Verifiable blind quantum computation with identity authentication for different types of clients

Junyu Quan¹, Qin Li² and Lvzhou Li³

¹School of Mathematics and Computational Science, Xiangtan University, Xiangtan, 411105, China.
²School of Computer Science, Xiangtan University, Xiangtan, 411105, China.
³Institute of Quantum Computing and Computer Theory, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China.

Abstract
Quantum computing has considerable advantages in solving some problems over its classical counterpart. Currently various physical systems are developed to construct quantum computers but it is still challenging and the first use of quantum computers may adopt the cloud style. Blind quantum computing (BQC) provides a solution for clients with limited quantum capabilities to delegate their quantum computation to remote quantum servers while keeping input, output, and even algorithm private. In this paper, we propose three multi-party verifiable blind quantum computing (VBQC) protocols with identity authentication to handle clients with varying quantum capabilities in quantum networks, such as those who can just make measurements, prepare single qubits, or perform a few single-qubit gates. They are client-friendly and flexible since the clients can achieve BQC depending on their own quantum devices and resist both insider outsider attacks in quantum networks. Furthermore, all the three proposed protocols are verifiable, namely that the clients can verify the correctness of their calculations.

Keywords: verifiable blind quantum computation, multi-party quantum computation, quantum network, quantum identity authentication
1 Introduction

Quantum computation is a revolutionary computer model that has the potential to provide significant computational benefits over traditional technologies by utilizing principles of quantum mechanics. For instance, Shor’s algorithm can solve the integer factorization problem in polynomial time [1], whereas the best existing classical algorithm requires super-polynomial time. Even though quantum computing has several benefits and numerous physical systems such as trapped ions [2, 3], superconducting qubits [4, 5], photons [6, 7], and silicon [8, 9] have been investigated for building quantum computers, it remains a challenge to construct large-scale quantum computers. In the near future, it is likely that only a few organizations will own quantum computers and other clients with limited quantum capabilities have to delegate their quantum computational tasks to quantum servers. Blind quantum computing (BQC) could make clients interact securely with a remote quantum server, that is, BQC enables a client with insufficient quantum capability to delegate her quantum computation to a remote quantum server while keeping input, output, and even algorithm private.

Childs presented the first BQC protocol based on the circuit model in 2005 [10], where the client requires quantum memory and the ability to rearrange qubits and perform SWAP gates. Then, Arrighi and Salvail introduced another BQC protocol that only requires Alice to generate and measure entangled states [11]. But it only provides both verification and blindness for certain special functions. In 2009, Broadbent, Fitzsimons, and Kashefi presented the first universal BQC protocol based on the measurement model where the client just prepares random single qubits [12], and it is called the BFK protocol. Furthermore, through the use of four-qubit blind cluster states, the BFK protocol has been physically realized by Barz et al. [13]. In addition, the single-server BFK protocol is expanded to double-server [14, 15], triple-server [16], and multiple-server modes [17]. In the double-server BQC protocols proposed in Refs. [14, 15], the client can be completely classical if the two delegated quantum servers are supposed to be non-communicating. Then Li et al. proposed a triple-server BQC protocol where communication is allowed among servers via entanglement swapping [16]. Different from the BFK protocol, Morimae and Fujii proposed a BQC protocol called the MF protocol based on a new model [18] in which the client only makes measurements. This protocol has also been demonstrated by using four-qubit cluster states [19]. Since then, numerous BQC protocols based on various methods were developed [20–25].

Almost all the proposed BQC protocols should satisfy two important properties, namely blindness and correctness. Blindness prevents the delegated quantum server from obtaining any valuable information about the client’s input, output, and algorithm, while correctness means the client could verify whether the computation results are right, which can be ensured by verifiable blind quantum computation (VBQC) [26–30]. Currently, VBQC protocols are mainly divided into two types, namely stabilizer test verification [29–31]
and trap-based verification [26]. For the first type, the server generates multiple copies of graph states and sends each qubit of them one by one to the client. Then the client verifies the stabilizer operators of these graph states to determine whether the server has constructed correct graph states. Trap-based verification, on the other hand, requires a client generating single-qubit states and embedding trap qubits to check the server’s behaviour. If the server does not follow the protocol, he may disturb the trap qubits which could be detected by the client with a high probability.

In addition, we are moving into the noisy intermediate-scale quantum (NISQ) era now [32], which means one can control a quantum system with qubits from 50 to a few hundred [33, 34]. More quantum nodes are expected to arise in quantum networks as a result of the development of quantum computing [35–37]. In such a complex network environment, the classical clients who want to perform BQC will face more problems compared with that in a single environment. For example, in a multi-node quantum network, outsider attacks from malicious third parties during transmission and effective authentication between individual nodes need to be considered; due to the decoherence caused by noise, clients need to verify the correctness of their calculation results even if the server is honest; and how different clients with various quantum devices to perform BQC securely also need consideration.

Recently, Li et al. first introduced identity authentication to BQC for resisting the man-in-the-middle attack and denial-of-service attack [38]. Then Shan et al. presented a BQC protocol with mutual authentication based on quantum network [39], which includes registration phase, mutual identity authentication phase, and multi-party blind quantum computation phase. In this protocol, there are numerous clients and servers. Each legitimate client needs to establish a registration key with a specific server through certificate authority (CA) and load balancers in the registration phase. After the client and the corresponding server authenticate each other with the registration key, she can delegate her quantum computation to the server. However, only the clients who are limited to the capability of making measurements are considered in this protocol. In fact, clients in various BQC protocols have different quantum capabilities and are suitable for special physical systems. In an optical system, for instance, measuring a single-qubit state is easier than generating a single qubit [18], hence the MF protocol is more favorable to clients who possess measurement devices in this situation. In superconducting systems, performing single-qubit gates is more precise than preparing or measuring single-qubit states [40]. Hence the protocol in which clients only perform a few quantum gates [24] is better for superconducting system. Therefore, it is necessary to consider how to enable clients to delegate their quantum computation securely depending on their different quantum capabilities in a quantum network. Furthermore, in the BQC protocol proposed in Ref. [39], the registration key shared by the client and the corresponding server could be obtained by the load balancers, and it is unrealistic that both the servers and the load balancers are expected
to be trustworthy. Moreover, the clients cannot verify the correctness of their computation.

Considering all the above, in this paper we propose three VBQC protocols with identity authentication for clients with different capabilities in the quantum network as shown in Table 1. Especially, in the proposed protocols, even if CA, load balancers, and servers are regarded as semi-honest, the clients can verify their computation successfully.

| Protocols | Participants | The quantum capability of clients | Verifiability | Identity authentication |
|-----------|--------------|----------------------------------|---------------|------------------------|
| RAM-SMVBQC in Sec. 4.1 | Semi-honest CA, load balancers, and servers | Measuring single-qubit states | Yes | Yes |
| PAS-SMVBQC in Sec. 4.2 | Semi-honest CA, load balancers, and servers | Preparing single-qubit states | Yes | Yes |
| CB-SMVBQC in Sec. 4.3 | Semi-honest CA, load balancers, and servers | Implementing $H$ and $\sigma_z^{1/4}$ gates | Yes | Yes |

The rest part of the paper is organized as follows. Section 2 introduces two typical verification methods in VBQC protocols. Section 3 briefly reviews the BQC protocol proposed by Shan et al. [39]. Section 4 proposes three VBQC protocols. Section 5 compares the proposed protocols with the similar ones in the literature. The last section 6 makes a conclusion.

2 Preliminaries

In this part, the stabilizer test and trap-based verification which are two important verification methods are introduced.

2.1 Stabilizer test

Standard graph states and related methods for the stabilizer test [29–31] are introduced briefly as follows. An $n$-qubit graph state $G \equiv (V, E)$ corresponding to graph $G$ is defined as

$$|G\rangle \equiv (\bigotimes_{e \in E} CZ_e)|+\rangle^\otimes n$$

(1)

where $V$ is a set of $n$ vertices, $E$ is a set of edges, $n \equiv |V|$, $e \in E$, $|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and $CZ_e$ is the Controlled-Z gate ($CZ \equiv |0\rangle\langle 0| \bigotimes I + |1\rangle\langle 1| \bigotimes Z$) acting on the pair of vertices sharing the edge $e$. Then the description of $i$th stabilizer $g_i$ ($1 \leq i \leq n$) of $|G\rangle$ is given as follows
\[ g_i \equiv (\prod_{e \in E} CZ_e)X_i(\prod_{e \in E} CZ_e^\dagger) = X_i \prod_{V_j \in N(i)} Z_j \] (2)

where \( N(i) \) is the set of neighbors of the \( i \)-th qubit. About the stabilizer test for \( g_i \) on an \( n \)-qubit quantum state \( \rho \), Alice measures \( X_i \) and \( Z_j \) for all \( j \in N(i) \) and the measurement results are recorded as \( x_i \in \{0, 1\} \) and \( z_j \in \{0, 1\} \). If \( x_i + \sum_{j \in N(i)} z_j \equiv 0 \mod 2 \), (3)

Alice will pass the stabilizer test, which means that the state \( \rho \) is close to the ideal state \( |G\rangle \). Otherwise, Alice rejects it as \( \rho \) is far from \( |G\rangle \). Obviously, we can get the acceptance probability satisfies \( (1 + \langle G|\rho|G\rangle)/2 \). If Bob is honest, he will generate the correct graph state which always satisfies Eq. (1) and Alice will pass the stabilizer test for \( g_i \) with unit probability for all \( i \).

### 2.2 Trap-based verification

The trap-based verification was introduced in the verifiable universal blind quantum computing (VUBQC) protocol by Fitzsimons and Kashefi [26]. This protocol can achieve blindness with the method in UBQC [12] and accomplish verifiability through embedding trap qubits.

In UBQC, Bob produces a universal graph state \( |G\rangle \) by using \( n \) qubits \( |+_{\theta_i}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta_i}|1\rangle) \) sent by Alice, where \( \theta_i \) is chosen from the set \{0, \pi/4, 2\pi/4, ..., 7\pi/4\}. Then, Alice requests Bob to measure the \( i \)-th qubit of \( |G\rangle \) in the basis \{\( |\pm_{\delta_i}\rangle \)\}, where \( \delta_i = \phi'_i + \theta_i + r_i \pi \), \( \phi'_i \) is determined by previous measurements and \( r_i \) is chosen from the set \{0, 1\} at random. Note that, Bob also sends the measurement result \( b_i \) to Alice, which will be used to update the angles related to computation. Then Alice will obtain the output after Bob has completed all measurements. To verify the computation, Alice can prepare qubits in the state \( |z\rangle \), which are randomly selected from \{\( |0\rangle \), \( |1\rangle \)\} and called dummy qubits. According to Eq. (4) and Eq. (5),

\[ CZ(|0\rangle \otimes |+_{\theta}\rangle) = |0\rangle \otimes |0\rangle + |0\rangle \otimes e^{i\theta}|1\rangle \]

\[ = |0\rangle \otimes |+_{\theta}\rangle, \]

\[ CZ(|1\rangle \otimes |+_{\theta}\rangle) = |1\rangle \otimes |0\rangle - |1\rangle \otimes e^{i\theta}|1\rangle \]

\[ = |1\rangle \otimes |-\theta\rangle, \]

a dummy qubit \( |0\rangle \) or \( |1\rangle \) will not entangle with the rest of the qubits \( |+_{\theta_i}\rangle \) of the graph state and thus it will not influence the correctness or blindness of the
computation. Suppose that Alice owns a trap qubit $|+\theta\rangle$, she may employ some dummy qubits to isolate $|+\theta\rangle$ from other adjacent states used for computation. If the trap qubit is measured in the specified basis, then the measurement result is also certain. However, from Eq. (5), if a CZ gate is applied to a dummy qubit $|1\rangle$ and a trap qubit $|+\theta\rangle$, the trap qubit will flip to $|-\theta\rangle$, where the measurement result will be flipped as well. Thus, Alice can check Bob’s measurements and the entangling operations to verify the results by embedding the trap qubits and the dummy qubits.

3 Review of Shan et al.’s BQC protocol with mutual authentication [39]

An overview of a multi-party BQC protocol with mutual authentication proposed by Shan et al. is given as follows [39]. It involves $m$ clients $A_i$ $(1 \leq i \leq m)$, $n$ servers $B_j$ $(1 \leq j \leq n)$, LoadBalancerA and LoadBalancerB who allocates resources to clients and servers respectively, and a semi-honest certificate authority (CA). Note that the number of servers may be smaller than that of clients and then the server associated with client $A_i$ should be denoted by $B_{i \mod n}$. The entire protocol consists of three phases, namely the registration phase mainly for quantum key distribution, the mutual identity authentication phase, and the blind quantum computing phase. In the following, the specific steps of the protocol are given.

Phase 1: the registration phase for quantum key distribution

S1-1 LoadBalancerA and LoadBalancerB generate qubit sequences $S_A$ and $S_B$ randomly in X or Z basis and decoy qubits in $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ with suitable number. Then the decoy qubits are randomly inserted into $S_A$ and $S_B$. LoadBalancerA and LoadBalancerB transmit the new sequences $S_{A,\text{decoy}}$ and $S_{B,\text{decoy}}$ to the client $A_i$ and the corresponding server $B_{i \mod n}$.

S1-2 Once $A_i$ and $B_{i \mod n}$ received $S_{A,\text{decoy}}$ and $S_{B,\text{decoy}}$, respectively, they need to measure the decoy qubits. The positions and corresponding bases of them are informed by LoadBalancerA and LoadBalancerB. If the error rate is higher than the predefined threshold, an eavesdropper Eve is considered to exist and the protocol should restart again. Otherwise, LoadBalancerA and LoadBalancerB send $S_A$ and $S_B$ to CA.

S1-3 CA performs Bell measurements and records the measurement results as $R_{AB}$ when he receives both sequences $S_A$ and $S_B$. Note that the measurement results $|\phi^+\rangle$, $|\phi^-\rangle$, $|\psi^+\rangle$ and $|\psi^-\rangle$ are encoded as 00, 01, 10, 11, respectively. Then CA sends $R_{AB}$ to $A_i$ and $B_{i \mod n}$ through LoadBalancerA and LoadBalancerB across classical channels.

S1-4 When $A_i$ and $B_{i \mod n}$ receive $R_{AB}$, LoadBalancerA and LoadBalancerB announces the basis of each state in $S_A$ and $S_B$. According to Table 2, they keep the $j$-th bit under the same basis as the initial raw key bit $K_{AB}^j \in \{0, 1\}$ when $R_{AB}^j = 11$. 
VBQC with identity authentication for different types of clients

S1-5 \(B_{i \mod n}\) and \(A_i\) both select a portion of their raw keys to estimate the error rate and detect eavesdropping. If the error rate is higher than the threshold, \(A_i\) has to terminate the protocol and move to S1-1.

S1-6 The procedures above are repeated until each client has completed the registration and shared a key with a specific network server. If \(m > n\) and \(i > n\), \(B_{i \mod n}\) may store more than one key in his memories.

Table 2 Correlation of Non-orthogonal States Produced by CA and Measurement Results on \(S'_A\) and \(S'_B\):

| \(R_{A_i} \) | \(R_{B_{i \mod n}} \) | \(|0\rangle_A \rightarrow 0\) | \(|1\rangle_A \rightarrow 1\) | \(|+\rangle_A \rightarrow 0\) | \(|-\rangle_A \rightarrow 1\) |
|---|---|---|---|---|---|
| \(|\phi^-\rangle_{AB}\) | \(|\phi^-\rangle_{AB}\) | \(|\psi^+\rangle_{AB}\) | \(|\Psi^+\rangle_{AB}\) | \(|\Phi^-\rangle_{AB}\) |
| \(|\phi^+\rangle_{AB}\) | \(|\phi^+\rangle_{AB}\) | \(|\phi^-\rangle_{AB}\) | \(|\Phi^-\rangle_{AB}\) | \(|\Psi^+\rangle_{AB}\) |
| \(|\Phi^-\rangle_{AB}\) | \(|\Phi^-\rangle_{AB}\) | \(|\phi^+\rangle_{AB}\) | \(|\Psi^+\rangle_{AB}\) | \(|\phi^-\rangle_{AB}\) |

Phase 2: the mutual identity authentication phase

S2-1 If a registered client \(A_i\) wants to delegate a BQC task to a remote server, he needs to send a request \(i\) to Load_Balancer_A firstly.

S2-2 According to the first-in-first-out (FIFO) principle, Load_Balancer_A resends the request \(i\) to CA, who prepares 4\(k\) non-orthogonal states \(|\varphi\rangle_{AB}\) randomly chosen from \{\(|\varphi^-\rangle, |\psi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle\}\}, where

\[
|\phi^-\rangle_{AB} = \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle)_{AB},
\]

\[
|\psi^+\rangle_{AB} = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)_{AB},
\]

\[
|\Phi^-\rangle_{AB} = \frac{1}{\sqrt{2}} (|\phi^-\rangle - |\psi^+\rangle)_{AB}
\]

\[
= \frac{1}{\sqrt{2}} (|0-\rangle - |1+\rangle)_{AB} = \frac{1}{\sqrt{2}} (|1 + 1\rangle - | - 0\rangle)_{AB},
\]

\[
|\Psi^+\rangle_{AB} = \frac{1}{\sqrt{2}} (|\phi^-\rangle + |\psi^+\rangle)_{AB}
\]

\[
= \frac{1}{\sqrt{2}} (|0+\rangle - |1-\rangle)_{AB} = \frac{1}{\sqrt{2}} (| + 0\rangle - | - 1\rangle)_{AB}.
\]

All the first qubits of these non-orthogonal states form the qubit sequence \(S'_A\) and the second qubits of them form the qubit sequence \(S'_B\). CA sends \(S'_A\) and \(S'_B\) to Load_Balancer_A and Load_Balancer_B, respectively. Note that, CA also encodes the non-orthogonal states as \(|\phi^-\rangle \rightarrow 00, |\psi^+\rangle \rightarrow 01, |\Phi^-\rangle \rightarrow 10, |\Psi^+\rangle \rightarrow 11\).
|Ψ⟩ → 11, and broadcasts them to $A_i$ and $B_i \mod n$. When Load_Balancer_A and Load_Balancer_B receive $S'_A$ and $S'_B$, they generate some decoy qubits and insert them into $S'_A$ and $S'_B$. Then they send the new qubit sequences $S'_{A,\text{decoy}}$ and $S'_{B,\text{decoy}}$ to $A_i$ and $B_i \mod n$, respectively.

S2-3 $A_i$ and $B_i \mod n$ perform eavesdropping detection by checking whether the states of decoy qubits are changed. If the error rate is over a certain threshold, the step S2-2 should be re-executed; otherwise they can obtain $S'_A$ and $S'_B$.

S2-4 $A_i$ and $B_i \mod n$ measure $S'_A$ and $S'_B$ in the bases in terms of the initial key $K_{AB}$ and record the measurement results $R_{A_i}$ and $R_{B_i \mod n}$. Then $A_i$ and $B_i \mod n$ randomly reveal $3k$ values of $R_{A_i}$ and $R_{B_i \mod n}$ and the corresponding positions via classical channels.

S2-5 According to Table 3, $A_i$ and $B_i \mod n$ check independently whether the other side’s measurement results are correct and complete the mutual identity authentication. For example, if CA prepares a non-orthogonal states $|\phi^-\rangle_{AB}$ and the measurement result $R_{A_i} = 1$, then $R_{B_i \mod n} = 1$.

### Table 3 The Probabilities of CA’s Bell Basis Measurement Results according to $S_A$ and $S_B$.

| $R_{AB}$ | $\phi^+\rangle_{AB} \rightarrow 00$ | $\phi^-\rangle_{AB} \rightarrow 01$ | $\psi^+\rangle_{AB} \rightarrow 10$ | $\psi^-\rangle_{AB} \rightarrow 11$ |
|----------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| $|0\rangle_A | 0\rangle_B$ | 1/2 | 1/2 | 0 | 0 |
| $|0\rangle_A | 1\rangle_B$ | 0 | 0 | 1/2 | 1/2 |
| $|1\rangle_A | 0\rangle_B$ | 0 | 0 | 1/2 | 1/2 |
| $|1\rangle_A | 1\rangle_B$ | 1/2 | 1/2 | 0 | 0 |
| $|+\rangle_A | +\rangle_B$ | 1/2 | 0 | 1/2 | 0 |
| $|+\rangle_A | -\rangle_B$ | 0 | 1/2 | 0 | 1/2 |
| $|-\rangle_A | +\rangle_B$ | 0 | 1/2 | 0 | 1/2 |
| $|-\rangle_A | -\rangle_B$ | 1/2 | 0 | 1/2 | 0 |

Phase 3: the blind quantum computation phase

S3-1 After $A_i$ and $B_i \mod n$ authenticate each other successfully, $A_i$ needs to ask $B_i \mod n$ to generate a quantum resource state $|G\rangle$. Note that, a qubit of the resource state is $|G_{x,y}\rangle$, where $x$ denotes a row number and $y$ denotes a column number.
S3-2 $B_{i \mod n}$ sends these qubits of the resource state to $A_i$ via quantum channels one by one.

S3-3 $A_i$ measures these qubits in basis $\{|\pm \theta'_{x,y}\rangle\}$, where $\theta'_{x,y} = (-1)^s_{x,y} \theta_{x,y} + s_{x,y} \pi$. Note that $\theta_{x,y} \in \{-\frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{\pi}{2}\}$ is the desired measurement angle, $s_{x,y}$ is the summation of all previous measurement results in $X$ basis, $s_{x,y}$ is the summation of all previous measurement results in $Z$ basis, and the measurement result is recorded as $s_{x,y} \in \{0, 1\}$. If all qubits have been measured, the computation is completed.

As mentioned above, the client shares a key $K_{AB}$ with the specific server under the help of load balancers and CA in the registration phase. Then, if the client wants to perform a BQC task, she can log into the remote server by using the key $K_{AB}$ in the identity authentication phase. After the client and the server authenticate each other, the client can delegate her BQC task to the server securely. Here a brief analysis of the security of the reviewed protocol is given. Firstly, a reasonable assumption is that an unjammable public channel is required to ensure the integrity of transmitted classical messages between each load balancer and CA. Therefore, in Phase 1 and Phase 2, as long as the devices for preparing single-qubit states owned by Load_Balancer_A and Load_Balancer_B are secure, neither the semi-honest CA nor an external attacker Eve can obtain the shared key $K_{AB}$. In Phase 3, based on the no-signaling principle, the privacy of the clients can be well protected. In addition, because the decoy-state check technology is used in each quantum information transmission, this protocol can resist intercept-resend attack and entangle-measure attack. However, Load_Balancer_A and Load_Balancer_B are regarded as honest and they can get information about $S_A$, $S_B$, $R_{AB}$, from which the value of the shared key can be deduced. Once the load balancers are successfully attacked, the entire network is no longer secure. What’s more, in this protocol, only the clients who can make measurements are considered. Actually, the clients usually have different quantum capabilities in a real quantum network. Especially, the clients cannot determine whether the nodes in the
quantum network are honest or not. Therefore, it is necessary to consider the verifiability of their quantum computation.

### 4 Three proposed multi-party VBQC protocols

Since the clients in quantum networks have different quantum capabilities, such as making measurements, generating single-qubit states, or performing a few single-qubit gates, we propose three multi-party VBQC protocols with identity authentication suitable for three types of clients based on the mainly blind quantum computing model.

#### 4.1 The first secure multi-party VBQC protocol based on the receive-and-measure method

In this part, we construct a secure multi-party VBQC protocol mainly based on the receive-and-measure method (called RAM-SMVBQC). Similar to Shan et al.’s BQC protocol [39], in this protocol, the clients only have the ability to make measurements and own a little quantum memory. First, the clients should share a register key $K_{AB}$ with the specific server with the help of load balancers and CA. Then, the clients could employ the key $K_{AB}$ to access the remote quantum server to delegate their quantum computational tasks. The specific steps are as follows.

**Phase 1: the registration phase**

**A1-1** $A_i$ first sends a request $i$ to LoadBalancer_A who puts it into the request queue. Conforming to the FIFO principle, LoadBalancer_A forwards the request $i$ to LoadBalancer_B and CA. Then LoadBalancer_A generates a series of qubits randomly in $\{|0\rangle, |1\rangle, |+\rangle, |−\rangle\}$ with the length of $m + l + k$. The qubit sequence is denoted by $S_{LA_{decoy}}$ and is sent to $A_i$.

**A1-2** When $A_i$ receives $S_{LA_{decoy}}$, she randomly measures $l$ qubits in the corresponding bases, which should be told by LoadBalancer_A, and returns $S_{A_{decoy}}$ consisting of the rest of $m + k$ qubits to LoadBalancer_A. If the measurement results are correct, $A_i$ tells LoadBalancer_A the positions and bases of other $m$ qubits. LoadBalancer_A measures them in the informed measurement bases. The remaining qubit sequence is represented by $S_A$. Either of the
error rates in these two decoy detections is higher than the predefined threshold, \( A_i \) has to terminate the protocol and move to A1-1. Note that if \( A_i \) owns massive quantum memory, she can do it efficiently. Otherwise, she needs to set the clock period carefully to ensure that Bob cannot distinguish which qubits she measured.

A1-3 After Load_Balancer_B receives the request \( i \), he assigns an idle server \( B_i \mod n \) and transmits the request \( i \) to \( B_i \mod n \). Then \( B_i \mod n \) randomly generates \( m+k \) qubits in \( \{ |0\rangle, |1\rangle, |+\rangle, |-\rangle \} \), which form the sequence \( S_{\text{B\_decoy}} \). Next he sends \( S_{\text{B\_decoy}} \) to Load_Balancer_B. Similar to A1-2, after Load_Balancer_B receives these qubits, he has to perform the decoy detection and the remaining qubit sequence is represented by \( S_B \).

A1-4 CA performs Bell measurements on \( S_A \) and \( S_B \), which are sent from Load_Balancer_A and Load_Balancer_B, and records the measurement results as \( R_{AB} \). The measurement results \( \{ |\phi^+\rangle, |\psi^+\rangle, |\phi^-\rangle, |\psi^-\rangle \} \) are recorded as \( \{00, 01, 10, 11\} \), respectively. Then, CA sends \( R_{AB} \) to Load_Balancer_A and Load_Balancer_B via public classical channels. Finally, they resend \( R_{AB} \) to \( A_i \) and \( B_i \mod n \).

A1-5 When \( A_i \) and \( B_i \mod n \) receive \( R_{AB} \) successfully, they have to securely exchange the bases of qubits in \( S_A \) and \( S_B \). Similarly, according to Table 2, they keep the \( j \)-th bit under the same basis as the initial raw key bit \( K_{AB}^j \in \{0, 1\} \) when \( R_{AB}^j = 11 \). For example, if the \( j \)-th qubit in \( S_A \) or \( S_B \) is prepared in Z basis, then \( K_{AB}^j = 0 \). Otherwise, \( K_{AB}^j = 1 \).

A1-6 \( A_i \) and \( B_i \mod n \) have to pick a portion of their raw keys to estimate error rate and detect eavesdropping as a consequence of the effect of noise in the actual channel. If the error rate is below the threshold, the channel is secure. Otherwise, \( A_i \) has to terminate protocol and move to A1-1.

A1-7 The steps from A1-1 to A1-6 should be repeated until all clients have completed the registration and each of them has obtained a shared key with a specific network server, as illustrated in Fig. 1. Note that, if \( m > n \) and \( i > n \), \( B_i \mod n \) must securely store multiple keys in his memories.

Phase 2: the identity authentication phase
VBQC with identity authentication for different types of clients

A2-1 $A_i$ sends request $i$ to $\text{Load Balancer}_A$ when he wants to delegate a BQC to a remote server.

A2-2 Based on the FIFO principle, $\text{Load Balancer}_A$ resends the request $i$ to $\text{CA}$, who needs to randomly prepare $4k$ non-orthogonal states selected in $\{|\phi^-, |\psi^+, |\Phi^-, |\Psi^+\rangle\}$ as shown in Eq. (6). All the first qubits of $|\varphi\rangle_{AB}$ form the qubit sequence $S_{A}'$ which should be sent to $\text{Load Balancer}_A$ and the second qubits of them form the qubit sequence $S_{B}'$ which should be sent to $\text{Load Balancer}_B$. Note that, $\text{CA}$ also encodes the non-orthogonal states as $S_{AB}$ which denoted $|\phi^+\rangle \rightarrow 00$, $|\phi^-\rangle \rightarrow 01$, $|\psi^+\rangle \rightarrow 01$, $|\psi^-\rangle \rightarrow 11$, and broadcasts them to $A_i$ and $B_{i \ mod \ n}$. When $\text{Load Balancer}_A$ and $\text{Load Balancer}_B$ receive $S_{A}'$ and $S_{B}'$, they insert some decoy qubits into $S_{A}'$ and $S_{B}'$ and send the new qubit sequences $S_{A_{\text{decoy}}}'$ and $S_{B_{\text{decoy}}}'$ to $A_i$ and $B_{i \ mod \ n}$, respectively.

A2-3 After eavesdropping detection, $A_i$ and $B_{i \ mod \ n}$ measure $S_{A}'$ and $S_{B}'$ in the bases in terms of the initial key $K_{AB}$ and record the measurement results $R_{A_i}$ and $R_{B_{i \ mod \ n}}$. Then $A_i$ and $B_{i \ mod \ n}$ randomly reveal $3k$ values of $R_{A_i}$ and $R_{B_{i \ mod \ n}}$ and the corresponding positions via classical channels.

A2-4 According to Table 3, $A_i$ and $B_{i \ mod \ n}$ check independently whether the other side’s measurement results are correct and complete the mutual identity authentication similar to S2-5 in the reviewed BQC protocol. The information flow of Phase 2 is shown in Fig. 2.

Phase 3: the blind quantum computation phase
A3-1 After $A_i$ and $B_{i \mod n}$ authenticate each other successfully, $B_{i \mod n}$ should generate a $N(\alpha + \beta + 1)$-qubit state $\rho$. The state $\rho$ consists of $\alpha + \beta + 1$ registers, and each register stores $N$ qubits, where $\alpha$ and $\beta$ are both real numbers. If $B_{i \mod n}$ is honest, he generates a graph state $|G\rangle \equiv (\bigotimes_{e \in E} CZ_e)|+\rangle^\otimes N$ in each register, which has been introduced in Sec. 2.

A3-2 To reduce the workload of $B_{i \mod n}$ and improve the efficiency of the protocol, $B_{i \mod n}$ can request for help from other idle servers through LoadBalancer_B. To be more specific, $B_{i \mod n}$ broadcasts a request to the other idle servers with the help of LoadBalancer_B. If $M$ idle servers respond to the request, $B_{i \mod n}$ announces the information of $|G\rangle$. Then, each of them needs to generate the state $|G\rangle^\otimes \frac{\alpha + \beta + 1}{M}$. $B_{i \mod n}$ sends each qubit of $|G\rangle^\otimes \frac{\alpha + \beta + 1}{M}$ to $A_i$ one by one. The graph state generated by idle servers are resent by $B_{i \mod n}$ to $A_i$. Note that, $B_{i \mod n}$ can accept $A_{i+1}$’s request after he generates the graph state and he only needs to send a qubit of $|G\rangle$ from his quantum memory.

A3-3 If $M$ idle servers respond to the request, $B_{i \mod n}$ chooses $\alpha$ registers uniform randomly and discards them to guarantee that the remaining state is close to an independent and identically distributed sample by using the quantum de Finetti theorem [41]. Then she chooses $\beta$ graph states $|G\rangle$ at random for stabilizer test. If the stabilizer test passes, she uses the remaining one graph state $\rho_A$ for MBQC.

Security and verifiability analysis. We first show the security of registration phase (Phase 1) and the identity authentication phase (Phase 2), which
means that no matter whether there is an external or internal attacker, the client’s register key $K_{AB}$ cannot be obtained by attackers in Phase 1. Likewise, the attackers also cannot replace the identity of the client in Phase 2. Then it is necessary to show the protocol in the blind quantum computation phase (Phase 3) satisfies blindness, correctness, and verifiability.

In Phase 1, load balancers and CA are semi-honest parties who should follow the protocol honestly, but can try to extract information by themselves without colluding with other entities. First, CA is considered as an insider attacker who wants to obtain the client’s register key $K_{AB}$. CA cannot distinguish the basis of the qubits which were sent by load balancers in A1-2 and A1-3 since the non-orthogonal basis is not distinguishable. Therefore, the probability of detecting dishonest behavior of CA is $\frac{1}{2^m}$ where $m$ is the length of $S_A$ and $S_B$. For example, if the qubits resent by Load_Balancer_A and Load_Balancer_B from $A_i$ and $B_{i \mod n}$ are $|0\rangle_A$ and $|0\rangle_B$ respectively, the measurement result of the CA should be $|\psi^+\rangle_{AB}$ or $|\psi^-\rangle_{AB}$ based on Table 2. The malicious CA will return a random result and the probability of returning an incorrect result is $\frac{1}{2}$. Similarly, Load_Balancer_A and Load_Balancer_B can be regarded as particle generators to randomly prepare single photon sequences and their roles in the quantum network are only to reduce the quantum capability of the clients. If they are dishonest and seen as insider attackers, their malicious behaviors can be detected with a certain probability due to the decoy detection in A1-2 and A1-3. As shown above, an insider cannot obtain the register key $K_{AB}$. In addition, if there exist an external attacker Eve, who can intercept the qubit sequence $S_A$ and $S_B$ from Load_Balancer_A and Load_Balancer_B and make Bell measurements on them. But based on the measurement results, Eve can only know whether the quantum states of $A_i$ and $B_{i \mod n}$ are the same or opposite and the shared key $K_{AB}$ cannot be deduced. For example, if Eve obtains the result $|\psi^-\rangle_{AB}$, then he can only get that the quantum states of $A_i$ and $B_{i \mod n}$ are the opposite. Eve cannot judge whether the quantum state is prepared in X basis or Z basis. To sum
up, neither insider nor external attackers can obtain the value of the shared key $K_{AB}$ in Phase 1.

In Phase 2, if CA is dishonest and considered as an insider attacker, he sends wrong states to $A_i$ and $B_i \mod n$. However, his dishonest behavior will be detected because $A_i$ and $B_i \mod n$ need to announce the measurement results and check each other’s measurement results independently. Besides, even if LoadBalancer\_A and LoadBalancer\_B are regarded as insider attackers, they still cannot know the value of $K_{AB}$ and thus cannot pretend to be $A_i$ or $B_i \mod n$. From the view of an outside attacker Eve, he cannot distinguish the intercepted non-orthogonal states that CA sent due to Heisenberg’s uncertainty principle. Furthermore, decoy check is used during each transmission. Then, any operation on the quantum state may change the measurement result and the client and the server will discover the attacks with a non-zero probability. Therefore, the proposed protocol can avoid the intercept-resend attack and the entangle-measure attack.

In Phase 3, a single-server BQC method is used, where only a one-way quantum communication from $B_i \mod n$ to $A_i$ is needed. The blindness is guaranteed by the no-signaling principle [42], which is more fundamental than quantum mechanics. Therefore, the server cannot get anything about the client’s input, output and algorithm except the size of the graph state $|G\rangle$. Next, we will show that the protocol satisfies $\epsilon$-verifiability, namely that the probability that client accepts an incorrect result should be bounded by $\epsilon$. It is easy to see that if $B_i \mod n$ behaves honestly, $A_i$ will pass the stabilizer test and accept the outcome of the computation with probability 1. If $A_i$ passes the test, the state $\rho_A$ which can be used for computation satisfies

$$\langle G | \rho_A | G \rangle \geq 1 - \frac{1}{N} = \epsilon,$$

with probability at least $1 - \frac{1}{N}$ [30]. It means that if $A_i$ passes the test, then the state $\rho_A$ used for computation is close to the ideal state $|G\rangle$. Otherwise, the computation state $\rho_A$ is far from the ideal state and $A_i$ will reject the
computation result. Then a short proof is given for verifiability based on the work in Refs. [30, 43].

Proof For any $N$-qubit state $\delta$, we can get
\[
\text{Tr}[(T^{\otimes \beta} \otimes \Pi^{\perp})^{\otimes \beta+1}] \leq \frac{1}{2N^2}.
\]  
(8)

Note that $\Pi^{\perp} = I^{\otimes N} - |G\rangle\langle G|$ and $T$ is a POVM that can make $A_i$ accept the test. Then we can get
\[
\text{Tr}(T\delta) = \frac{1}{2} + \frac{1}{2} \langle G|\delta|G\rangle,
\]  
(9)
\[
\text{Tr}(\Pi^{\perp}\delta) = 1 - \langle G|\delta|G\rangle
\]  
(10)

According to Eq. (9) and Eq. (10), we obtain
\[
\text{Tr}[(T^{\otimes \beta} \otimes \Pi^{\perp})^{\otimes \beta+1}] = \text{Tr}(T\delta)^{\beta}\text{Tr}(\Pi^{\perp})
\]  
(11)
\[
= 2\text{Tr}(T\delta)^{\beta}(1 - \text{Tr}(T\delta)).
\]  
(12)

Let $\text{Tr}(T\delta) = \frac{\beta}{\beta+1}$, Eq. (11) can achieve the maximum
\[
2\text{Tr}(T\delta)^{\beta}(1 - \text{Tr}(T\delta))
\]  
(13)
\[
= 2\left(\frac{\beta}{\beta+1}\right)^{\beta}(1 - \frac{\beta}{\beta+1}) \leq \frac{1}{2N^2}.
\]  
(14)

Then we have
\[
\text{Tr}[(T^{\otimes \beta} \otimes \Pi^{\perp})^{\otimes \beta+1}] = \text{Tr}(T^{\otimes \beta})\text{Tr}(\Pi^{\perp})\text{Tr}(\rho_A)
\]  
(15)
\[
\leq \frac{1}{2N^2} + \frac{1}{2N^2} = \frac{1}{N^2}.
\]  
(16)

If $\text{Tr}(\Pi^{\perp}\rho_A) \geq \frac{1}{N}$, then $\text{Tr}(T^{\otimes \beta}) \leq \frac{1}{N}$. Therefore
\[
\text{Tr}(T^{\otimes \beta})\text{Tr}(\Pi^{\perp})\text{Tr}(\rho_A) = \text{Pr}(\text{accept})E(1 - \langle G|\rho_A|G\rangle),
\]  
(17)
where $E(\cdot)$ is the expected value and $\text{Pr}(\text{accept})$ is the probability that $A_i$ accepts the test. For Eq. (14), according to Markov inequality
\[
\frac{1}{N}\text{Pr}[(1 - \langle G|\rho_A|G\rangle) > \frac{1}{N}] \leq E(1 - \langle G|\rho_A|G\rangle) \leq \frac{1}{N},
\]  
(18)

it is easy to deduce
\[
\text{Pr}(\text{accept})\text{Pr}[(1 - \langle G|\rho_A|G\rangle) > \frac{1}{N}] \leq \frac{1}{N}
\]  
(19)

which means $\langle G|\rho_A|G\rangle \geq 1 - \frac{1}{N}$ with probability $1 - \frac{1}{N}$ when $A_i$ accepts the test.

### 4.2 The second secure multi-party VBQC protocol based on the prepare-and-send method

Next we introduce the second secure multi-party VBQC protocol based on the prepare-and-send method (called PAS-SMVBQC) where the clients can only generate single qubits. In this protocol, the clients also have to register first and then delegate their computational tasks to remote quantum servers according to register keys. The specific steps are given as follows.

Phase 1: the registration phase
B1-1 The client $A_i$ sends a request $i$ to Load_Balancer_A who puts it into the request queue. According to the FIFO principle, Load_Balancer_A retransmits the request $i$ to Load_Balancer_B.

B1-2 The client $A_i$ and the server $B_{i \mod n}$ prepare a series of single quantum states randomly in X basis or Z basis which are denoted by $S_A$ and $S_B$, respectively, and the length of them is $k$. Then they insert decoy states in $\{|0\rangle, |1\rangle, |+\rangle, |−\rangle\}$ with fixed length into $S_A$ and $S_B$ and send the new sequences $S_{A_{\text{decoy}}}$ and $S_{B_{\text{decoy}}}$ to Load_Balancer_A and Load_Balancer_B. After they receive these qubits, $A_i$ and $B_{i \mod n}$ tell positions and the basis of decoy qubits. Load_Balancer_A and Load_Balancer_B measure them in the informed measurement bases. If the error rate exceeds the predefined threshold, $A_i$ has to terminate the protocol and move to B1-1.

B1-3 CA performs Bell measurements on $S_A$ and $S_B$, which are sent from Load_Balancer_A and Load_Balancer_B, and records the measurement results as $R_{AB}$. CA sends $R_{AB}$ to Load_Balancer_A and Load_Balancer_B across public classical channels. Then Load_Balancer_A resends $R_{AB}$ to $A_i$ and Load_Balancer_B resends $R_{AB}$ to $B_{i \mod n}$.

B1-4 When $A_i$ and $B_{i \mod n}$ receive $R_{AB}$ successfully, they have to securely exchange the bases of qubits in $S_A$ and $S_B$. Similar to A1-5, they keep the $j$-th bit under the same basis as the initial raw key bit $K^j_{AB} \in \{0, 1\}$ when $R^j_{AB} = 11$.

B1-5 $A_i$ and $B_{i \mod n}$ have to pick a portion of their raw keys to estimate error rate and detect eavesdropping because of the effect of noise in the actual channel. If the error rate is below the threshold, the channel is secure. Otherwise, $A_i$ has to terminate the protocol and move to B1-1.

B1-6 The steps from B1-1 to B1-5 should be repeated until all clients have completed the registration and each of them has obtained a shared key with a specific network server.

Phase 2: the identity authentication phase

B2-1 As shown in Fig. 3, if a registered client $A_i$ wants to delegate a quantum computational task to a remote server, he needs to send a request $i$ to Load_Balancer_A firstly.

B2-2 According to the FIFO principle, Load_Balancer_A resends the request $i$ to Load_Balancer_B and CA, and Load_Balancer_B resends the request to $B_{i \mod n}$.

B2-3 $A_i$ and $B_{i \mod n}$ prepare single qubit sequences $S'_A$ and $S'_B$ randomly in X basis or Z basis according to the raw key $K_{AB}$. The encoding method is shown in Table 4. Then they also prepare decoy sequences in $\{|0\rangle, |1\rangle, |+\rangle, |−\rangle\}$ with fixed length and insert into $S'_A$ and $S'_B$ to generate new sequences $S'_{A_{\text{decoy}}}$ and $S'_{B_{\text{decoy}}}$, which should be sent to Load_Balancer_A and Load_Balancer_B.

B2-4 When Load_Balancer_A and Load_Balancer_B receive $S'_{A_{\text{decoy}}}$ and $S'_{B_{\text{decoy}}}$, they both need to measure the decoy qubits, the positions and basis of which are informed by $A_i$ and $B_{i \mod n}$. If the eavesdropping detection passes, they transmit $S'_A$ and $S'_B$ to CA.

B2-5 After CA receives the sequences $S'_A$ and $S'_B$, he randomly generates $k_{AB}/2$ bits, where $k_{AB}$ is the length of $K_{AB}$ and the $j$-th bit is denoted by $M_j \in \{0, 1\}$ for $j \in \{0, 1, ..., k_{AB}/2\}$. If $M_j = 0$, he measures the $j$-th qubit of $S'_A$ in Z basis and records the measurement results $R^j_{A}$. Otherwise he measures it in X basis. He also needs to measure $S'_{B}$ according to the value of $M$ and records the measurement results as $R^j_{B}$. Then, CA announces the value of $R^j_{A}$ and $R^j_{B}$.
VBQC with identity authentication for different types of clients

Fig. 3  (Color Online) Information flow of identity authentication phase in the PAS-SMVBQC protocol

B2-6 According to $R'_A$ and $R'_B$, $A_i$ and $B_{i \bmod n}$ could check the identity of the other party. For example, assume that $K_{AB}$ is an 8-bit string and $K_{AB} = 01001011$, as shown in Table 4, the qubit sequence of $S'_A$ and $S'_B$ are $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. If CA generates $M = 0101$ and the measurement result $R'_A = 1011$, according to Table 4 and the value of $K_{AB}$, $A_i$ can confirm the identity of $B_{i \bmod n}$ when $R'_{1B} = 1$ and $R'_{4B} = 1$.

| $K_nK_{n+1}$ | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| $S'_A$ and $S'_B$ | $|0\rangle$ | $|1\rangle$ | $|+\rangle$ | $|-\rangle$ |

| Measurement basis |
|---|
| Z | X | Z | X | Z | X |

| $R'_A$ and $R'_B$ |
|---|
| 0 | 0 or 1 | 1 | 0 or 1 | 0 or 1 | 0 | 0 or 1 | 1 |

Table 4 The Rules of Code $S'_A$ and $S'_B$ according to $K_nK_{n+1}$ and Corresponding Measurement Results $R'_A$ and $R'_B$

Phase 3: the blind quantum computation phase

B3-1 After $A_i$ and $B_{i \bmod n}$ authenticate each other successfully, $A_i$ wants to ask the quantum server $B_{i \bmod n}$ to help her perform the task of VBQC.

B3-2 $A_i$ generates $C + T$ qubits and the state of each qubit is $|+\theta_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta_i}|1\rangle)$, where $\theta_i$ is uniformly and randomly chosen from the set $\{0, \pi/4, 2\pi/4, ..., 7\pi/4\}$. Note that, $C$ is the number of computation qubits used for performing computation and $T$ is the number of trap qubits used for verification. $A_i$ should also generate $D$ dummy qubits which are chosen at random from $\{|0\rangle, |1\rangle\}$. Then $A_i$ sends these qubits to Bob.

B3-3 $B_{i \bmod n}$ generates graph state $|G\rangle$ as required by Alice, where $|G\rangle$ is an $N$-qubit universal graph state and $N = C + T + D$.

B3-4 $A_i$ asks $B_{i \bmod n}$ to measure these qubits. If a qubit in $|G\rangle$ is used for computation, then it will be measured in the basis $\{|0\rangle \pm e^{i\delta_i}|1\rangle\}$ where $\delta_i = \phi_i + \pi/4$.
\( \theta_i + r_i \pi, \phi_i' \) is obtained according to the previous measurements, and \( r_i \) is chosen from the set \{0, 1\} at random. However, trap qubits will be measured in the basis \(|0\rangle \pm e^{i \theta_i + r_i \pi} |1\rangle\} and dummy qubits are measured in the basis randomly chosen from \{|0\rangle \pm |1\rangle, |0\rangle \pm e^{i 2\pi/4} |1\rangle, |0\rangle \pm e^{i 3\pi/4} |1\rangle\}.

B3-5 If all the measurement results of trap qubits are correct, \( A_i \) recovers the outcome of the computation as she knows the value of each random number \( r_i \).

Security and verifiability analysis. Similar to the RAM-SVMBQC protocol, both CA and load balancers are regarded as semi-honest. The insider attacks and outsider attacks of Phase 1 and Phase 2 will be analyzed. In addition, the correctness, blindness and verifiability of Phase 3 are also discussed in the following.

Clients can only measure single qubits in the first RAM-SVMBQC protocol and they must ask the load balancers for help to generate single qubits. However, in the second PAS-SMVBQC protocol, the clients could generate single qubit by themselves. The role of load balancers is only to resend qubits. If they are dishonest and seen as insider attackers, their malicious behaviors will be detected with a non-zero probability in B1-2 because of the decoy detection. Besides, similar to the RAM-SVMBQC, if there exist an insider attack CA, he cannot distinguish the basis of \( S_A \) and \( S_B \) and his malicious behaviors will be detected with probability \( \frac{1}{2^n} \), where \( n \) is the length of \( S_A \) and \( S_B \). Furthermore, due to decoy detection in each transmission, the proposed protocol can defend against external attackers. In Phase 2, CA only needs to measure \( S_A \) and \( S_B \) according to the values of \( M \), which are generated by himself. Neither a dishonest CA nor an outside attacker Eve can get any information without being detected. Thus, the security of the client and the server can be guaranteed by using the initial security key shared between the client and the server.

In Phase 3, if \( B_{i \ mod \ n} \) behaves honestly, then all the measurements on trap qubits will produce correct results and the computation will be performed correctly. Even if \( B_{i \ mod \ n} \) behaves maliciously, he cannot get any information about \( A_i \) as well. Then, we will prove the blindness of the protocol.

**Proof** The client’s quantum input is blind for server, as the state that \( B_{i \ mod \ n} \) received from \( A_i \) is

\[
\frac{1}{10} \sum_{\theta_i} (|+\theta_i\rangle\langle+\theta_i| + |0\rangle\langle0| + |1\rangle\langle1|)
\]

\[
= \frac{1}{10} (|+\rangle\langle+| + \cdots + |+\frac{7\pi}{4}\rangle\langle+\frac{7\pi}{4}| + |0\rangle\langle0| + |1\rangle\langle1|)
\]

\[
= \frac{I}{2},
\]

which is a maximally mixed state.

Then we show the client \( A_i \)'s algorithm and output are also hidden from the server by using the method similar to that in Refs. [44, 45].

**Proof** Let \( \Omega_i = \{\delta_i\}_{i=1}^N \) is the set of angles that \( A_i \) sends to \( B_{i \ mod \ n} \), \( \Theta_i = \{\theta_i\}_{i=1}^N \) is the set of \( A_i \)'s privacy, and \( R_j = \{r_i\}_{i=1}^N \in \{0, 1\} \) is a hidden binary parameters chosen by \( A_i \). Let \( \{\Pi_i\}_{i=1}^m \) be a POVM which is performed by \( B_{i \ mod \ n} \), and \( \Lambda \in \{1 \ldots m\} \) are the corresponding results of the POVM. Suppose that \( B_{i \ mod \ n} \) wants to know \( A_i \)'s privacy. Then we have
A verifiable blind quantum computation protocol (called CB-SMVBQC) is proposed.

By using the circuit model similar to that in Ref. [24], the third secure multi-party verifiable blind quantum computation protocol (called CB-SMVBQC) is proposed.
In this protocol, if the clients can only perform quantum gates $H$ and $\sigma_z^{1/4}$, they can also remotely access a quantum server with the help of load balancer and the CA, and delegate their quantum computational tasks to the server. The specific steps of the proposed protocol are as follows.

Phase 1: the registration phase

C1-1 The client $A_i$ sends a request $i$ to Load_Balancer_A who puts it into the request queue. According to the FIFO principle, Load_Balancer_A retransmits the request $i$ to Load_Balancer_B. Load_Balancer_A also prepares $m + k$ qubits $|0\rangle$ which form the sequence $S_{LA}$ and send them to $A_i$.

C1-2 $A_i$ randomly generates $|0\rangle = |0\rangle, |1\rangle = H(\sigma_z^{1/4})^4 H|0\rangle, |+\rangle = H|0\rangle, \text{ or } |-\rangle = (\sigma_z^{1/4})^4 H|0\rangle$ by applying some quantum gates such as $H$ and $\sigma_z^{1/4}$ on each received qubit of $S_{LA}$. Then $A_i$ returns the new qubit sequences $S_{A_{\text{decoy}}}$ to Load_Balancer_A. After Load_Balancer_A receives these qubits, $A_i$ tells the positions and bases of $m$ qubits. Load_Balancer_A measures them in the informed measurement basis and the remaining qubits are expressed as $S_A$ which should be sent to CA. If the error rate is higher than the predefined threshold, an eavesdropper Eve is considered to exist in the channel and the protocol should restart.

C1-3 $B_{i \mod n}$ randomly prepares $m + k$ single qubits in $\{|0\rangle, |1\rangle, |+\rangle, |−\rangle\}$, which compose the qubit sequence $S_{B_{\text{decoy}}}$ and will be sent to Load_Balancer_B. After eavesdropping detection, the remaining qubits $S_B$ should be sent to CA. Similar to that in C1-2, if the error rate is higher than the predefined threshold, eavesdropping may exist and the protocol need start again.

C1-4 CA performs Bell measurements on the qubit sequences $S_A$ and $S_B$ and records the measurement results $R_{AB} \in \{|\phi^+\rangle \rightarrow 00, |\psi^+\rangle \rightarrow 01, |\phi^-\rangle \rightarrow 10, |\psi^-\rangle \rightarrow 11\}$. CA sends $R_{AB}$ to Load_Balancer_A and Load_Balancer_B across public classical channels. Then they resend $R_{AB}$ to $A_i$ and $B_{i \mod n}$.

C1-5 When $A_i$ and $B_{i \mod n}$ receive $R_{AB}$, they have to securely exchange the bases of qubits in $S_A$ and $S_B$. Similarly, according to Table 2, they keep the $j$-th bit under the same basis as the initial raw key bit $K_{AB}^j \in \{0, 1\}$ when $R_{AB}^j = 11$.

C1-6 $A_i$ and $B_{i \mod n}$ need to pick a portion of their raw key bits to estimate error rate and detect eavesdropping because of the effect of noise in the actual channel. If the error rate exceeds the threshold, $A_i$ has to terminate the protocol and move to C1-1 again.

C1-7 The steps from C1-1 to C1-6 should be repeated until all clients have completed the registration and each of them has shared a key with a specific network server. Note that, if $m > n$ and $i > n$, $B_{i \mod n}$ must store multiple keys in his memories securely.

Phase 2: the identity authentication phase

C2-1 If a registered client $A_i$ wants to securely delegate a quantum computational task to a remote server, he needs to send a request $i$ to Load_Balancer_A firstly.

C2-2 According to the FIFO principle, Load_Balancer_A resends the request $i$ to Load_Balancer_B and CA and Load_Balancer_B resends the request to $B_{i \mod n}$.

C2-3 Load_Balancer_A prepares $m + k$ qubits in the form of $|0\rangle$ and sends them to $A_i$. When each qubit arrives, $A_i$ performs the corresponding single-qubit gates on it to make its state be one of $\{|0\rangle, |1\rangle, |+\rangle, |−\rangle\}$, where $|0\rangle = |0\rangle, |1\rangle = H(\sigma_z^{1/4})^4 H|0\rangle, |+\rangle = H|0\rangle$, and $|−\rangle = (\sigma_z^{1/4})^4 H|0\rangle$. Then $A_i$ returns these qubits which form the sequence $S_{A_{\text{decoy}}}'$ to Load_Balancer_A.

C2-4 $B_{i \mod n}$ prepares single qubit sequences $S_A'$ and $S_B'$ randomly in X basis or Z basis according to the raw key $K_{AB}$. Then they also prepare decoy sequences
VBQC with identity authentication for different types of clients

in \{0, 1, |+\rangle, |−\rangle\} with fixed length and insert into \(S'_B\) to make a new sequences \(S'_{B_{\text{decoy}}}\) which will be transmitted to Load_Balancer_B.

C2-5 After Load_Balancer_A and Load_Balancer_B receive \(S'_{A_{\text{decoy}}}\) and \(S'_{B_{\text{decoy}}}\), they both need to measure the decoy qubits, the positions and basis of which are informed by \(A_i\) and \(B_{i \mod n}\). If the eavesdropping detection passes, they transmit \(S'_A\) and \(S'_B\) to CA.

C2-6 When CA receives the sequences \(S'_A\) and \(S'_B\), he randomly generates \(\frac{k_{AB}}{2}\) bits, where \(k_{AB}\) is the length of \(K_{AB}\) and the \(j\)-th bit is denoted by \(M_j \in \{0, 1\}\) for \(j \in \{0, 1, ..., \frac{k_{AB}}{2}\}\). If \(M_j = 0\), he measures the \(j\)-th qubit of \(S'_A\) in Z basis and records the measurement results \(R'_j\); otherwise he measures it in X basis. Similarly, he also needs to measure \(S'_B\) and records the measurement results as \(R'_B\). Then, CA announces the value of \(R'_A\) and \(R'_B\).

C2-7 Similar to B2-6, \(A_i\) and \(B_{i \mod n}\) could check the identity of the other party based on the value of \(R'_A\) and \(R'_B\).

Phase 3: the blind quantum computation phase

C3-1 After \(A_i\) and \(B_{i \mod n}\) authenticate each other successfully, she asks the specific server \(B_{i \mod n}\) to initiate the task of BQC.

C3-2 \(B_{i \mod n}\) sends \(N\) qubits in states \(|0\rangle\) to \(A_i\), where \(N = C + T + D\). \(A_i\) randomly chooses \(C\) qubits to be used for computation, \(T\) qubits to be considered as trap qubits, and \(D\) qubits to be taken as dummy qubits. For computation and trap qubits, she performs the \(H\) gate on these qubits and then randomly applies the gate \(\sigma_z^{1/4}\) on it for \(n\) times in order to make the state of the qubit be \(\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{n\pi i}{4}}|1\rangle)\), where \(n\) is uniformly chosen from \(\{0, 1, ..., 7\}\). For dummy qubits, \(A_i\) performs the corresponding single-qubit gates to make them become \(|0\rangle = |0\rangle\) or \(|1\rangle = H(\sigma_z^{1/4})^4\). All of these qubits will be sent to \(B_{i \mod n}\) after \(A_i\) generates her expected states.

C3-3 \(B_{i \mod n}\) generates an \(N\)-qubit universal graph state \(|G\rangle\) as required by \(A_i\). Then \(A_i\) asks \(B_{i \mod n}\) to measure the qubits as follows: computation qubits will be measured in the basis \(\{|0\pm e^{i\delta_i}|1\rangle\}\) where \(\delta_i = \phi'_i + \theta_i + r_i\pi\) and \(\phi'_i\) is obtained according to the previous measurements and \(r_i\) is chosen from the set \(\{0, 1\}\) at random; trap qubits will be measured in the basis \(\{|0\pm e^{i\theta_i+r_i\pi}|1\rangle\}\); and dummy qubits are measured randomly in the basis chosen from \(\{|0\pm 1|0\rangle, |0\pm e^{i\pi/4}|1\rangle, |0\pm e^{i2\pi/4}|1\rangle, |0\pm e^{i3\pi/4}|1\rangle\}\).

C3-4 After all the measurements have been performed, \(A_i\) recovers the outcome of the computation by using \(r_i\). If all trap measurements succeed, \(A_i\) accepts the result; otherwise she rejects.

Security and verifiability analysis. Similar to the proposed RAM-SMVBQC and PAS-SMVBQC protocols, Phase 1 and Phase 2 of the CB-SMVBQC protocol can be shown to be secure against the insider attack and outside attack, even if load balancers and CA are both semi-honest. In Phase 3, the proposed CB-SMVBQC also satisfies blindness, correctness and verifiability.

In Phase 1, same as the previous two protocols, even if CA is dishonest, his malicious behavior will be detected with probability \(\frac{1}{2^n}\), where \(n\) is the length of \(S_A\) and \(S_B\). In addition, load balancers can be regarded as particle generators to prepare state \(|0\rangle\). They need to generate initial quantum states and measure the qubits returned by \(A_i\). Since the qubits should be measured by load balancers in a random basis \(\{|0\rangle, |1\rangle\}\) or \(\{|+, |−\rangle\}\), any malicious behavior of load balancers will be discovered with a nonzero probability. Furthermore, in Phase 2, the role of CA is only
to measure states sent by Load_Balancer_A and Load_Balancer_B. The client and server can use the initial shared key $K_{AB}$ to perform mutual identity authentication at the same time. No matter whether there is dishonest CA or Eve in Phase 2, the security of the client and the server can be guaranteed by using the initial security key shared between the client and the server. In Phase 3, if $B_i \mod n$ is honest, $A_i$ can get the correct result of her computation. Similarly, according to Eqs. (17-19), the blindness of the protocol is guaranteed. Besides, even if $B_i \mod n$ is dishonest, he cannot get any useful information about $A_i$. Finally, the verifiability of the proposed protocol is based on trap verification [26] and $A_i$ could verify her result through embedding trap and dummy qubits in computation.

5 Comparisons

We compare three proposed multi-party VBQC protocols in quantum networks that depend on different quantum capabilities of clients with Shan et al.’s multi-party BQC protocol [39] and other similar BQC protocols [12, 18] mainly from the trust-worthiness of CA, load balancers, and servers, the quantum capability that a client needs, the quantum capability of servers, the verifiability of the computation, and the security of key for identity authentication key as shown in Table 5. Both in the BFK protocol and the MF protocol, only an honest remote server is considered. However, in Shan et al.’s protocol and three proposed protocols, there are multiple parties in quantum networks. But in Shan et al.’s protocol, both the servers and load balancers should be honest, while they can be semi-honest in the three proposed protocols. Besides, only one type of the clients that have the ability to make measurements are considered and they cannot verify their computation results in Shan et al.’s protocol [39]. In the proposed protocols, clients’ quantum capabilities are flexible and three types of clients with different quantum capabilities have been taken into consideration. Furthermore, clients can both verify the correctness of the computation and the honesty of load balancers and the servers. In addition, in the BFK protocol and the MF protocol, identity authentication between the client and the server is not considered, while in the proposed protocols and Shan et al.’s protocol, mutual authentication among multiple clients and servers is realized. Especially, the raw key shared by clients and the server cannot be obtained by load balancers in the proposed protocols, while it is possible in Shan et al.’s protocol [39].

6 Conclusion

Different models are suitable for the clients with various quantum capabilities. In order to provide secure delegation of quantum computing for three types of clients with different quantum capabilities in quantum networks, three VBQC protocols with identity authentication have been put forwarded in this paper. In the first protocol, the clients only have to measure a single qubit. In the second protocol, the clients have the ability to generate single qubits. In the third protocol, the clients only have to perform a few single-qubit gates. Besides, in the all three proposed protocols, the CA, servers, and load balancers are considered to be semi-honest, and the load balancers are unable to obtain the shared key between the client and the server. The identity authentication between the clients and the specific servers can also be achieved. Furthermore, the clients with various quantum abilities can delegate their quantum computation in a secure way while they can also verify the correctness of
Table 5 Comparisons Between The Proposed Protocols and Other BQC Protocols

| Protocol                  | Trustworthiness of CA, load balancers and servers | The quantum capability of client | The quantum capability of server | Verifiability | Raw key for identity authentication |
|---------------------------|---------------------------------------------------|---------------------------------|---------------------------------|---------------|-------------------------------------|
| Shan et al.’s protocol [39] | Honest load balancers and servers, a semi-honest CA | Measuring single-qubit states | Full quantum | None | Obtained by load balancer, client and server |
| BFK protocol [12]          | A trust server | Preparing single-qubit states | Full quantum | Trap verification | N/A |
| MF protocol [18]           | A trust server | Measuring single-qubit states | Full quantum | None | N/A |
| RAM-SMVQC protocol         | Semi-honest load balancers, servers and a CA | Measuring single-qubit states | Full quantum | Stabilizer test | Only obtained by client and server |
| PAS-SMVQC protocol         | Semi-honest load balancers, servers and a CA | Preparing single-qubit states | Full quantum | Trap verification | Only obtained by client and server |
| CB-SMVQC protocol          | Semi-honest load balancers, servers and a CA | Implementing $H$ and $\sigma_x^{1/4}$ gates | Full quantum | Trap verification | Only obtained by client and server |

their calculation. However, the consumed quantum resources are still high in the proposed protocols. It deserves further study to improve the efficiency since quantum resources are still precious in quantum networks at present.

References

[1] Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994)

[2] Leibfried, D., Blatt, R., Monroe, C., et al.: Quantum dynamics of single trapped ions. Rev Mod Phys 75(1), 281–324 (2003)

[3] Blatt, R., Roos, C.F.: Quantum simulations with trapped ions. Nat Phys 8(4), 277–284 (2012)

[4] Krantz, P., Kjaergaard, M., Yan, F., et al.: A quantum engineer’s guide to superconducting qubits. Appl Phys Rev 6(2), 21318 (2019)

[5] Kjaergaard, M., Schwartz, M.E., Braumüller, J., et al.: Superconducting qubits: current state of play. Annu Rev Condens Matter Phys 11(1), 369–395 (2020)

[6] Wang, X.L., Luo, Y.H., Huang, H.L., et al.: 18-qubit entanglement with six photons’ three degrees of freedom. Phys Rev Lett 120(26), 260502
VBQC with identity authentication for different types of clients

(2018)

[7] Wang, H., Qin, J., Ding, X., et al.: Boson sampling with 20 input photons and a 60-mode interferometer in a \(10^{14}\)-dimensional hilbert space. Phys Rev Lett 123(25), 250503 (2019)

[8] He, Y., Gorman, S.K., Keith, D., et al.: A two-qubit gate between phosphorus donor electrons in silicon. Nature 571(7765), 371–375 (2019)

[9] Hensen, B., Huang, W., Yang, C.H., et al.: A silicon quantum-dot-coupled nuclear spin qubit. Nat Nanotechnol 15(1), 13–17 (2020)

[10] Childs, A.M.: Secure assisted quantum computation. Quantum Inf Comput 5(6), 456–466 (2005)

[11] Arrighi, P., Salvail, L.: Blind quantum computation. Int J Quantum Inf 4(5), 883–898 (2006)

[12] Broadbent, A., Fitzsimons, J.F., Kashefi, E.: Universal blind quantum computation. In: Proceeding of the 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 517–526 (2009)

[13] Barz, S., Kashefi, E., Broadbent, A., et al.: Demonstration of blind quantum computing. Science 335(6066), 303–308 (2012)

[14] Morimae, T., Fujii, K.: Secure entanglement distillation for double-server blind quantum computation. Phys Rev Lett 111(2), 20502 (2013)

[15] Sheng, Y.B., Zhou, L.: Deterministic entanglement distillation for secure double-server blind quantum computation. Sci Rep 5(1), 7815 (2015)

[16] Li, Q., Chan, W.H., Wu, C., et al.: Triple-server blind quantum computation using entanglement swapping. Phys Rev A 89(4), 40302 (2014)

[17] Kong, X., Li, Q., Wu, C., et al.: Multiple-server flexible blind quantum computation in networks. Int J Theor Phys 55(6), 3001–3007 (2016)

[18] Morimae, T., Fujii, K.: Blind quantum computation protocol in which Alice only makes measurements. Phys Rev A 87(5), 50301 (2013)

[19] Greganti, C., Roehsner, M.C., Barz, S., et al.: Demonstration of measurement-only blind quantum computing. New J Phys 18(1), 13020 (2016)

[20] Reichardt, B.W., Unger, F., Vazirani, U.V.: Classical command of quantum systems. Nature 496(7446), 456–460 (2013)

[21] Huang, H.L., Zhao, Q., Ma, X., et al.: Experimental blind quantum
VBQC with identity authentication for different types of clients

computing for a classical client. Phys Rev Lett 119(5), 50503 (2017)

[22] Perez-Delgado, C.A., Fitzsimons, J.F.: Iterated gate teleportation and blind quantum computation. Phys Rev Lett 114(22), 220502 (2015)

[23] Xu, H.R., Wang, B.H.: Universal single-server blind quantum computation for classical clients. Laser Phys Lett 19(1), 15202 (2022)

[24] Li, Q., Liu, C., Peng, Y.: Blind quantum computation where a user only performs single-qubit gates. Opt Laser Technol 142, 107190 (2021)

[25] Li, W., Lu, S., Deng, D.L.: Quantum federated learning through blind quantum computing. Sci China Phys Mech 64(10), 100312 (2021)

[26] Fitzsimons, J.F., Kashefi, E.: Unconditionally verifiable blind quantum computation. Phys Rev A 96(1), 12303 (2017)

[27] Fitzsimons, J.F., Hajdušek, M., Morimae, T.: Post hoc verification of quantum computation. Phys Rev Lett 120(4), 40501 (2018)

[28] Hajdusek, M., Pérez-Delgado, C.A., Fitzsimons, J.F.: Device-independent verifiable blind quantum computation. arXiv:1502.02563 (2015)

[29] Morimae, T., Takeuchi, Y., Hayashi, M.: Verification of hypergraph states. Phys Rev A 96(6), 62321 (2017)

[30] Takeuchi, Y., Morimae, T.: Verification of many-qubit states. Phys Rev X 8(2), 21060 (2018)

[31] Hayashi, M., Morimae, T.: Verifiable measurement-only blind quantum computing with stabilizer testing. Phys Rev Lett 115(22), 200502 (2015)

[32] Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum 2, 79 (2019)

[33] Zhang, J., Pagano, G., Hess, P., et al.: Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. Nature 551(7682), 601–604 (2017)

[34] Arute, F., Arya, K., Babbush, R., et al.: Quantum supremacy using a programmable superconducting processor. Nature 574(7779), 505–510 (2019)

[35] Wang, G.Y., Long, G.L.: Entanglement purification for memory nodes in a quantum network. Sci China Phys Mech 63(2), 220311 (2020)

[36] Liu, Z.H., Chen, H.W.: Universal and general quantum simultaneous secret distribution with dense coding by using one-dimensional high-level
VBQC with identity authentication for different types of clients

cluster states. J Comput Sci Tech 36(1), 221 (2021)

[37] Shi, W.J., Cao, Q.X., Deng, Y.X., et al.: Symbolic reasoning about quantum circuits in Coq. J Comput Sci Tech 36(6), 1291 (2021)

[38] Li, Q., Li, Z., Chan, W.H., et al.: Blind quantum computation with identity authentication. Phys Lett A 382(14), 938–941 (2018)

[39] Shan, R.T., Chen, X., Yuan, K.G.: Multi-party blind quantum computation protocol with mutual authentication in network. Sci China Inf Sci. 64(6), 1–14 (2021)

[40] Barends, R., Kelly, J., Megrant, A., et al.: Superconducting quantum circuits at the surface code threshold for fault tolerance. Nature 508(7497), 500–503 (2014)

[41] Li, K., Smith, G.: Quantum de Finetti theorem under fully-one-way adaptive measurements. Phys Rev Lett 114(16), 160503 (2015)

[42] Popescu, S., Rohrlich, D.: Quantum nonlocality as an axiom. Found Phys 24(3), 379–385 (1994)

[43] Sato, G., Koshiba, T., Morimae, T.: Arbitrable blind quantum computation. Quantum Inf Process 18(12), 370 (2019)

[44] Morimae, T., Dunjko, V., Kashefi, E.: Ground state blind quantum computation on AKLT state. Quantum Inf Comput 15(3), 200–234 (2015)

[45] Morimae, T., Fujii, K.: Blind topological measurement-based quantum computation. Nat Commun 3(1), 1036–1036 (2012)

[46] Kashefi, E., Wallden, P.: Optimised resource construction for verifiable quantum computation. J Phys A-Math Theor 50(14), 145306 (2017)