLICATIONS **214** CITATIONS

SEE PROFILE

Houzhen Wang
University of Victoria
14 PUBLICATIONS **40** CITATIONS

SEE PROFILE

Quantum Information Processing (2019) 18:168
https://doi.org/10.1007/s11128-018-2141-1

Quantum protocols for secure multi-party summation

ZhaoXu Ji[1] · HuanGuo Zhang[1] · HouZhen Wang[1] · FuSheng Wu[1] · JianWei Jia[1] · WanQing Wu[2]

Received: 27 March 2018 / Accepted: 28 November 2018

Abstract
Quantum secure multi-party summation is a kind of privacy-preserving summation whereby multiple mutually distrustful parties can securely compute the summation of their secret data, which can be utilized to execute many tasks of quantum secure multi-party computation, such as quantum anonymous surveying. In this paper, we present two quantum secure multi-party summation protocols, both of which allow multiple mutually distrustful parties to securely compute the summation of their secret data, where the dataset of the summation is supposed to be a set of nonnegative integers. Our protocols have two main common features. One is the assumption of a semi-honest third party who helps multiple parties to perform the summation computations and announces the results, at the same time he is allowed to misbehave on his own, but cannot conspire with any party. Another is the use of the entanglement swapping of d-level cat states and Bell states to maintain privacy, and pass information between each party and the semi-honest third party. We analyze the success probability of our two protocols, respectively. In addition, we generalize the use of our second protocol to execute the tasks of quantum anonymous ranking and quantum anonymous voting. What's more, we show that our protocols can resist various attacks from both outside attackers and inside participants.

Keywords Secure multi-party computation · Quantum secure multi-party summation · Entanglement swapping

✉ HouZhen Wang
whz@whu.edu.cn

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

[2] School of Cyber Security and Computer, Hebei University, Baoding 071002, China

Published online: 20 April 2019

1 Introduction

Information security concerns national security, and concerns the society stabilization [1]. With the development of information technology, the threat events of information security often happen, which has triggered enormous demand for multi-party computation in which some people jointly conduct a series of computation tasks based on their private data.

Secure multi-party computation (SMC) [2], which enables $n (n \geq 2)$ parties to jointly compute a function based on their private inputs while keeping these inputs private at the same time, is a fundamental primitive in modern cryptography. It has wide applications in private bidding and auctions, secret ballot elections, e-commerce and data mining and so on, and has long been the object of intensive study in classical cryptography. The security of SMC is based on the computation complexity assumption. However, with the rapid development of quantum computers, this condition becomes impractical due to the strong ability of quantum computers [3–5], thus inspiring much interest in generalizing the classical SMC into its counterpart in the realm of quantum mechanics, i.e., quantum secure multi-party computation (QSMC) can provide the unconditional security, which is guaranteed by physical principles of quantum mechanics. Since the pioneering work of quantum cryptography was proposed by Bennett and Brassard [6] in 1984, various QSMC protocols have been proposed, such as quantum secret sharing (QSS) [7–9], quantum anonymous ranking (QAR) [10,11].

Secure multi-party summation problem is a special problem in real life, which can be described as follows: There are n parties $P_1, P_2, \ldots, P_n$, and each party P_i has a secret value x_i. They want to correctly calculate a summation function $F(x_1, x_2, \ldots, x_n)$ without revealing any party's secret value. The result of the function F could be revealed in public or privately to some particular participant. It is known that secure multi-party summation is a fundamental primitive of SMC, which can be used to build complex secure protocols for other multi-party computations, especially numerical computations. In addition, there are also lots of other important applications of secure multi-party summation in distributed networks, such as secret sharing, electronic voting, secure sorting, and data mining [12,13].

Quantum secure multi-party summation (QSMS), which can be seen as the generalization of classical secure multi-party summation into the realm of quantum mechanics, was first proposed by Vaccaro et al. [14] in their quantum protocols for anonymous voting and surveying. Quantum anonymous voting (QAV) is an attractive application of quantum mechanics, and it is also an important branch of QSMC [14–16]. In a QAV protocol, assume that there are multiple voters, and they are each to vote yes or no on a question, where the value of the vote is a binary. The identities of the voters are kept private, although the sum of the votes is made public at the end of the protocol. Besides the voters, there is also a tallyman who provides the quantum resources for voting and counts the votes. Surveying is similar to voting in most respects [14]. The main difference between voting and surveying is the value in surveying is not restricted to a binary yes or no but may take any integer value. As such, surveying corresponds to collecting estimates of some numerical quantity, such as profit and loss values. The process of counting the votes or collecting estimates of some numerical quantity is actually a kind of summation process.

Inspired by the ideas in Refs. [14,17], in this paper, we propose two QSMS protocols, one is the quantum secure multi-party multi-data summation (QSMMS) protocol, the other is the quantum secure multi-party single-datum summation (QSMSS) protocol. Both of our protocols allow multiple mutually distrustful parties to securely compute the summation of their secret data while at the same time keeping their data private, where the dataset of the summation is supposed to be a set of nonnegative integers. Our protocols have two main common features. One is the assumption of a semi-honest third party who helps multiple parties to perform some necessary computations, and at the same time, he is allowed to misbehave on his own, but cannot conspire with any party [17]. Another is the use of the entanglement swapping of d-level cat states and Bell states [8] (we call this entanglement swapping ESCB for the sake of simplicity in this paper) to maintain privacy, and pass information between each party and the semi-honest third party. The main difference between QSMMS and QSMSS is that each party has a secret number set in the former, while each party just has one datum in the latter. One can choose one of them to compute the summation of some data according to real-life situations. For instance, a class of students take exams and the test subjects include math, Chinese, and English, while the test scores are confidential. Now a teacher and these students want to compute the summation of the test scores for each subject so as to compute the average score. However, these students want their scores to be kept secret during the process of the summation in order to protect self-esteem. In this situation, they can choose the QSMMS protocol to perform the computation. In addition, if they just want to compute the summation of one subject (e.g., math), they can choose the QSMSS protocol.

Based on quantum protocols for anonymous voting and surveying [14], we set a number of general rules for QSMS.

(R1) The data of each party should be kept secret from others, even after the end of the protocol.
(R2) All parties get the summation result at the same time.
(R3) The summation result should be correct, and all malicious behaviors during the process can be discovered.
(R4) The semi-honest third party is allowed to misbehave on its own but cannot conspire with any party.

The rest of this paper is organized as follows. We devote Sect. 2 to introduce quantum resources used in our protocols. We devote Sect. 3 to present our QSMMS protocol, analyze its success probability, and introduce a simple example. We devote Sect. 4 to present our QSMSS protocol and analyze its success probability, and then we generalize its use to QAR and QAV, respectively. The security of these protocols with respect to various kinds of attacks is analyzed in Sect. 5. In Sect. 6, we first point out the main differences between our two protocols, and then we compare our QSMMS protocol with previous protocols. Finally, we conclude our paper with conclusions.

2 Quantum resources used in our protocols

Let us start by introducing the quantum resources used in our protocols.

The quantum Fourier transform on d-level basis state $|k\rangle$ ($k \in \{0, 1, \ldots, d-1\}$) is defined as

$$F\,|k\rangle = \frac{1}{\sqrt{d}} \sum_{r=0}^{d-1} \zeta^{kr}\,|r\rangle\,, \tag{1}$$

where $\zeta = e^{2\pi i/d}$. One can construct two common non-orthogonal bases $V_1 = \{|k\rangle\}_{k=0}^{d-1}$ and $V_2 = \{F\,|k\rangle\}_{k=0}^{d-1}$.

A generalization of the familiar Bell states in d-level system (qudits) is a set of d^2 maximally entangled states which form an orthonormal basis for the space of two qudits [8], can be expressed as

$$|\Psi(u, v)\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \zeta^{ju}\,|j\rangle\,|j \oplus v\rangle\,, \tag{2}$$

where the variables u and v run from 0 to $d-1$, and the symbol $\oplus$ denotes addition modulo d throughout this paper. Easily, one can get

$$|\Psi(0, 0)\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} |j\rangle\,|j\rangle\,. \tag{3}$$

A Bell state $|\Psi(u, v)\rangle$ can be generated by acting on $|\Psi(0, 0)\rangle$ with $\left|U_{(u,v)}\right\rangle$,

$$\left|U_{(u,v)}\right\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \zeta^{ju}\,|j \oplus v\rangle\,\langle j|\,, \tag{4}$$

i.e.,

$$(I \otimes \left|U_{(u,v)}\right\rangle)\,|\Psi(0, 0)\rangle = |\Psi(u, v)\rangle\,, \tag{5}$$

where I is the identity matrix and the symbol $\otimes$ denotes the tensor product throughout this paper.

The d-level n-particle cat states have the form

$$|\Psi(u_1, u_2, \ldots, u_n)\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \zeta^{ju_1}\,|j, j \oplus u_2, j \oplus u_3, \ldots, j \oplus u_n\rangle\,, \tag{6}$$

where each of the labels $u_1, u_2, \ldots, u_n$ runs from 0 to $d-1$ [8]. These cat states are orthonormal and complete, that is,

$$\langle\Psi(u_1, u_2, \ldots, u_n)\,\left|\Psi(u_1', u_2', \ldots, u_n')\right\rangle = \delta_{u_1,u_1'}\delta_{u_2,u_2'}\cdots\delta_{u_n,u_n'}\,. \tag{7}$$

The formula of entanglement swapping between a cat state $|\Psi(u_1, u_2, \ldots, u_n)\rangle$ and a d-level Bell state $\left|\Psi(v, v')\right\rangle_{s,s'}$, which does not involve the first particle of the cat state (i.e., $m \in \{2, 3, \ldots, n\}$) [8], can be expressed as

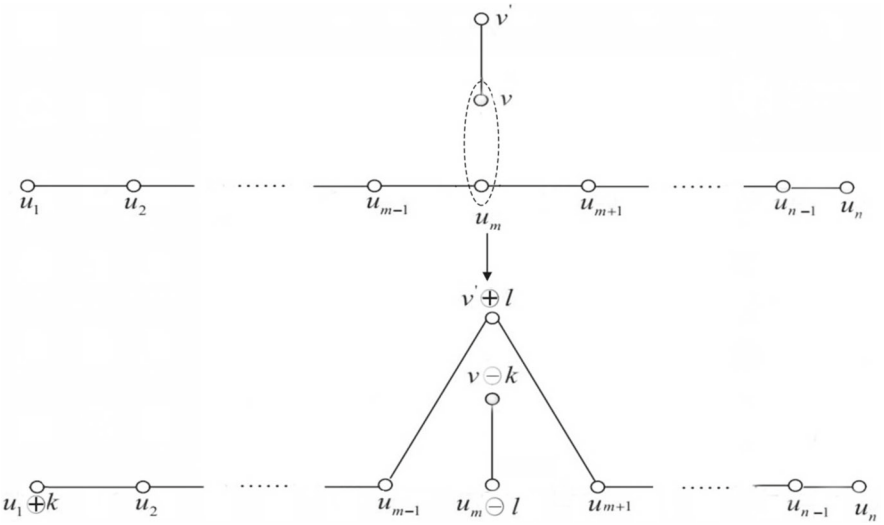

Fig. 1 The graphical description of entanglement swapping between one d-level n-particle cat state and one d-level Bell state. The small circle represents a qubit, and the two circles connected by a line represents a pair of entangled qubits. The dotted ellipse represents the d-level Bell measurement

$$
\begin{aligned}
&|\Psi(u_1, u_2, \ldots, u_n)\rangle_{1,2,\ldots,n} \otimes \left|\Psi(v, v')\right\rangle_{s,s'} = \\
&\frac{1}{d}\sum_{k,l}^{d-1} \zeta^{kl} \left|\Psi(u_1 \oplus k, u_2, u_3, \ldots, v' \oplus l, \ldots, u_n)\right\rangle_{1,2,\ldots,s',\ldots,n} \\
&\quad \otimes |\Psi(v \ominus k, u_m \ominus l)\rangle_{s,m} ,
\end{aligned} \tag{8}
$$

where the symbol $\ominus$ denotes subtraction modulo d throughout this paper. The graphical description of this entanglement swapping is given in Fig. 1, where the cat state is depicted by n small circles connected by $n-1$ lines, and the d-level Bell state is depicted by two small circles connected by one line [8].

3 Protocol I: quantum secure multi-party multi-data summation protocol

Assumptions: Assume that there are $n (n \geq 2)$ mutually distrustful parties labeled $P_1, P_2, \ldots, P_n$, and each party P_i $(i = 1, 2, \ldots, n)$ has a secret number set $D_{P_i} = \{x_i^1, x_i^2, \ldots, x_i^L\}$, where $D_{P_i} \subset \{0, 1, 2, \ldots, d-1\}$ and d is a positive integer which is larger than the maximum element in $D = D_{P_1} \cup D_{P_2} \cup \cdots \cup D_{P_n}$. Here, we set $S = \{x_i^j | x_i^j \in N, 0 \leq x_i^j \leq d-1, i = 1, 2, \ldots, n, j = 1, 2, \ldots, L\}$, and then we have $d > \sup\{S\}$. $P_1, P_2, \ldots, P_n$ want to jointly compute the summations $\sum_{i=1}^n x_i^1$, $\sum_{i=1}^n x_i^2$, ..., $\sum_{i=1}^n x_i^L$ with the assistance of a semi-honest third party (named TP) who is allowed to misbehave on its own but cannot conspire with others.

Now let us describe the steps of this protocol in detail.

Step 1: Prepare quantum states. Each party (P_i) prepares L copies of d-level Bell state

$$|\Psi(0,0)\rangle = \frac{1}{\sqrt{d}} \sum_{t=0}^{d-1} |t\rangle\,|t\rangle\,. \tag{9}$$

TP prepares L copies of the cat states $|\Psi(u_0, u_1, \ldots, u_n)\rangle$, and marks them by

$$\left|\Psi(u_0^1, u_1^1, \ldots, u_n^1)\right\rangle, \left|\Psi(u_0^2, u_1^2, \ldots, u_n^2)\right\rangle, \ldots, \left|\Psi(u_0^L, u_1^L, \ldots, u_n^L)\right\rangle \tag{10}$$

in turn to generate an ordered sequence, where the superscripts denote the order of the cat states in the sequence. Then he takes the particles with marks $(u_1^j, u_2^j, \ldots, u_n^j)$ out from each cat state $\left|\Psi(u_0^j, u_1^j, u_2^j, \ldots, u_n^j)\right\rangle$ to construct new sequences, and marks them by

$$(u_1^1, u_1^2, \ldots, u_1^L), (u_2^1, u_2^2, \ldots, u_2^L), \ldots, (u_n^1, u_n^2, \ldots, u_n^L), \tag{11}$$

in turn, and denotes them by $S_1, S_2, \ldots, S_n$, respectively. Subsequently, TP tells P_i the marks ($u_i^1, u_i^2, \ldots, u_i^L$).

Step 2: Distribution. For the sake of checking the presence of eavesdroppers, TP prepares n sets of decoy photons, and denotes them by $D_1, D_2, \ldots, D_n$, respectively, where each decoy photon randomly is in one of the states from the set V_1 or V_2. Then he inserts the decoy photons of D_i into S_i at random positions and records their insertion positions. Denote the new sequence by S_i'. Finally, TP sends S_i' to P_i.

Step 3: Security checking. After receiving S_i', P_i sends acknowledgements to TP. TP announces the insertion positions and the bases of the decoy photons in S_i' to P_i. Then P_i measures the decoy photons and returns the measurement results to TP. Afterward, TP checks whether eavesdroppers exist in the quantum channels according to their measuring results. If the error rate is higher than the threshold determined by the channel noise, TP cancels this protocol and restarts it. Otherwise, TP proceeds to the next step.

Step 4: Encoding and measurement. P_i extracts all the decoy photons from D_i and discards them, then he encodes his data. Concretely, P_i establishes a variable $v_i'^j$ and sets $v_i'^j = x_i^j$, and then he generates the d-level Bell state

$$\left|\Psi(v_i^j, v_i'^j)\right\rangle = \frac{1}{\sqrt{d}} \sum_{t=0}^{d-1} \zeta^{t v_i^j}\,|t\rangle\left|t \oplus v_i'^j\right\rangle \tag{12}$$

according to Eq. (5). Finally, P_i performs the d-level Bell state measurement on the particle with the mark u_i^j from the cat state and the particle with the mark v_i^j from his Bell state, and then ESCB happens (please see Fig. 2 for clarity).

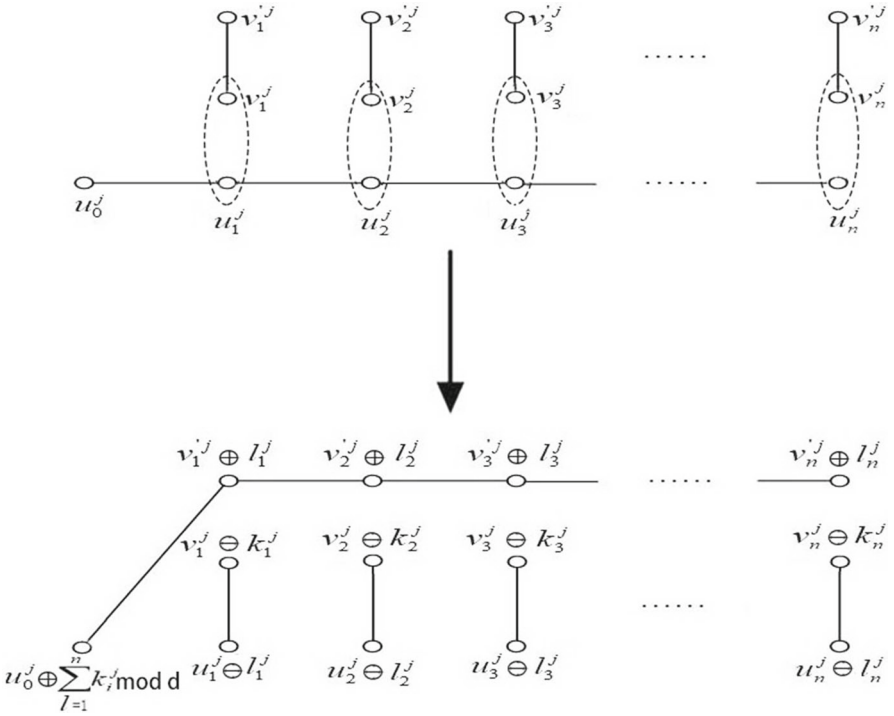

Fig. 2 The graphical description of entanglement swapping process of Protocol I

Step 5: Joint computation. All parties cooperate together to compute

$$S_L^j = \sum_{i=1}^{n} l_i^j, \tag{13}$$

then they announce S_L^j to TP.

Step 6: TP's measurement and computation. At this stage, all the cat states are sent back to TP. Please note that the security checking steps are adopted to guarantee the security of quantum transmissions, all of which are the same as those of Step 3. After security checking, TP measures his cat states and obtains the marks

$$\left(u_0^j + \sum_{i=1}^{n} k_i^j\right) \bmod d, v_1'^j \oplus l_1^j, v_2'^j \oplus l_2^j, \ldots, v_n'^j \oplus l_n^j. \tag{14}$$

Then he can compute

$$S_C^j = \sum_{i=1}^{n} \left(v_i'^j \oplus l_i^j\right). \tag{15}$$

By deducting S_L^j from S_C^j, TP can get

$$\sum_{i=1}^{n} v_i'^j. \tag{16}$$

Then TP gets the summation results

$$\sum_{i=1}^{n} x_i^1, \sum_{i=1}^{n} x_i^2, \ldots, \sum_{i=1}^{n} x_i^L. \tag{17}$$

Finally, TP announces the summation results to P_1, P_2, ..., P_n.

3.1 The success probability of Protocol I

Now let us analyze the success probability of Protocol I. In Step 6, TP measures his cat states and obtains the labels

$$v_1'^j \oplus l_1^j, v_2'^j \oplus l_2^j, \ldots, v_n'^j \oplus l_n^j, \tag{18}$$

where $j = 1, 2, \ldots, L$. Then he computes

$$\sum_{i=1}^{n} \left(v_i'^j \oplus l_i^j\right) - \sum_{i=1}^{n} l_i^j = \sum_{i=1}^{n} v_i'^j = \sum_{i=1}^{n} x_i^j. \tag{19}$$

TP will obtain the correct summation results iff $v_i'^j \oplus l_i^j \geq v_i'^j$ holds for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, L$, it means this protocol will output the correct summation results successfully under this condition. Otherwise, this protocol will fail, that is, at least one inequation $v_i'^j \oplus l_i^j < v_i'^j$ holds.

Let A_i^j be the event: $v_i'^j \oplus l_i^j \geq v_i'^j$ $(j = 1, 2, \ldots, L)$, and let B^j be the event: $v_i'^j \oplus l_i^j \geq v_i'^j$ holds for $i = 1, 2, \ldots, n$ (i.e., the summation result $\sum_{i=1}^{n} x_i^j$ is correct). In step 4, all parties perform the d-level Bell state measurement on the particles of the cat states and Bell states, without loss of generality, we assume that the order of their measurements is $P_1 \to P_2 \to P_3 \to \cdots \to P_n$. A rigorous discussion of the measurements of this protocol is, however, beyond the scope of the present work. For more details, please see Ref. [18].

Now let us first compute the probability of the event B^j, denote it by $P(B^j)$, we have

$$\begin{aligned} P(B^j) &= P(A_1^j A_2^j \cdots A_n^j) \\ &= P(A_1^j) \cdot P(A_2^j | A_1^j) \cdot P(A_3^j | A_1^j A_2^j) \cdot \cdots \cdot P(A_n^j | A_1^j A_2^j A_3^j \cdots A_{n-1}^j) \\ &= P(A_1^j) \cdot P(A_2^j) \cdot P(A_3^j) \cdot \cdots \cdot P(A_n^j) \\ &= \frac{d - x_1^j}{d} \cdot \frac{d - x_2^j}{d} \cdot \frac{d - x_3^j}{d} \cdot \cdots \cdot \frac{d - x_n^j}{d} \\ &= \prod_{i=1}^{n} \frac{d - x_i^j}{d} \quad , j = 1, 2, \ldots, L, \end{aligned} \tag{20}$$

where $P(A_2^j|A_1^j)$, $P(A_3^j|A_1^jA_2^j)$, ..., $P(A_n^j|A_1^jA_2^jA_3^j\cdots A_{n-1}^j)$ refer to conditional probabilities. Then we can easily compute the success probability of this protocol, denote it by P_I, we have

$$\begin{aligned} P_I &= P(B^1B^2B^3\cdots B^L) \\ &= P(B^1)\cdot P(B^2)\cdot P(B^3)\cdot\cdots\cdot P(B^L) \\ &= \prod_{j=1}^{L}\prod_{i=1}^{n}\frac{d-x_i^j}{d}. \end{aligned} \tag{21}$$

Define a function $f(y)$ with a variable y as follows

$$f(y) = \prod_{j=1}^{L}\prod_{i=1}^{n}\frac{y-x_i^j}{y}, \quad y\in(\sup\{S\},+\infty), \tag{22}$$

here please note that n, x_i^j $(i=1,2,\ldots,n,\ j=1,2,\ldots,L)$ are all constants based on our previous *assumptions*. Next we show that $f(y)$ is an increasing function of y. Define a new function

$$g(x) = \frac{x-c}{x}, \quad x\in(c,+\infty), \tag{23}$$

here we assume c is any positive integer. In fact, we only need to show $g(x)$ is an increasing function of x in order to show $f(y)$ is an increasing function of y. Taking the first derivative of $g(x)$, we have

$$g'(x) = \frac{c}{x^2}. \tag{24}$$

Obviously $g'(x)>0$, it means $g(x)$ is an increasing function of x, $f(y)$ is an increasing function of y accordingly. That is to say, the success probability P_I increases with the increase in d, and we have

$$\lim_{d\to+\infty} P_I = \lim_{d\to+\infty}\prod_{j=1}^{L}\prod_{i=1}^{n}\frac{d-x_i^j}{d} = 1. \tag{25}$$

that is, P_I will get closer and closer to 1 with the increase in d. Therefore, we can set the value of d to an enormous positive integer in order to increase the success probability of Protocol I.

3.2 A simple application: multi-party quantum private comparison

Multi-party quantum private comparison (MQPC) aims to compare the equality of the private information of multiple mutually distrustful parties. Suppose that there are n parties labeled $P_1, P_2, \ldots, P_n$, where $P_i(i=1,2,...,n)$ has a secret x_i. They want to judge whether all of their secrets are identical with the assistance of the semi-honest

TP. In order to complete this task, x_i need to be converted into its binary equivalent. We have showed that Protocol I can be used to complete the task of MQPC in our recent work [17], which is just a simple application of Protocol I.

4 Protocol II: quantum secure multi-party single-datum summation protocol

Assumptions: In this protocol, P_i $(i = 1, 2, \ldots, n)$ just has a secret number x_i where $x_i \in \{0, 1, 2, \ldots, M\}$ (Obviously, $M \geq \sup\{x_1, x_2, \ldots, x_n\}$). They can get the summation result $\sum_{i=1}^{n} x_i$ with the assistance of TP, while their secret numbers should be kept secret. It is worth pointing that the value of M can be set based on the actual situations, just as the example cited in the introduction (i.e., Sect. 1), we can set the value of M to a positive integer greater than or equal to 100 if the full mark is 100 for those examination courses.

For the sake of simplicity, we would like to briefly describe this protocol, and we intentionally left out a few processes.

Step 1: Prepare quantum states. P_i prepares $M + 1$ copies of d-level Bell state

$$|\Psi(0,0)\rangle = \frac{1}{\sqrt{d}} \sum_{t=0}^{d-1} |t\rangle\,|t\rangle\,, \tag{26}$$

where d is a positive integer greater than 1, and marks them by

$$|\Psi(0,0)\rangle_0\,, |\Psi(0,0)\rangle_1\,, \ldots, |\Psi(0,0)\rangle_M\,, \tag{27}$$

in turn to generate an ordered sequence, where the subscripts denote the order of the Bell states in the sequence.

TP prepares $M+1$ copies of d-level $(n+1)$-particle cat states $|\Psi(u_0, u_1, \ldots, u_n)\rangle$, and marks them by

$$\left|\Psi(u_0^0, u_1^0, \ldots, u_n^0)\right\rangle, \left|\Psi(u_0^1, u_1^1, \ldots, u_n^1)\right\rangle, \ldots, \left|\Psi(u_0^M, u_1^M, \ldots, u_n^M)\right\rangle, \tag{28}$$

in turn to generate an ordered sequence, where the superscripts denote the order of the cat states in the sequence. Then, TP takes the particles with marks $(u_1^j, u_2^j, \ldots, u_n^j)$ $(j = 0, 1, \ldots, M)$ out from each cat state $\left|\Psi(u_0^j, u_1^j, u_2^j, \ldots, u_n^j)\right\rangle$ to construct new sequences, and marks them by

$$(u_1^0, u_1^1, \ldots, u_1^M), (u_2^0, u_2^1, \ldots, u_2^M), \ldots, (u_n^0, u_n^1, \ldots, u_n^M), \tag{29}$$

in turn, and denotes them by $S_1, S_2, \ldots, S_n$, respectively. Subsequently, TP announces the marks $(u_i^0, u_i^1, \ldots, u_i^M)$ to P_i.

Step 2: Distribution. TP prepares n sets of decoy photons where each decoy photon randomly is in one of the states from the set V_1 or V_2. Then he inserts the n sets of

decoy photons into $S_1, S_2, \ldots, S_n$, respectively, at random positions and records their insertion positions. Then TP sends the new sequences to $P_1, P_2, \ldots, P_n$, respectively.

Step 3: Security checking. TP and P_i check the security of the transmission of S_i. If the error rate is higher than the threshold determined by the channel noise, TP cancels this protocol and restarts it. Otherwise, he proceeds to the next step.

Step 4: Encoding and measurement. After discarding the decoy photons, P_i encodes his secret number x_i. Concretely, P_i chooses the d-level Bell state with the subscripts x_i, i.e., $|\Psi(0,0)\rangle_{x_i}$, then he establishes a variable $v_i'^{x_i}$ and sets $v_i'^{x_i} = 1$. For the rest of d-level Bell states, P_i sets $v_i'^{k} = 0$ ($k = 0, 1, \ldots, x_i - 1, x_i + 1, \ldots, M$). Then he generates the d-level Bell state

$$\left|\Psi(v_i^j, v_i'^j)\right\rangle = \frac{1}{\sqrt{d}} \sum_{t=0}^{d-1} \zeta^{t v_i^j} |t\rangle \left|t \oplus v_i'^j\right\rangle, \tag{30}$$

according to Eq. (5), where $j = 0, 1, \ldots, M$. Subsequently, P_i performs the d-level Bell state measurement on the particle with the mark u_i^j from the cat state and the particle with the mark v_i^j from his Bell state.

Step 5: Joint computation. For $j = 0, 1, \ldots, M$: All parties cooperate together to compute

$$S_L^j = \sum_{i=1}^{n} l_i^j, \tag{31}$$

then they announce S_L^j to TP.

Step 6: TP's measurement and computation. For $j = 0, 1, \ldots, M$:

After TP has received all cat states, he measures each cat state and obtains the marks

$$\left(u_0^j + \sum_{i=1}^{n} k_i^j\right) \bmod d, v_1'^j \oplus l_1^j, v_2'^j \oplus l_2^j, \ldots, v_n'^j \oplus l_n^j. \tag{32}$$

After the similar computations (please see Protocol I), TP can get $\sum_{i=1}^{n} v_i'^j$. Then TP can compute the summation of all parties' secret numbers, that is

$$\sum_{i=1}^{n} x_i = \sum_{j=0}^{M} j \cdot \sum_{i=1}^{n} v_i'^j. \tag{33}$$

Finally, TP announces the summation result to $P_1, P_2, \ldots, P_n$.

4.1 The success probability of Protocol II

Similar to the analysis in Sect. 3.1, we can compute the success probability of this protocol, denoted by P_{II}, we have

$$P_{II} = \prod_{j=0}^{M} \prod_{i=1}^{n} \frac{d - v_i'^j}{d} = \left(\frac{d-1}{d}\right)^n, \tag{34}$$

where $v_i'^j \in \{0, 1\}$ $(i = 1, 2, \ldots, n, j = 0, 1, \ldots., M)$ (please see step 4). Define a function $f(x)$ with a variable x as follows

$$f(x) = \left(\frac{x-1}{x}\right)^n, \quad x \in (1, +\infty), \tag{35}$$

we have

$$f'(x) = \frac{n(x-1)^{n-1}}{x^{n+1}} > 0, \tag{36}$$

which means $f(x)$ is an increasing function of x. Accordingly, the success probability P_{II} increases with the increase in d, and we have

$$\lim_{d \to +\infty} P_{II} = \lim_{d \to +\infty} \left(\frac{d-1}{d}\right)^n = 1, \tag{37}$$

which means P_{II} will get closer and closer to 1 with the increase in d. Therefore, we can also set the value of d to an enormous positive integer in order to increase the success probability of Protocol II.

4.2 Generalization to quantum anonymous ranking

Quantum anonymous ranking (QAR) [10] aims to rank multiple parties' data anonymously, where each party should know the position of his secret number in the ascending (or descending) sequence of all the ranked numbers at the end of the protocol.

Suppose that there are n parties labeled $P_1, P_2, \ldots, P_n$, where each party P_i $(i = 1, 2, \ldots, n)$ has a secret number x_i $(x_i \in \{1, 2, \ldots, L\})$. P_i can correctly get the ranking of x_i, that is, the position of x_i in the ascending (or descending) ranking sequence of all the numbers $x_1, x_2, \ldots, x_n$ with the assistance of a semi-honest third party (TP). For $j = 1, 2, \ldots, L$, TP computes $\sum_{i=1}^{n} v_i'^j$ and announces it to $P_1, P_2, \ldots, P_n$. P_i knows the ranking of his data x_i is:

$$\sum_{j=1}^{x_i - 1} \sum_{i=1}^{n} v_i'^j + 1. \tag{38}$$

4.3 Generalization to quantum anonymous voting

Assume that there are n voters labeled $V_1, V_2, \ldots, V_n$ and L candidates labeled $C_1, C_2, ..., C_L$, where each voter has one vote. If a voter want to vote for a candidate,

he votes "yes," for the others he votes "no." The tallyman is in charge of counting the votes of C_j ($j = 1, 2, \ldots, L$) by computing $\sum_{i=1}^{n} v_i'^j$ at the end of the protocol.

The identities of all voters are kept private since the sum $\sum_{i=1}^{n} v_i'^j$ contains no information about who voted how, although it is made public by the tallyman. In addition, at the end of the protocol, the tallyman computes

$$S = \sum_{j=1}^{L} \sum_{i=1}^{n} v_i'^j . \tag{39}$$

If $S > n$, the tallyman can conclude that at least one voter cheats by making more than m votes, and then he cancels this protocol and restarts it. Therefore, this protocol can prevent voters from cheating.

5 Security analysis

In this section, we will analyze the security of our protocols. Please note that we just analyze the security of Protocol I since the security of Protocol II is the same as that of Protocol I, and we assume that the quantum channels are authenticated in our protocols. We show that both the attacks from outside and the attacks from all participants are invalid to our protocols.

5.1 Outside attack

In Protocol I, we use the decoy photons to prevent the eavesdropping. This idea is derived from the BB84 QKD protocol [6], which has been proved unconditionally safe [19]. Any eavesdropping will be discovered in the security checking steps, thus outside Eve's all kinds of attacks, such as the intercept-resend attack, the measurement-resend attack, the entanglement-measurement attack and the denial-of-service (DOS) attack will be caught during the process of security checking. We take the intercept-resend attack as an example here: If an outside eavesdropper Eve attempts to intercept the particles sent from TP to P_i and replaces them with his own fake particles, he will introduce extra error rate which makes him be detected during the process of security checking since he does not know the exact position and the original state of the decoy photons. When we use m decoy particles for eavesdropping, exposed probability will be

$$1 - \left(\frac{d+1}{2d}\right)^m . \tag{40}$$

Therefore, Eve will be detected through security analysis.

5.2 Participant attack

In this section, we analyze the participant attack. To see it in a sufficient way, two cases should be considered: the participant attack from one or more dishonest parties, and the participant attack from TP.

Case 1: The participant attack from one or more dishonest parties

We consider two situations: One is one party wants to steal the secret numbers from others; the other is more than one party collude together to steal secret numbers from others.

(a) The participant attack from one dishonest party

Since n parties play the same roles in our protocols, without loss of generality, we just make analysis on the case of P_2 wants to steal the secret numbers of P_1 here.

In Protocol I, no qudit is exchanged between P_1 and P_2; thus, P_2 will be caught as an outside eavesdropper analyzed above if he tries to intercept the particles transmitted between TP and P_1, because he has no knowledge about the insertion positions and bases of the decoy photons.

In step 4, P_2 can learn l_2^j. In step 5, P_2 knows S_L^j. However, he cannot learn $v_1'^j$ since he has no way to learn $v_1'^j \oplus l_1^j$ from TP who is not allowed to conspire with any party.

Therefore, P_2 cannot obtain the secret numbers of P_1.

(b) The participant attack from more than one dishonest party

Here, we only consider the extreme case that there are $n-1$ parties colluding together to steal the left party's secret numbers, because $n-1$ parties have the most powerful strength. We assume that $n-1$ parties $P_1, P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ cooperate together to steal the secret numbers of P_i.

In step 4, P_m $(m = 1, 2, \ldots, i-1, i+1, \ldots, n)$ can learn l_m^j. In step 5, $P_1, P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ can learn l_i^j from S_L^j if they collude together. However, they still cannot learn $v_i'^j$ since they have no way to know $v_i'^j \oplus l_i^j$.

Therefore, $P_1, P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ cannot learn the secret number of P_i even if they collude together.

Case 2: The participant attack from TP

Since TP is assumed as a semi-honest third party in our protocol, he may try his best to steal the secret data of P_i without conspiring with any one. In step 5, TP receives S_L^j. In step 6, TP knows $v_i'^j \oplus l_i^j$. However, he still cannot learn $v_i'^j$ due to the lack of knowledge about l_i^j even though he knows S_L^j and $v_i'^j \oplus l_i^j$.

6 Discussion

6.1 The comparison between Protocol I and II

It seems to make sense that we point out the main differences between Protocol I and Protocol II. The first one is the processes of summation. In fact, Protocol I performs the sum of the data in the target dataset directly, while Protocol II computes the summation results of the same data in the target dataset first, and then performs the sum of these results. For example, suppose that there is a dataset {50, 60, 70, 80, 90, 60, 60, 80}, the process of the summation in Protocol I can be briefly described as "50 + 60 + 70 + 80 + 90 + 60 + 60 + 80 = 550," while it can be briefly described as "50 × 1 + 60 ×

$3 + 70 \times 1 + 80 \times 2 + 90 \times 1 = 550$" in Protocol II. We will show the advantages of this algorithm adopted by Protocol II in the following text.

In Protocol I, we set the value of d to a positive integer greater than the maximum element in D ($D = D_{P_1} \cup D_{P_2} \cup \cdots \cup D_{P_n}$), while we set its value to a positive integer greater than 1 in Protocol II. We have showed that the higher the value of d, the higher the success probability for both of our protocols.

In what follows, consider the following question: If one wants to choose Protocol II to complete the task of Protocol I, what is the success probability? To answer this question, we would like to assume that all parties' data are a set of positive integers, and ensure the value of d big enough (at least meets the condition of $d > \sup\{S\}$, please see the *assumptions* in Protocol I). Easily, we can compute the success probability, denote it by P,

$$P = \left(\frac{d-1}{d}\right)^{nL}. \tag{41}$$

As for P_I (please see Eq. (21)), let us define a new function $f(x)$ with a variable x as follows

$$f(x) = \frac{d-x}{d}, \quad x \in (1, +\infty). \tag{42}$$

Obviously, $f(x)$ is a decreasing function of x, and thus $\frac{d-x_i^j}{d}$ decreases with the increase in x_i^j, and we have

$$\frac{d-x_i^j}{d} \leq \frac{d-1}{d}. \tag{43}$$

Furthermore, we have

$$\prod_{j=1}^{L}\prod_{i=1}^{n}\frac{d-x_i^j}{d} \leq \left(\frac{d-1}{d}\right)^{nL}, \tag{44}$$

the equalities in Eqs. (42) and (43) hold iff $x_i^j = 1$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, L$. From Eqs. (21) to (40), we have

$$P_I \leq P. \tag{45}$$

That is to say, P is an upper bound of P_I.

It is worthy pointing that in order to complete this summation task, $L \times (M+1)$ copies of d-level $(n+1)$-particle cat state and $n \times L \times (M+1)$ copies of d-level Bell state are needed in Protocol II, while L copies of the cat state and $n \times L$ copies of the Bell state are needed in Protocol I.

Consequently, we give preference to choose Protocol I to complete this summation task from the point of view of the consumption of quantum resources, while Protocol

Table 1 The comparison results of Protocol I and II

Protocol	I	II
The selection of d	$d > Max\{S\}$	$d > 1$
Algorithm	Computing the summation directly	First computing the summation results of the same data, then computing the summation of these results
Consumption of quantum resources	L cat states and $n \times L$ Bell states	$L \times (M+1)$ cat states and $n \times L \times (M+1)$ Bell states
Success probability	$\prod_{j=1}^{L} \prod_{i=1}^{n} \frac{d-x_i^j}{d}$	$\left(\frac{d-1}{d}\right)^{nL}$
Different applications	MQPC	QAV and QAV

II maybe the better choice from the point of view of the success probability. Besides, we have showed that two algorithms adopted in our two protocols can be used for completing different tasks of QSMC, respectively, in previous sections. Finally, we show these comparison results of Protocol I and II in Table 1 for the sake of clarity.

6.2 The comparison between the proposed protocols and previous protocols

The first several QSMS algorithms were proposed in quantum anonymous voting and surveying protocols in order to calculate the summation of some private data including binary numbers and integers [14–16]. In 2005, Hillery et al. proposed a QSMS algorithm in their quantum anonymous voting protocol, in which each number is 0 or 1 [15,16]. In their protocol, the tallyman who is responsible for preparing and distributing the required quantum state is assumed to follow the protocol, but does whatever he can beyond this to learn the individual votes. Each party (voter) makes a vote through performing one of the two different operations on his(her) qubits. In 2007, Vaccaro et al. proposed another QSMS algorithm for calculating the summation of N integers where N denotes the number of parties (voters) in their quantum anonymous surveying protocol, in which a $(N+1)$-particle entangled state is employed [14]. Each of these parties makes a vote by performing a phase-shifting operation on their respective particles. Then all the parties send their particles to the tallyman who is responsible for counting the votes (i.e., calculating the summation of N integers). Recently, we presented a QSMS algorithm in our proposed multi-party quantum private comparison (MQPC) protocol [17]. In the MQPC protocol, similar to the QSMS algorithms proposed in quantum voting protocols [15,16], each number required for the summation is limited to be 0 or 1. The third party (TP) is assumed to be semi-honest who may misbehave on his own, but cannot conspire with any party.

We further develop our previous work (Ref. [17]) for securely computing the summation of a set of nonnegative integers in this paper. We first propose a QSMS protocol (i.e., Protocol I) with the following improvements: (1) All secret data are nonnegative integers rather than binary numbers adopted by Ref. [17]. (2) The algorithm is simpler than that of Ref. [17] (please see step 5 and 6 in Protocol I). (3) The success

Table 2 The comparison among Protocol I and Refs. [14,15,17]

Protocol	Ref. [14]	Ref. [15]	Ref. [17]	Protocol I
Quantum states used	d-level n-particle entangled states	d-level n-particle entangled states	d-level $(n+1)$-particle cat states and d-level Bell states	d-level $(n+1)$-particle cat states and d-level Bell states
Quantum technology adopted	Unitary operations	Phase-shifting operation	Unitary operations and entanglement swapping	Unitary operations and entanglement swapping
The data type for summation	Binary number	Integer	Binary number	Nonnegative integer

probability of the protocols is analyzed, which was neglected in Ref. [17]. We then propose a new QSMS protocol (i.e., Protocol II) inspired by Ref. [14] (the comparison between Protocol I and II is given in Sect. 6.1). For clarity, we compare Protocol I with Refs. [14–17], and show the comparison results in Table 2. (The symbol n in the table denotes the number of voters.)

7 Conclusion

In this paper, we present two QSMS protocols, both of which allow multiple mutually distrustful parties to securely compute the summation of their secret data, where the dataset of the summation is supposed to be a set of nonnegative integers. Our protocols use ESCB to maintain privacy, and pass information between each party and TP. In each protocol, all parties employ unitary operations to encode their secret data, and with the help of the semi-honest TP they can obtain the summation result at the end of the protocol. We generalize the use of our second protocol to execute the tasks of QAR and QAV. We also show that our protocols can withstand both the outside attacks and the participant attacks. It is worthy pointing that our protocols can withstand the extreme collusion attack, that is, if $n-1$ parties conspire to steal the left party's secret numbers, they will not succeed, and hence our protocols have high security.

We wish that our algorithms adopted in our two protocols may find more applications in designing protocols for QSMC, and we will make more efforts.

Acknowledgements The authors would like to thank the anonymous reviewers for their important discussions. This work is partially supported by the State Key Program of National Natural Science of China No. 61332019, the Major State Basic Research Development Program of China (973 Program) No. 2014CB340601, the National Science Foundation of China No. 61202386, 61402339, the National Cryptography Development Fund No. MMJJ201701304, the Science and Technology Research Project of Hebei higher education No. QN2017020.

References

1. Zhang, H.G., Han, W.B., Lai, X.J., et al.: Survey on cyberspace security. Sci. China Inf. Sci. **58**(11), 1–43 (2015)
2. Yao, A.C.: Protocols for secure computations. Foundations of Computer Science. In: SFCS'08. 23rd Annual Symposium on. IEEE **1982**, 160–164 (1982)
3. Wu, W.Q., Zhang, H.G.: Quantum algorithm to solve function inversion with timespace trade-off. Quantum Inf. Process. **16**(7), 171 (2017)
4. Wu, W.Q., Zhang, H.G., Mao, S.W., Wang, H.Z.: Quantum algorithm to find invariant linear structure of MD hash functions. Quantum Inf. Process. **14**(3), 813–829 (2015)
5. Wu, W.Q., Zhang, H.G., Wang, H.Z., Mao, S.W.: Polynomial-time quantum algorithms for finding the linear structures of Boolean function. Quantum Inf. Process. **14**(4), 1215–1226 (2015)
6. Bennett, C.H., Brassard, G.: Quantum cryptography: public-key distribution and coin tossing. In: Proceeding of the IEEE International Conference on Computers, Systems and Signal, pp. 175179. Bangalore, India (1984)
7. Hillery, M., Bužek, V., Berthiaume, A.: Quantum secret sharing. Phys. Rev. **59**(3), 1829 (1999)
8. Karimipour, V., Bahraminasab, A., Bagherinezhad, S.: Entanglement swapping of generalized cat states and secret sharing. Physical Review A **65**, 042320 (2002)
9. Bagherinezhad, S., Karimipour, V.: Quantum secret sharing based on reusable Greenberger–Horne–Zeilinger states as secure carriers. Phys. Rev. A **67**(4), 044302 (2003)

10. Huang, W., Wen, Q.Y., Liu, B., et al.: Quantum anonymous ranking. Phys. Rev. A **89**(3), 032325 (2014)
11. Lin, S., Guo, G.D., Huang, F., et al.: Quantum anonymous ranking based on the Chinese remainder theorem. Phys. Rev. A **93**(1), 012318 (2016)
12. Clifton, C., Kantarcioglu, M., Vaidya, J., et al.: Tools for privacy preserving distributed data mining. ACM Sigkdd Explor. Newsl. **4**(2), 28–34 (2002)
13. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications: a review and open problems. In: Proceedings of the 2001 Workshop on New Security Paradigms. ACM, 13–22 (2001)
14. Vaccaro, J.A., Spring, J., Chefles, A.: Quantum protocols for anonymous voting and surveying. Phys. Rev. A **75**(1), 012333 (2007)
15. Hillery, M., Ziman, M., Bužek, V., et al.: Towards quantum-based privacy and voting. Physics Letters A **349**(1), 75–81 (2006)
16. Bonanome, M., Bužek, V., Hillery, M., et al.: Toward protocols for quantum-ensured privacy and secure voting. Phys. Rev. A **84**(2), 022331 (2011)
17. Ji, Z.X., Ye, T.Y.: Multi-party quantum private comparison based on the entanglement swapping of d-level cat states and d-level Bell states. Quantum Inf. Process. **16**(7), 177 (2017)
18. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
19. Shor, P.W., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Phys. Rev. Lett. **85**(2), 441 (2000)