


Article

# A Distributed Architecture for Secure Delegated Quantum Computation

Shuquan Ma <sup>1</sup> , Changhua Zhu <sup>1,2,3,\*</sup>, Dongxiao Quan <sup>1</sup> and Min Nie <sup>3,4</sup>

<sup>1</sup> State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China; msqloveslife@outlook.com (S.M.); dxquan@xidian.edu.cn (D.Q.)

<sup>2</sup> Collaborative Innovation Center of Quantum Information of Shaanxi Province, Xidian University, Xi'an 710071, China

<sup>3</sup> Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China; niemin@xupt.edu.cn

<sup>4</sup> School of Communications and Information Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710121, China

\* Correspondence: chhzhu@xidian.edu.cn

**Abstract:** In this paper, we propose a distributed secure delegated quantum computation protocol, by which an almost classical client can delegate a ( $dk$ )-qubit quantum circuit to  $d$  quantum servers, where each server is equipped with a  $2k$ -qubit register that is used to process only  $k$  qubits of the delegated quantum circuit. None of servers can learn any information about the input and output of the computation. The only requirement for the client is that he or she has ability to prepare four possible qubits in the state of  $(|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$ , where  $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ . The only requirement for servers is that each pair of them share some entangled states  $(|0\rangle|+\rangle + |1\rangle|-\rangle)/\sqrt{2}$  as ancillary qubits. Instead of assuming that all servers are interconnected directly by quantum channels, we introduce a third party in our protocol that is designed to distribute the entangled states between those servers. This would simplify the quantum network because the servers do not need to share a quantum channel. In the end, we show that our protocol can guarantee unconditional security of the computation under the situation where all servers, including the third party, are honest-but-curious and allowed to cooperate with each other.

**Keywords:** quantum computation; secure delegated computation; distributed architecture



**Citation:** Ma, S.; Zhu, C.; Quan, D.; Nie, M. A Distributed Architecture for Secure Delegated Quantum Computation. *Entropy* **2022**, *24*, 794. <https://doi.org/10.3390/e24060794>

Academic Editors: Shao-Ming Fei, Ming Li and Shunlong Luo

Received: 7 May 2022

Accepted: 3 June 2022

Published: 7 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Quantum computing has been extensively studied from theory to practice [1,2]. It is widely accepted that noisy intermediate-scale quantum (NISQ) computers may be available in the coming decades [3]. However, the limited quantum memory of NISQ devices means that they may not have the capability to deal with large-scale quantum information processing. This is obviously a severe constraint, as many practical problems, e.g., *machine learning*, usually require immense memory overhead. A feasible way to overcome this obstacle is to utilize *distributed architecture* for quantum computations [4]. That is, using a group of small-scale quantum computers interconnected by classical and quantum networks to implement large-scale quantum computation tasks. However, considering the tremendous cost of building a quantum computer, it is not likely that ordinary consumers will be able to afford an NISQ computer in the foreseeable future. In fact, it is widely believed that the role of quantum computers is similar to today's classical supercomputers, which means only a few organizations or enterprises can have quantum computers at their disposal. Thus, for ordinary customers, a better way to access quantum computers is to delegate their computations to the companies that offer quantum computing as cloud services. Indeed, this computation pattern has been applied in today's Internet, e.g., IBM Quantum platform [5].

Delegated quantum computation is actually closely related to distributed quantum computation [4]. The client-to-server pattern in delegated computation naturally belongs

to the category of distributed quantum computation. A class of delegated quantum computation protocols are constructed under the framework of measurement-based quantum computation (MBQC) [6–8], which is driven by a sequence of single-qubit measurements on some specific entangled state, where the entangled resource is also a basic module in the distributed quantum computation. Another class of delegated quantum computation protocols are obtained using the technique *quantum computing on encrypted data* (QCED) [9] or *quantum homomorphic encryption* (QHE) [10]. Although QCED and QHE are distinct concepts, the basic idea behind them is identical. Both of them use the *quantum one-time pad* to encrypt the input and output states but use different the methods to achieve the non-Clifford gates. Nevertheless, most schemes use the entangled states as the ancillary resources, for example [10–12].

Both distributed quantum computation and delegated quantum computation have been investigated broadly; see references [13–21] and [6,11,22–28], respectively. Typically, the distributed architecture for quantum computation makes use of photons as *flying qubits* between computational nodes, where each node is equipped with a quantum computer. The flying qubits are usually used to generate entangle states between distinct servers (i.e., nodes). By means of quantum entanglement, the non-local operations, such as controlled-NOT gate, can be done between two distant servers. Note that the quantum computer in each server is not necessarily an optical quantum computer; it can be made up of some other quantum system [29], such as ion traps or cloud atoms. Related experiments have been successfully demonstrated (see references [30,31]). Recently, researchers also investigated the possibility of simulating large-scale quantum systems in a hybrid quantum-classical manner [32]. That is, using a classical computer combined with a small quantum computer to simulate a large quantum computer [33]. However, the computational model considered in [32,33] is slightly different from the traditional model of circuit-based quantum computation. In this paper, we will not consider the method in [32], but rather the quantum entanglement to implement the non-local operation. In general, delegated quantum computation refers specifically to the *secure delegated quantum computation* (SDQC), which requires that no one except the client can obtain the right input and output of the computation. Typically, the client is required to have some basic quantum capacities, for example, preparing some single qubits or performing single-qubit measurements. In [34], the authors proposed a more rigorous SDQC protocol, which they called *universal blind quantum computation* (UBQC). The new protocol can guarantee that not only the input and output but also the computation itself, i.e., the algorithm, are unknown to the server. Although it seems that UBQC is more secure than SDQC, they are equivalent. That is, SDQC can be converted into UBQC [35]. As delegated quantum computation protocols effectively release the quantum resources in the client side, related experimental demonstrations have rapidly been implemented using the linear optics components (see References [9,25,36,37]).

Based on the above observations, in this paper we formally propose a distributed secure delegated quantum computation protocol that allows a half-classical client who can only prepare special single qubits to implement a large-scale quantum circuit on several quantum servers interconnected by entangled channels. Each server only has a limited quantum memory so that it can only compute a fraction of the delegated circuit. Moreover, during the computation, servers get nothing about the input and output of the computation. We also give a detailed security proof for our protocol. The rest of this paper is organized as follows. Section 2 introduces some basic preliminaries and notation. Section 3 presents the basic modules for delegated quantum computation. Section 4 gives the complete distributed delegated quantum computation protocol. Section 5 analyzes the security of our protocol. The last section discusses some remaining problems in our work.

### 2. Preliminaries and Notation

We assume that readers are familiar with the basics of quantum computation. In this work, we will use the following basic quantum gates:

$$Z |s\rangle = e^{is\pi} |s\rangle, \tag{1}$$

$$X |s\rangle = |s \oplus 1\rangle, \tag{2}$$

$$H |s\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{is\pi} |1\rangle), \tag{3}$$

$$P |s\rangle = e^{i\frac{s}{2}\pi} |s\rangle, \tag{4}$$

$$T |s\rangle = e^{i\frac{s}{4}\pi} |s\rangle, \tag{5}$$

$$CZ |s, t\rangle = e^{ist\pi} |s, t\rangle, \tag{6}$$

where  $s, t \in \{0, 1\}$  and  $i = \sqrt{-1}$ ;  $P$  and  $T$  refer to the phase gate and the  $\pi/8$  gate, respectively; and  $CZ$  denotes the controlled- $Z$  gate. In order to analyze conveniently, we also introduce the  $Z$ -rotation operator defined as follows:

$$R_z(\alpha) = \begin{pmatrix} e^{-i\frac{\alpha}{2}} & 0 \\ 0 & e^{i\frac{\alpha}{2}} \end{pmatrix}, \tag{7}$$

where  $\alpha \in [0, 2\pi)$  is referred as the *rotation angle*. Regardless of the global phases, we can see that  $Z \equiv R_z(\pi)$ ,  $P \equiv R_z(\frac{\pi}{2})$ , and  $T \equiv R_z(\frac{\pi}{4})$ . We use  $|+\varphi\rangle$  to denote the following single qubit:

$$|+\varphi\rangle = \frac{|0\rangle + e^{i\varphi\pi} |1\rangle}{\sqrt{2}}, \tag{8}$$

where we consider  $\varphi \in [0, 2\pi)$ . It is clear that, up to an unimportant global phase,  $R_z(\alpha) |+\varphi\rangle \equiv |+(\varphi+\alpha)\rangle$ . Thus,  $\varphi$  is also called as the *rotation angle*. By this definition, we can see that  $|+\rangle = |+\_0\rangle$  and  $|-\rangle = |+\_\pi\rangle$ . Note that for any  $\theta \in [0, 2\pi)$  the states  $|+\_\theta\rangle$  and  $|+\_{(\theta+\pi)}\rangle$  comprise a basis, thus we can define a single-qubit measurement operator as follows:

$$M(\theta) = \sum_{s \in \{0,1\}} (-1)^s |+\_{(\theta+s\pi)}\rangle \langle +\_{(\theta+s\pi)}|, \tag{9}$$

where  $\theta$  is referred as the *measurement angle* in this case, and  $s \in \{0, 1\}$  denotes the classical measurement outcome. Specifically,  $s = 0$  if the post-measurement state is  $|+\_\theta\rangle$ , otherwise  $s = 1$ . Finally, in this work we will also use a special two-qubit entangled state defined as follows:

$$|H\rangle = \frac{|0\rangle |+\rangle + |1\rangle |-\rangle}{\sqrt{2}}, \tag{10}$$

which can be prepared by applying a  $CZ$  gate on two qubits  $|+\rangle |+\rangle$ .

### 3. Secure Delegated Quantum Computation

In this work, the delegated quantum computation model we adopt is from [38], in which the authors improved the original QCED protocol [11] in two aspects. First, the quantum capacities of clients are further reduced. In theory, they only need to prepare the qubits  $|+\varphi\rangle$ , where  $\varphi \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ . Second, the security of the protocol can be still guaranteed even if some information is leaked to servers.

First of all, we specify that the client's input is encoded in  $X$  basis. That is, encoding 0 and 1 as  $|+\rangle$  and  $|-\rangle$ , respectively. Let  $x = x_1x_2 \cdots x_n \in \{0, 1\}^n$  be the  $n$ -bit classical input string, then the corresponding encoded input state can be expressed as  $|+\_{x\pi}\rangle \equiv |+\_{x_1\pi}\rangle |+\_{x_2\pi}\rangle \cdots |+\_{x_n\pi}\rangle$ . For simplicity, we abbreviate  $|+\_{x\pi}\rangle$  as  $|+\_x\rangle$ . The universal gate set we consider is  $\mathbb{U} = \{X, Z, P, T, H, CZ\}$ . Note that this gate set is not minimal because  $X, Z$ , and  $P$  can be obtained from  $\{T, H\}$ . Despite that, additional basic gates can effectively decrease the circuit complexity.

Now suppose the client’s input state is  $|+x\rangle$ , where  $x \in \{0, 1\}^n$ . In [38], the client uses the random operator  $X_i^{a_i} Z_i^{b_i} P_i^{c_i}$  to encrypt each qubit  $|+x_i\rangle$ , where  $x_i \in \{0, 1\}$ , and  $a_i, b_i, c_i \in \{0, 1\}$  are referred as the *encryption keys*, and for any operator  $U$  we define  $U^0 = I$  and  $U^1 = U$ . The subscript  $i$  in  $X_i, Z_i$ , and  $P_i$  is used to denote that the corresponding gate is applied on the  $i$ th qubit (hereinafter referred to as qubit  $i$ ). Similarly, the subscript  $i$  in  $a_i, b_i, c_i$  is used to denote that the corresponding encryption keys are related to qubit  $i$ . We can check that this encryption scheme is a quantum one-time pad (see Equation (11)), thus it provides an information-theoretical security for any qubit  $\rho$ .

$$\frac{1}{4} \sum_{a,b,c \in \{0,1\}} X^a Z^b P^c \rho P^{3c} Z^b X^a = \frac{I}{2}. \tag{11}$$

In theory, to achieve this encryption, the client needs to perform random gates  $P^c, Z^b$ , and  $X^a$  on the state  $\rho$  in sequence. However, for the qubit  $|+x_i\rangle$ , it can be easily verified that

$$X^{a_i} Z^{b_i} P^{c_i} |+x_i\rangle \equiv |+ \varphi_i\rangle, \tag{12}$$

where  $\varphi_i = (-1)^{a_i}(x_i + b_i + \frac{c_i}{2})\pi \pmod{2\pi} \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ . Thus, instead of preparing  $|+x_i\rangle$  then encrypting it by  $X_i^{a_i} Z_i^{b_i} P_i^{c_i}$ , the client can directly generate the encrypted qubit. Specifically, given the  $i$ th input bit  $x_i \in \{0, 1\}$ , the client randomly chooses the corresponding encryption keys  $a_i, b_i, c_i \in \{0, 1\}$ , then computes the value  $\varphi_i = (-1)^{a_i}(x_i + b_i + \frac{c_i}{2})\pi \pmod{2\pi}$ . Finally, the client prepares the qubit  $|+ \varphi_i\rangle$  as the encrypted qubit  $i$ .

After preparing all encrypted input qubits, the client sends them to the server. The server then performs the delegated quantum circuit  $U$  on the encrypted qubits. Here, the circuit  $U$  is known to both client and server (they can negotiate in advance via a classical channel). We assume that this circuit has been decomposed into a sequence of basic gates from the gate set  $\mathbb{U}$ . That is,  $U = U_m U_{m-1} \cdots U_2 U_1$ , where each  $U_i \in \mathbb{U}$  and the positive integer number  $m$  is the total number of gates. The following identities, which all hold up to an irrelevant global phase, can be easily verified.

$$X_i(X_i^{a_i} Z_i^{b_i} P_i^{c_i}) \equiv (X_i^{a_i} Z_i^{b_i \oplus c_i} P_i^{c_i}) X_i, \tag{13}$$

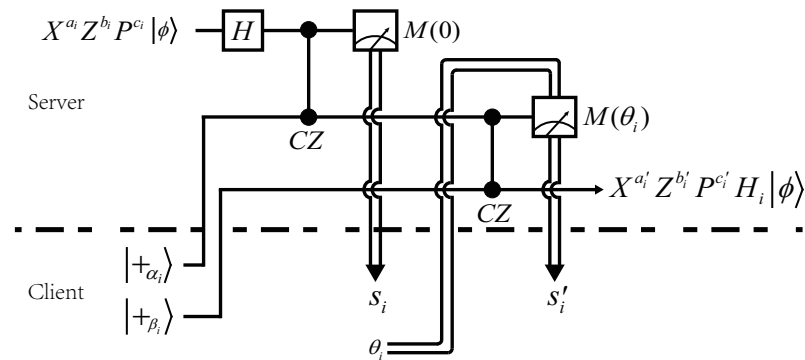
$$Z_i(X_i^{a_i} Z_i^{b_i} P_i^{c_i}) \equiv (X_i^{a_i} Z_i^{b_i} P_i^{c_i}) Z_i, \tag{14}$$

$$P_i(X_i^{a_i} Z_i^{b_i} P_i^{c_i}) \equiv (X_i^{a_i} Z_i^{a_i \oplus b_i} P_i^{c_i}) P_i, \tag{15}$$

$$T_i(X_i^{a_i} Z_i^{b_i} P_i^{c_i}) \equiv (X_i^{a_i} Z_i^{a_i \oplus b_i \oplus (a_i c_i)} P_i^{a_i \oplus c_i}) T_i, \tag{16}$$

$$CZ_{i,j}(X_i^{a_i} Z_i^{b_i} P_i^{c_i} X_j^{a_j} Z_j^{b_j} P_j^{c_j}) \equiv (X_i^{a_i} Z_i^{a_j \oplus b_i} P_i^{c_i} X_j^{a_j} Z_j^{a_i \oplus b_j} P_j^{c_j}) CZ_{i,j}, \tag{17}$$

It follows from Equations (13)–(17) that the basic gates  $X, Z, P, T, CZ$  are *commutable* with the encryption operator  $X^a Z^b P^c$ , although the encryption keys may need to be updated. For example, Equation (13) indicates that performing an  $X_i^{a_i} Z_i^{b_i} P_i^{c_i}$  followed by an  $X_i$  is equivalent to performing an  $X_i$  followed by an  $X_i^{a_i} Z_i^{b_i \oplus c_i} P_i^{c_i}$ . Thus, the client only needs to update the value of  $b_i$  such that  $b_i := b_i \oplus c_i$ . The cases for  $Z_i, P_i, T_i$ , and  $CZ_{i,j}$  follow the same reason. The related updating rules of encryption keys are shown in Equations (14)–(17). Note, however, that the commutativity noted above is not suited for the Hadamard gate  $H$ , as there is no  $HP^c \equiv P^{c'}H$  for any  $c, c' \in \{0, 1\}$ . In [38], the authors proposed a quantum teleportation scheme that they called the  $H$ -gadget (see Figure 1) so as to implement the  $H$  gate in a similar manner. Specifically, the client needs to prepare two ancillary qubits  $|+\alpha_i\rangle, |+\beta_i\rangle$  and a measurement angle  $\theta_i$ , where  $\alpha_i$  and  $\beta_i$  are chosen randomly, whereas  $\theta_i$  can be determined by the following way.



**Figure 1.** The  $H$ -gadget in Ref. [38], which is designed for implementing an  $H$  gate on an encrypted qubit  $i$ , where  $s_i, s'_i \in \{0, 1\}$  are the measurement outcomes and  $\alpha_i, \beta_i \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$  are the rotation angles of two ancillary qubits, and  $\theta_i \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$  is the measurement angle of the second measurement.

Note that for any  $\alpha_i, \beta_i \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ , we can express them uniquely as follows:

$$\alpha_i = (d_i + \frac{e_i}{2})\pi, \beta_i = (f_i + \frac{g_i}{2})\pi, \tag{18}$$

where  $d_i, e_i, f_i, g_i \in \{0, 1\}$ . Thus, the client can first generate random bits  $d_i, e_i, f_i, g_i$  then compute the values of  $\alpha_i$  and  $\beta_i$ . To determine  $\theta_i$ , the client generates a random bit, denoted by  $h_i \in \{0, 1\}$ , then computes  $\theta_i$  such that

$$\theta_i = [h_i \oplus b_i \oplus d_i \oplus (a_i c_i) \oplus (s_i c_i) \oplus (c_i e_i)]\pi + \frac{c_i \oplus e_i}{2}\pi. \tag{19}$$

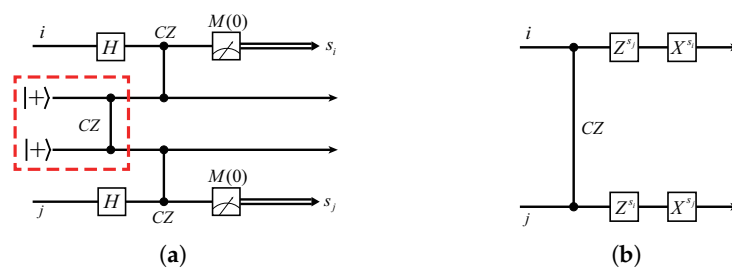
Note also that  $\theta_i$  is relevant to the measurement outcome  $s_i$ , which means it can be determined until the client obtains the first measurement outcome  $s_i$  from the server. Nevertheless, in theory, all qubits including ancillary qubits can be sent to the server before the computation begins. Thus, the complete procedure is classically interactive. Finally, the updating rule for  $H$  is shown as follows:

$$a'_i = s'_i \oplus h_i, b'_i = a_i \oplus s_i \oplus f_i \oplus [g_i(s'_i \oplus h_i)], c'_i = g_i, \tag{20}$$

where  $a'_i, b'_i, c'_i$  denote the updated encryption keys related to qubit  $i$ . The correctness of the  $H$ -gadget is given in the Appendix A. The detailed security proof of the protocol can be found in [38].

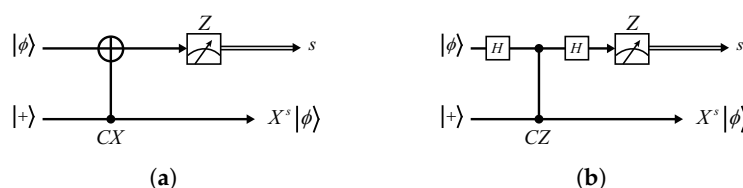
#### 4. Distributed Architecture for Secure Delegated Quantum Computations

In this section, we give a simple scheme to implement the non-local CZ gate between two quantum servers. Our method uses the entangled state  $|H\rangle$  (see Equation (10) for its definition) as ancillary qubits. The similar schemes have been studied intensively, for example, in [39,40]. The basic circuit is shown in Figure 2a. In the following content, we first verify the circuit identity shown in Figure 2, then, based on this circuit identity, we construct a distributed architecture for secure delegated quantum computations.



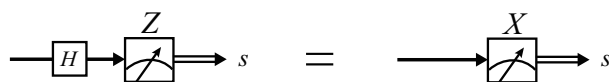
**Figure 2.** (a) The basic circuit used to implement a non-local CZ gate on two distant qubits  $i$  and  $j$ , where the partial circuit in the red dotted box is used to generate the entangled state  $|H\rangle$ . (b) The equivalent quantum circuit for (a).

We start with a circuit named *X-teleportation* [40] (see Figure 3a), which is easy to verify.

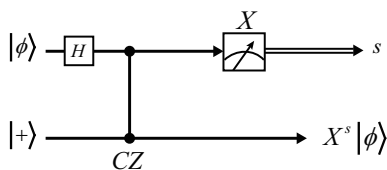


**Figure 3.** (a) The original X-teleportation in [40]; (b) the X-teleportation that replaces the CX with a CZ and two H gates. In both circuits, the measurement is performed under Z basis.

First, we substitute a CZ and two H gates for the CX gate, obtaining the equivalent circuit, as shown in Figure 3b. We then convert the measurement basis from Z to X by the following identity (see Figure 4), which is also easy to verify. Finally, we obtain a variant of the X-teleportation that consists of H, CZ, and X-basis measurement, as shown in Figure 5.



**Figure 4.** Measurement identity that converts Z-basis to X-basis.



**Figure 5.** The variant X-teleportation consisting of CZ and H gates, where the measurement basis is X.

We now turn back to Figure 2a. Note first that the CZ gate commutes with itself, thus the circuit can be reorganized, as in Figure 6a. Obviously, the partial circuits in the red-dotted line and blue-dotted line boxes are exactly the same circuit as the one in Figure 5, where  $X = M(0)$ . Therefore, we can see that, after measuring qubits  $i, j$ , the rest qubits and the rest CZ gate comprise the circuit as, in Figure 6b. Finally, we use the following identity to exchange the positions of X and CZ, which can be easily verified:

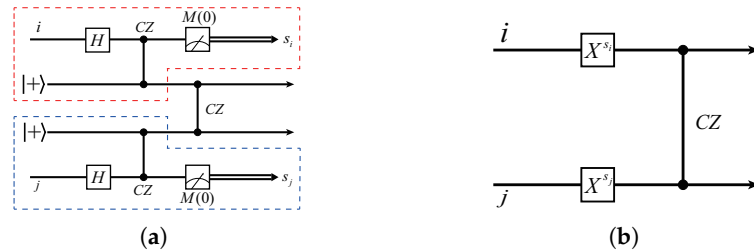
$$CZ \cdot (X^s \otimes I) = (X^s \otimes Z^s) \cdot CZ, \tag{21}$$

where  $s \in \{0, 1\}$ . Substituting the above identity in Figure 6b and considering the symmetry of CZ gate, we immediately obtain the desired circuit, as shown in Figure 2b.

Considering the encryption operators  $X_i^{a_i} Z_i^{b_i} P_i^{c_i}$  and  $X_j^{a_j} Z_j^{b_j} P_j^{c_j}$  on qubits  $i$  and  $j$ , we can see from Figure 6b that the non-local CZ can be thought to be performed on qubits  $i, j$ , which are encrypted by  $X_i^{a_i \oplus s_i} Z_i^{b_i} P_i^{c_i}$  and  $X_j^{a_j \oplus s_j} Z_j^{b_j} P_j^{c_j}$ , thus according to the updating rule

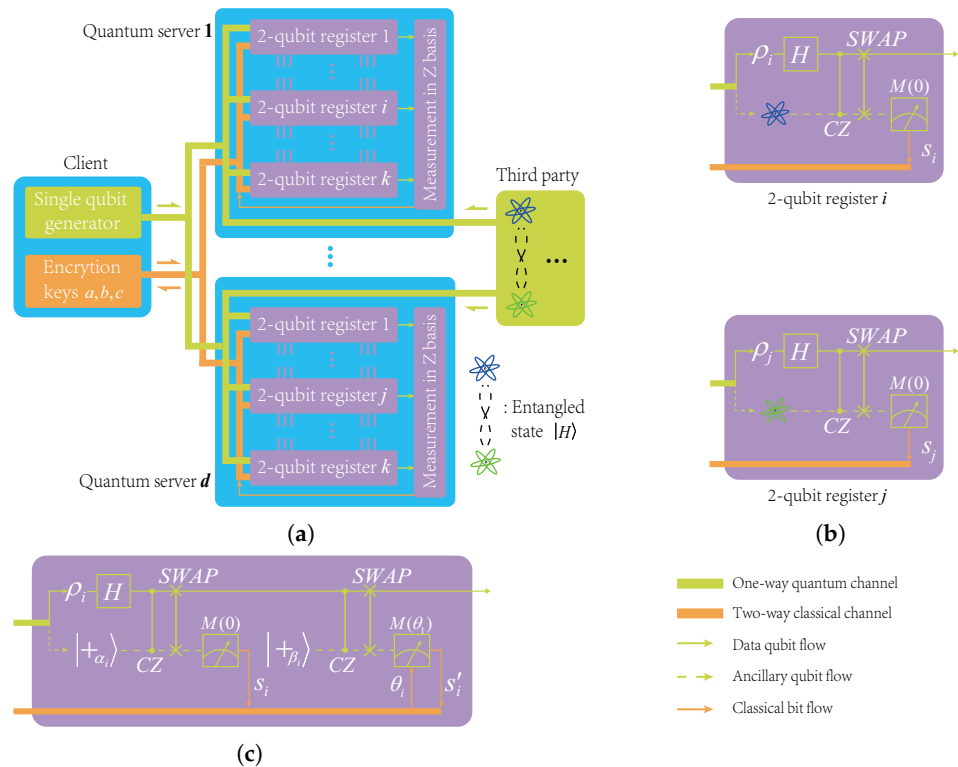
shown in Equation (17), we immediately obtain the updating rule of the non-local CZ gate as follows:

$$\begin{cases} a'_i = a_i \oplus s_i, \\ b'_i = a_j \oplus s_j \oplus b_i, \\ c'_i = c_i, \end{cases} \quad \begin{cases} a'_j = a_j \oplus s_j, \\ b'_j = a_i \oplus s_i \oplus b_j, \\ c'_j = c_j. \end{cases} \quad (22)$$



**Figure 6.** (a) The equivalent form of the circuit shown in Figure 2a. (b) The resulting circuit after measuring qubits  $i, j$ .

Based on the above analysis, we construct a distributed architecture for secure delegated quantum computation, where a classical client equipped with some qubit generator can delegate an  $n$ -qubit circuit to  $d$  small-scale quantum servers. Without loss of generality, we assume that  $n = dk$ . In this configuration, each server typically needs a  $2k$ -qubit register to process  $k$  input qubits of the  $n$ -qubit circuit. That is, for each qubit in the  $n$ -qubit circuit, the server needs a 2-qubit register to simulate it. To make sure  $2k < n$ , it requires that  $d > 2$ . We show this distributed architecture in Figure 7. Note that there is a special third party in this distributed architecture, which is used to generate and distribute entangled states  $|H\rangle$  between all quantum servers. Thus, all servers do not need to be interconnected directly by a quantum (even classical) channel, as there is no information exchange between servers during the computation.



**Figure 7.** (a) The distributed architecture for secure delegated quantum computations; (b) the circuits for a CZ gate between two nonlocal registers  $i$  and  $j$ ; (c) the circuit for an  $H$  gate in any register  $i$ .

We give the complete procedure of the protocol in terms of pseudo-code (see Algorithms 1–3). For simplicity, we use  $\mathcal{C}$  and  $\{\mathcal{S}_q\}_{q=1}^d$  to denote the client and  $d$  servers, respectively. That is, the  $q$ th quantum server is referred to as  $\mathcal{S}_q$ . As noted, each server only processes  $k$  input qubits of the  $n$ -qubit delegated circuit. More specifically, for  $\mathcal{S}_q$ , it only processes the qubits indexed by  $(q - 1)k + 1, (q - 1)k + 2, \dots, qk$ . Thus, in the case of no confusion, we also use  $\mathcal{S}_q = \{(q - 1)k + 1, (q - 1)k + 2, \dots, qk\}$  to denote the corresponding qubits. In addition, the delegated circuit  $U$  is formally expressed as  $U = U_m^{p_m} U_{m-1}^{p_{m-1}} \dots U_1^{p_1}$ , where  $p_i \subset \{1, 2, \dots, n\}$  denotes the qubits on which the basic gate  $U_i$  is exerted. For example, if  $U_i^{p_i}$  is a CZ gate on qubits  $k$  and  $l$ , then  $p_i = \{k, l\}$ . By this definition, we can see that there must be  $p_i \subset \mathcal{S}_q$  if  $U_i^{p_i}$  is a local gate in  $\mathcal{S}_q$ , otherwise it only can be  $p_i \subset \mathcal{S}_q \cup \mathcal{S}_{q'}$  for some  $\mathcal{S}_q$  and  $\mathcal{S}_{q'}$ .

---

**Algorithm 1** Distributed Secure Delegated Quantum Computations

---

**Input:**  $x = x_1 x_2 \dots x_n$  // private against all  $\mathcal{S}_q$   
 $U = U_m^{p_m} U_{m-1}^{p_{m-1}} \dots U_1^{p_1}$  // public for  $\mathcal{C}$  and all  $\mathcal{S}_q$

**Output:**  $y = y_1 y_2 \dots y_n$  // private against all  $\mathcal{S}_q$

- 1:  $\mathcal{C}$  generates  $a, b, c \leftarrow_R \{0, 1\}^n$  and computes rotation angles  $(\varphi_1, \dots, \varphi_n)$  according to Equation (12), then prepares  $|+\varphi_1\rangle \dots |+\varphi_n\rangle$  as the encrypted input state, finally sends the qubits  $(q - 1)k + 1, q(k - 1) + 2, \dots, qk$  to  $\mathcal{S}_q$  where  $q = 1, 2, \dots, d$ . Specifically,  $\mathcal{C}$  sends the qubits  $1, 2, \dots, k$  to  $\mathcal{S}_1$  then sends the qubits  $k + 1, k + 2, \dots, 2k$  to  $\mathcal{S}_2$ , and so on
- 2: **for**  $i \leftarrow 1, m$  **do**
- 3:   **if**  $U_i^{p_i} \in \{X, Z, P, T, H\}$  and  $p_i \subset \mathcal{S}_q$  for some  $q \in \{1, 2, \dots, d\}$  **then**
- 4:     **if**  $U_i^{p_i}$  is not  $H$  **then**
- 5:        $\mathcal{S}_q$  performs  $U_i^{p_i}$  on qubit  $p_i$  while  $\mathcal{C}$  updates the encryption keys of this qubit according to the updating rules shown in Equations (13)–(16)
- 6:     **else**
- 7:        $\mathcal{C}$  calls the **procedure** HADAMARD( $p_i, q$ ) (See Algorithm 2)
- 8:     **end if**
- 9:   **else** //  $U_i^{p_i}$  is a CZ gate on qubits  $p_i$
- 10:     **if**  $p_i \subset \mathcal{S}_q$  for some  $q \in \{1, 2, \dots, d\}$  **then**
- 11:        $\mathcal{S}_q$  performs  $U_i^{p_i}$  on qubits  $p_i$  while  $\mathcal{C}$  updates the encryption keys of those qubits according to the updating rule shown in Equation (17)
- 12:     **else** //  $p_i \subset \mathcal{S}_q \cup \mathcal{S}_{q'}$  for some  $q, q' \in \{1, 2, \dots, d\}$
- 13:        $\mathcal{C}$  calls the **procedure** NONLOCAL-CZ( $p_i, q, q'$ ) (See Algorithm 3)
- 14:     **end if**
- 15:   **end if**
- 16: **end for**
- 17: Each server measures the final  $k$  qubits in  $Z$  basis, then sends the measurement outcomes to  $\mathcal{C}$  // let  $\tilde{y} \in \{0, 1\}^n$  be the result collected from all servers
- 18:  $\mathcal{C}$  computes the output  $y = \tilde{y} \oplus a$ . //  $a$  is the  $X$  encryption keys of the final state

---



**Algorithm 2** Implement an  $H$  gate on qubit  $i$  where  $i$  is in  $\mathcal{S}_q$ 

- 1: **procedure** HADAMARD( $i, q$ ) // qubit  $i$  is encrypted by  $X^{a_i} Z^{b_i} P^{c_i}$
- 2:  $\mathcal{C}$  generates  $d_i, e_i \leftarrow_R \{0, 1\}$  and computes the angle  $\alpha_i$  according to Equation (18), then prepares and sends the ancillary qubit  $|+\alpha_i\rangle$  to  $\mathcal{S}_q$
- 3:  $\mathcal{S}_q$  performs  $H_i$  and CZ gates on qubit  $i$  and  $|+\alpha_i\rangle$ , then measures qubit  $i$  and sends the measurement outcome  $s_i$  to  $\mathcal{C}$ , finally labels the ancillary qubit as  $i$
- 4:  $\mathcal{C}$  generates  $f_i, g_i, h_i \leftarrow_R \{0, 1\}$  and computes the angles  $\beta_i$  and  $\theta_i$  according to Equations (18) and (19), respectively, then prepares the ancillary qubit  $|+\beta_i\rangle$  and sends it with  $\theta_i$  to  $\mathcal{S}_q$
- 5:  $\mathcal{S}_q$  performs a CZ gate on qubit  $i$  and  $|+\beta_i\rangle$ , then measures qubit  $i$  with  $M(\theta_i)$  and sends the measurement outcome  $s'_i$  to  $\mathcal{C}$ , finally labels the ancillary qubit as  $i$
- 6:  $\mathcal{C}$  updates the encryption keys of qubit  $i$  according to Equation (20)
- 7: **end procedure**

**Algorithm 3** Implement a nonlocal CZ gate on qubits  $i$  and  $j$  where  $i$  is in  $\mathcal{S}_q$  while  $j$  is in  $\mathcal{S}_{q'}$ , that is,  $\{i, j\} \subset \mathcal{S}_q \cup \mathcal{S}_{q'}$ 

- 1: **procedure** NONLOCAL-CZ( $\{i, j\}, q, q'$ ) // qubits  $i$  and  $j$  are encrypted by  $X^{a_i} Z^{b_i} P^{c_i}$  and  $X^{a_j} Z^{b_j} P^{c_j}$ , respectively
- 2:  $\mathcal{C}$  delegates the third party to prepare an entangled state  $|H\rangle$  and distribute it to  $\mathcal{S}_q$  and  $\mathcal{S}_{q'}$ , that is, each server holds one qubit of  $|H\rangle$  as the ancillary qubit
- 3:  $\mathcal{S}_q$  ( $\mathcal{S}_{q'}$ ) performs  $H_i$  ( $H_j$ ) and CZ gates on qubit  $i$  ( $j$ ) and its ancillary qubit, then measures qubit  $i$  ( $j$ ) and sends the measurement outcome  $s_i$  ( $s_j$ ) to  $\mathcal{C}$ , finally labels its ancillary qubit as  $i$  ( $j$ )
- 4:  $\mathcal{C}$  updates the encryption keys of qubits  $i$  and  $j$  according to Equation (22)
- 5: **end procedure**

**5. The Security of the Distributed Delegated Quantum Computation**

We show that our protocol can guarantee the unconditional privacy of the input and output of the computation. We only consider that all servers and the third party who serves as an entanglement resource are *honest-but-curious*, which means they follow the algorithm honestly but try to obtain the information about the input and output. For example, they may record all classical information generated during the computation and cooperate with each other, even with the third party.

For the input, the conclusion is obvious as the client encrypts each input qubit by a quantum one-time pad. Therefore, to complete the proof, we only need to prove that the output state of the computation is also encrypted by a *unbiased* quantum one-time pad. In other words, there is no information leakage about the encryption keys during the computation. From the procedures of Algorithm 1, we can see that only when the client calls the **procedures** HADAMARD and NONLOCAL-CZ will there be an interaction between client and servers. In the other cases, the algorithm is non-interactive, which means there is no information leakage about the encryption keys from client to server as they do not exchange any information. Based on this observation, we infer that to prove the privacy we only need to analyze the procedures that implement the  $H$  and the nonlocal CZ gates.

We first consider the **procedure** HADAMARD( $i, q$ ). In the following content, we use  $\mathcal{S}$  to denote all servers including the *untrusted* third party. According to Algorithm 2, we can see that given the qubit  $i$  encrypted by  $X^{a_i} Z^{b_i} P^{c_i}$  where  $i \in \mathcal{S}_q$ ,  $\mathcal{S}$  controls two ancillary qubits  $Z^{d_i} P^{e_i} |+\rangle$  and  $Z^{f_i} P^{g_i} |+\rangle$ , and receives a measurement angle  $\theta_i$  from  $\mathcal{C}$ , it also generates two measurement outcomes  $s_i, s'_i \in \{0, 1\}$  from two independent measurements. We can infer from the below state evolution that the measurement outcomes  $s_i, s'_i$  are uniformly random, thus  $\mathcal{S}$  can obtain no information gain about any encryption keys according to  $s_i$  and  $s'_i$ .

$$|\phi\rangle |+\rangle \xrightarrow{H \otimes I} (H|\phi\rangle) |+\rangle \xrightarrow{\text{CZ}} \frac{|+\rangle}{\sqrt{2}} |\phi\rangle + \frac{|-\rangle}{\sqrt{2}} X|\phi\rangle. \quad (23)$$

The only available information to  $\mathcal{S}$  now is the measurement angle  $\theta_i$ . Let  $\theta_i$  be  $u_i\pi + \frac{v_i\pi}{2}$ , where  $u_i, v_i \in \{0, 1\}$ , then according to Equation (19), we know that  $u_i$  and  $v_i$  can be expressed as follows:

$$u_i = h_i \oplus b_i \oplus d_i \oplus (a_i c_i) \oplus (s_i c_i) \oplus (c_i e_i), \tag{24a}$$

$$v_i = c_i \oplus e_i, \tag{24b}$$

where  $u_i, v_i$ , and  $s_i$  are known to  $\mathcal{S}$ . Intuitively, given  $u_i, v_i$ , and  $s_i$ , no server can determine the correct values of  $a_i, b_i, c_i, d_i, e_i, h_i$ , as there are six variables in two equations. Nevertheless,  $\mathcal{S}$  may gain some information utilizing  $u_i$  and  $v_i$ . For example, if  $v_i = 1$ , then  $\mathcal{S}$  can infer that  $c_i e_i = 0$ . Substituting this into Equation (24a),  $\mathcal{S}$  can obtain a simplified equality  $u_i = h_i \oplus b_i \oplus d_i \oplus (a_i \oplus s_i) c_i$ . Despite this fact, we can show that there is no information leakage about all variables from  $a_i$  to  $h_i$ . That is, we prove that in the view of  $\mathcal{S}$ , the following equality holds true:

$$\Pr[r_i | u_i, v_i] = \Pr[r_i] = \frac{1}{2}, \tag{25}$$

where the random variable  $r_i$  represents the possible parameters  $\{a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i\}$ . To see that, we need to know the following simple facts.

First, if  $x, y \in \{0, 1\}$  and  $x$  is uniform, i.e.,  $x \in_R \{0, 1\}$ , then  $x \oplus y$  is also uniform. Second, if  $x, y \in \{0, 1\}$  are uniform and let  $z = x \oplus y$ , then  $\Pr[x|z] = \Pr[x] = 1/2$ . Finally, if  $x, y_1, y_2 \in \{0, 1\}$  and  $x$  is uniform, let  $z = x \oplus (y_1 y_2)$ , then  $\Pr[y_1|z] = \Pr[y_1]$ . These three basic facts can be easily verified. With these facts, we can complete our proof. Define  $\zeta_i = b_i \oplus d_i \oplus (a_i c_i) \oplus (s_i c_i) \oplus (c_i e_i)$  so that  $u_i = h_i \oplus \zeta_i$ . As  $b_i, d_i \in_R \{0, 1\}$ , we first know that  $\zeta_i \in_R \{0, 1\}$ . Furthermore, as  $h_i, \zeta_i \in_R \{0, 1\}$ , we can get that  $\Pr[h_i|u_i] = \Pr[h_i] = 1/2$ . Likewise, we can also get  $\Pr[b_i|u_i] = \Pr[b_i] = 1/2$  and  $\Pr[d_i|u_i] = \Pr[d_i] = 1/2$ . For  $a_i \in_R \{0, 1\}$ , define  $\xi_i = h_i \oplus b_i \oplus d_i \oplus (s_i c_i) \oplus (c_i e_i)$  so that  $u_i = \xi_i \oplus (a_i c_i)$ , from which we can infer that  $\Pr[a_i|u_i] = \Pr[a_i] = 1/2$ . Note that  $h_i, b_i, d_i$ , and  $a_i$  are irrelevant to  $v_i$ , which means  $\Pr[r_i|u_i, v_i] = \Pr[r_i|u_i]$  for any  $r_i \in \{h_i, b_i, d_i, a_i\}$ . As for  $c_i, e_i \in_R \{0, 1\}$ , as they are related to both  $u_i$  and  $v_i$ , in order to simplify our analysis, we define  $h'_i = h_i \oplus (a_i c_i)$ ,  $b'_i = b_i \oplus (s_i c_i)$ , and  $d'_i = d_i \oplus (c_i e_i)$ , then obtain that  $u_i = h'_i \oplus b'_i \oplus d'_i$ . Clearly,  $h'_i, b'_i, d'_i \in_R \{0, 1\}$ , so  $c_i$  and  $e_i$  are only related to  $v_i$ . By this, we can easily get that  $\Pr[c_i|u_i, v_i] = \Pr[c_i|v_i] = \Pr[c_i] = 1/2$  and  $\Pr[e_i|u_i, v_i] = \Pr[e_i|v_i] = \Pr[e_i] = 1/2$ . Finally,  $f_i$  and  $g_i \in_R \{0, 1\}$  are obviously irrelevant to  $u_i$  and  $v_i$  (see Equations (24a) and (24b)), which means  $\Pr[f_i|u_i, v_i] = \Pr[f_i] = 1/2$  and  $\Pr[g_i|u_i, v_i] = \Pr[g_i] = 1/2$ . So far, we have proved the statement in Equation (25), from which we know that the servers can obtain no information gain about  $a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i$  from the  $\theta_i$ . Thus, after the **procedure** HADAMARD( $i, q$ ), the updated keys  $a'_i, b'_i, c'_i$  are also secure.

Finally, we consider the **procedure** NONLOCAL-CZ( $\{i, j\}, q, q'$ ), where  $\{i, j\} \in \mathcal{S}_q \cup \mathcal{S}_{q'}$ . Note that in this procedure,  $\mathcal{S}$  can only obtain two independent and uniform measurement outcomes  $s_i, s_j$ . According to the updating rules shown in Equation (22), we can see that as long as the encryption keys  $\{a_i, b_i, c_i\}$  and  $\{a_j, b_j, c_j\}$  are secure then the updated keys will also be secure against the servers. As a result, we conclude that, from the perspective of all servers, the output state of the computation is still encrypted by a sound quantum one-time pad.

### 6. Discussion

In this work, we proposed a secure distributed delegated quantum computation protocol, which allows clients to delegate their private computation to several quantum servers. We have shown that unconditional security of the input and output of the computation can be guaranteed as long as all servers follow the protocol honestly. Nevertheless, there are some notable problems in our work when we consider it in practice. In the end of this paper, we discuss those practical problems.

First, note that our protocol can only work well in a noise-free environment. To make our protocol fault-tolerant, we assume that each quantum server must be capable of performing *fault-tolerant quantum computation* [41]. However, this would inevitably increase the overhead of ancillary qubits. In addition, we need to consider two channel noises: one is between the client and each server, the other is between the third party and each server. The former will introduce errors in the input state, whereas the latter will introduce errors in the entangled state. There are some methods to remedy this problem. For the input state, the client can utilize some *quantum error-correct code* [42] to protect each qubit. However, it requires that the client can perform additional quantum operations. As for the entangled state, each pair of servers can use some *quantum entanglement distill* [43] protocol to obtain the entangled states with high fidelity. Similarly, it requires additional local operations and classical communications between the servers.

Second, note that our protocol can only protect the security of the input and output of the computation. This is because the model of the delegated quantum computation we used in our work is SDQC protocol instead of UBQC protocol. Nevertheless, we can convert, in principle, a SDQC protocol into a UBQC protocol. To do that, we first encode the delegated circuit  $U$  as a binary string denoted by  $C(U)$ . Next, according to the quantum computation theory [44], there exists a universal quantum circuit  $\mathcal{U}$  such that

$$\mathcal{U} |_{+C(U)} \rangle |_{+x} \rangle = |_{+C(U)} \rangle U |_{+x} \rangle, \quad (26)$$

where the input of the universal circuit  $\mathcal{U}$  consists of two parts:  $|_{+x} \rangle$  is the input state of  $U$  and  $|_{+C(U)} \rangle$  is the canonical and quantum description of the circuit  $U$ . Performing this universal circuit  $\mathcal{U}$  in our protocol, we can apparently achieve a blind distributed delegated quantum computation.

Last, we should note that in this work we only consider the honest servers and the third party who perform the protocol as the client desires. However, a real server may not follow the protocol honestly, and an untrusted third party may prepare some other entangled states for the servers. To detect such a malicious server including the untrusted third party, we should introduce a verification mechanics in our protocol. Indeed, verification is an important topic in the quantum computation theory (see [45,46]). There is an easy way to achieve the verification in our protocol. Specifically, given the delegated circuit  $U$ , the client can introduce another small quantum circuit  $V$ , for example, a permutation circuit [47], which is easy to simulate on a classical computer. The client then randomly inserts the qubits of  $V$  into the circuit  $U$  and runs this hybrid circuit on the universal quantum circuit  $\mathcal{U}$ . After the computation, the client check the result of  $V$ ; if the result does not match the desired, then the client rejects the output.

**Author Contributions:** Conceptualization, S.M. and C.Z.; formal analysis, S.M.; funding acquisition, C.Z. and D.Q.; methodology, S.M.; supervision, C.Z. and M.N.; validation, C.Z. and D.Q.; writing—original draft, S.M.; writing—review and editing, C.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (Grant Nos. 62001351, 61372076, 61971348); Natural Science Basic Research Program of Shaanxi, China (Grant No. 2021JM-142); Foundation of Shaanxi Key Laboratory of Information Communication Network and Security (ICNS201802); Key Research and Development Program of Shaanxi Province (2019ZDLGY09-02).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Appendix A. The Correctness of the H-Gadget

In this section, we briefly prove the correctness of the *H*-gadget proposed in [38]. We first translate the circuit of this gadget (see Figure 1) into an equivalent form. Note that the ancillary qubits  $|+\alpha_i\rangle = R_z(\alpha_i)|+\rangle$ ,  $|+\beta_i\rangle = R_z(\beta_i)|+\rangle$  and any *Z*-rotation operator is commutable with the controlled-*Z* gate, thus the circuit of the *H*-gadget can be expressed equivalently as follows:

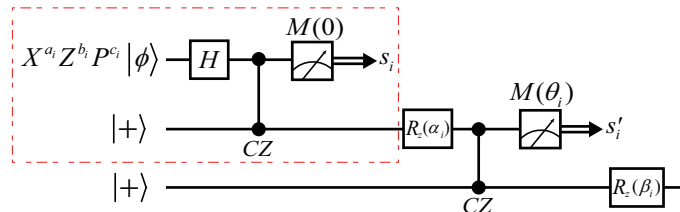


Figure A1. An equivalent circuit of the *H*-gadget of [38].

In Section 4, we obtained a variant *X*-teleportation (see Figure 5), which is identical to the above circuit in the red-dotted box. According to this, we can infer immediately that after performing the measurement  $M(0)$ , the rest circuit is equivalent to the following form, where the operator  $R_z(\alpha_i)$  has been absorbed into the input state.

We then use the identity shown in Figure A2, which is easy to verify. Applying this measurement identity to the circuit in Figure A3, we can obtain the following circuit (see Figure A4), where we exchange the positions of  $R_z(\theta_i)$  and CZ, and insert a pair of *H* gates between them. Obviously, the partial circuit in Figure A4 surrounded by the red-dotted box is the variant *X*-teleportation. Thus, we can infer that after the measurement the remaining qubit will be

$$R_z(\beta_i) X^{s'_i} H R_z(\alpha_i - \theta_i) X^{a_i \oplus s_i} Z^{b_i} P^{c_i} |\phi\rangle \tag{A1}$$

where  $R_z(\beta_i)$  is the *Z*-rotation operator in the end.

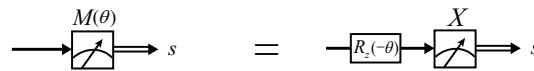


Figure A2. Measurement identity that converts  $M(\theta)$  basis to *X* basis.

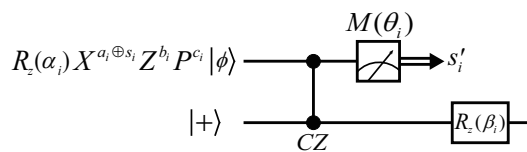


Figure A3. The rest circuit after performing the measurement  $M(0)$  on the top line.

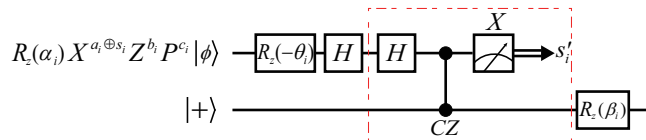


Figure A4. The variant *X*-teleportation where the input qubit is  $H R_z(\alpha_i - \theta_i) X^{a_i \oplus s_i} Z^{b_i} P^{c_i} |\phi\rangle$ .

In the following content, we simplify this output qubit. For simplicity, we temporarily drop the subscript  $i$  and define  $R_z(\gamma) \equiv Z^b P^c$ , that is,  $\gamma = (b + \frac{c}{2})\pi$ . It is easy to check that  $X^a R_z(\theta) X^a = R_z((-1)^a \theta)$  for any  $\theta$ . Thus, the output qubit can be rewritten as follows:

$$\begin{aligned} & R_z(\beta) X^{s'} H R_z(\alpha - \theta) X^{a \oplus s} R_z(\gamma) |\phi\rangle \\ &= X^{s'} R_z\left((-1)^{s'} \beta\right) H X^{a \oplus s} R_z\left((-1)^{a \oplus s} (\alpha - \theta)\right) R_z(\gamma) |\phi\rangle \\ &= X^{s'} R_z\left((-1)^{s'} \beta\right) Z^{a \oplus s} H R_z\left(\gamma + (-1)^{a \oplus s} (\alpha - \theta)\right) |\phi\rangle \\ &= X^{s'} R_z\left((-1)^{s'} \beta + (a \oplus s)\pi\right) H R_z\left(\gamma + (-1)^{a \oplus s} (\alpha - \theta)\right) |\phi\rangle \end{aligned} \tag{A2}$$

Let  $\theta = (-1)^{a \oplus s} \gamma + \alpha + h\pi$ , where  $h \in \{0, 1\}$ . Note that  $\theta$  here is seemingly not the same as the one defined in Equation (19). Despite that, we will show they are exactly the same one. Substitute  $\theta$  in the above equation, we can easily get the following result:

$$X^{s'} R_z\left((-1)^{s'} \beta + (a \oplus s)\pi\right) H R_z\left(-(-1)^{a \oplus s} h\pi\right) |\phi\rangle. \tag{A3}$$

As  $R_z$  is an operator with a period of  $2\pi$ , which means  $R_z(\pi) \equiv R_z(-\pi) \equiv Z$ , thus the output qubit can be expressed as follows:

$$\begin{aligned} & X^{s'} R_z\left((-1)^{s'} \beta + (a \oplus s)\pi\right) H Z^h |\phi\rangle \\ &= X^{s'} R_z\left((-1)^{s'} \beta + (a \oplus s)\pi\right) X^h H |\phi\rangle \\ &= X^{s' \oplus h} R_z\left((-1)^{s' \oplus h} \beta + (-1)^h (a \oplus s)\pi\right) H |\phi\rangle. \end{aligned} \tag{A4}$$

We further express the  $Z$ -rotation in Equation (A4) in terms of  $Z$  and  $P$ . Recalling that  $\beta = (f + \frac{g}{2})\pi$  (see Equation (18)) and considering the periodicity of  $Z$ -rotation operators, we can get that

$$\begin{aligned} & R_z\left((-1)^{s' \oplus h} (f + \frac{g}{2})\pi + (-1)^h (a \oplus s)\pi\right) \\ &\equiv R_z\left((a \oplus s \oplus f)\pi + (-1)^{s' \oplus h} \frac{g}{2}\pi\right) \\ &\equiv R_z\left((a \oplus s \oplus f)\pi + (-1)^{s' \oplus h} \frac{g}{2}\pi + 2(s' \oplus h)g\pi\right) \\ &= R_z\left((a \oplus s \oplus f \oplus [(s' \oplus h)g])\pi + \frac{(-1)^{s' \oplus h} + 2(s' \oplus h)}{2} g\pi\right). \end{aligned} \tag{A5}$$

Note that for any  $r \in \{0, 1\}$ ,  $(-1)^r + 2r = 1$ , so the above  $Z$ -rotation operator can be further rewritten as follows:

$$R_z\left((a \oplus s \oplus f \oplus [g(s' \oplus h)])\pi + \frac{g\pi}{2}\right) \equiv Z^{a \oplus s \oplus f \oplus [g(s' \oplus h)]} P g. \tag{A6}$$

Substituting the above equation to Equation (A4), we get the output qubit in the following form:

$$X^{s' \oplus h} Z^{a \oplus s \oplus f \oplus [g(s' \oplus h)]} P g H |\phi\rangle \tag{A7}$$

Finally, we substitute  $\gamma = (b + \frac{c}{2})\pi$  and  $\alpha = (d + \frac{e}{2})\pi$  in  $\theta = (-1)^{a \oplus s} \gamma + \alpha + h\pi$ , obtaining

$$\begin{aligned}
 \theta &= (-1)^{a \oplus s} (b\pi + \frac{c}{2}\pi) + (d\pi + \frac{e}{2}\pi) + h\pi \\
 &= b\pi + (-1)^{a \oplus s} \frac{c}{2}\pi + d\pi + \frac{e}{2}\pi + h\pi \\
 &= b\pi + c(a \oplus s)\pi + \frac{c}{2}\pi + d\pi + \frac{e}{2}\pi + h\pi \\
 &= h \oplus b \oplus d \oplus (ac) \oplus (sc)\pi + \frac{c+e}{2}\pi \\
 &= h \oplus b \oplus d \oplus (ac) \oplus (sc) \oplus (ce)\pi + \frac{c \oplus e}{2}\pi.
 \end{aligned} \tag{A8}$$

where in the last term we use another simple equality: for any  $c, e \in \{0, 1\}$ ,  $c + e = 2ce + c \oplus e$ . From the above results, the correctness of the  $H$ -gadget is obvious.

## References

- Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [[CrossRef](#)]
- Harrow, A.W.; Montanaro, A. Quantum computational supremacy. *Nature* **2017**, *549*, 203–209. [[CrossRef](#)]
- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [[CrossRef](#)]
- Campbell, E.T.; Fitzsimons, J. An introduction to one-way quantum computing in distributed architectures. *Int. J. Quantum Inf.* **2010**, *8*, 219–258. [[CrossRef](#)]
- Castelvecchi, D. IBM's quantum cloud computer goes commercial. *Nat. News* **2017**, *543*, 159. [[CrossRef](#)]
- Raussendorf, R.; Briegel, H.J. A One-Way Quantum Computer. *Phys. Rev. Lett.* **2001**, *86*, 5188–5191. [[CrossRef](#)] [[PubMed](#)]
- Briegel, H.J.; Browne, D.E.; Dür, W.; Raussendorf, R.; Van den Nest, M. Measurement-based quantum computation. *Nat. Phys.* **2009**, *5*, 19–26. [[CrossRef](#)]
- Morimae, T.; Fujii, K. Blind quantum computation protocol in which Alice only makes measurements. *Phys. Rev. A* **2013**, *87*, 050301. [[CrossRef](#)]
- Fisher, K.A.; Broadbent, A.; Shalm, L.; Yan, Z.; Lavoie, J.; Prevedel, R.; Jennewein, T.; Resch, K.J. Quantum computing on encrypted data. *Nat. Commun.* **2014**, *5*, 3074. [[CrossRef](#)] [[PubMed](#)]
- Broadbent, A.; Jeffery, S. Quantum homomorphic encryption for circuits of low T-gate complexity. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 609–629.
- Broadbent, A. Delegating private quantum computations. *Can. J. Phys.* **2015**, *93*, 410–413. [[CrossRef](#)]
- Liang, M. Teleportation-Based quantum homomorphic encryption scheme with quasi-compactness and perfect security. *Quantum Inf. Process.* **2020**, *19*, 28. [[CrossRef](#)]
- Jiang, L.; Taylor, J.M.; Sørensen, A.S.; Lukin, M.D. Distributed quantum computation based on small quantum registers. *Phys. Rev. A* **2007**, *76*, 062323. [[CrossRef](#)]
- Moehring, D.L.; Madsen, M.J.; Younge, K.C.; Kohn, R.N., Jr.; Maunz, P.; Duan, L.M.; Monroe, C.; Blinov, B.B. Quantum networking with photons and trapped atoms (Invited). *J. Opt. Soc. Am. B* **2007**, *24*, 300–315. [[CrossRef](#)]
- Li, Y.; Benjamin, S.C. High threshold distributed quantum computing with three-qubit nodes. *New J. Phys.* **2012**, *14*, 093008. [[CrossRef](#)]
- Nickerson, N.H.; Li, Y.; Benjamin, S.C. Topological quantum computing with a very noisy network and local error rates approaching one percent. *Nat. Commun.* **2013**, *4*, 1756. [[CrossRef](#)]
- Monroe, C.; Raussendorf, R.; Ruthven, A.; Brown, K.R.; Maunz, P.; Duan, L.M.; Kim, J. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Phys. Rev. A* **2014**, *89*, 022317. [[CrossRef](#)]
- Cacciapuoti, A.S.; Caleffi, M.; Tafuri, F.; Cataliotti, F.S.; Gherardini, S.; Bianchi, G. Quantum internet: Networking challenges in distributed quantum computing. *IEEE Netw.* **2019**, *34*, 137–143. [[CrossRef](#)]
- Liu, J.X.; Ye, J.Y.; Yan, L.L.; Su, S.L.; Feng, M. Distributed quantum information processing via single atom driving. *J. Phys. B At. Mol. Opt. Phys.* **2020**, *53*, 035503. [[CrossRef](#)]
- Zhong, Y.; Chang, H.S.; Bienfait, A.; Dumur, É.; Chou, M.H.; Conner, C.R.; Grebel, J.; Povey, R.G.; Yan, H.; Schuster, D.I.; et al. Deterministic multi-qubit entanglement in a quantum network. *Nature* **2021**, *590*, 571–575. [[CrossRef](#)]
- Daiss, S.; Langenfeld, S.; Welte, S.; Distant, E.; Thomas, P.; Hartung, L.; Morin, O.; Rempe, G. A quantum-logic gate between distant quantum-network modules. *Science* **2021**, *371*, 614–617. [[CrossRef](#)]
- Childs, A.M. Secure assisted quantum computation. *Quantum Inf. Comput.* **2005**, *5*, 456–466. [[CrossRef](#)]
- Rohde, P.P.; Fitzsimons, J.F.; Gilchrist, A. Quantum Walks with Encrypted Data. *Phys. Rev. Lett.* **2012**, *109*, 150501. [[CrossRef](#)]
- Dunjko, V.; Fitzsimons, J.F.; Portmann, C.; Renner, R. Composable security of delegated quantum computation. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taiwan, China, 7–11 December 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 406–425.

25. Marshall, K.; Jacobsen, C.S.; Schäfermeier, C.; Gehring, T.; Weedbrook, C.; Andersen, U.L. Continuous-Variable quantum computing on encrypted data. *Nat. Commun.* **2016**, *7*, 13795. [[CrossRef](#)]
26. Zhou, Q.; Lu, S.; Cui, Y.; Li, L.; Sun, J. Quantum search on encrypted data based on quantum homomorphic encryption. *Sci. Rep.* **2020**, *10*, 5135. [[CrossRef](#)]
27. Wang, D.; Liu, Y.; Ding, J.; Qiang, X.; Liu, Y.; Huang, A.; Fu, X.; Xu, P.; Deng, M.; Yang, X.; et al. Remote-controlled quantum computing by quantum entanglement. *Opt. Lett.* **2020**, *45*, 6298–6301. [[CrossRef](#)]
28. Zhao, X.; Zhao, B.; Wang, Z.; Song, Z.; Wang, X. Practical distributed quantum information processing with LOCCNet. *Npj Quantum Inf.* **2021**, *7*, 159. [[CrossRef](#)]
29. Sherson, J.F.; Krauter, H.; Olsson, R.K.; Julsgaard, B.; Hammerer, K.; Cirac, I.; Polzik, E.S. Quantum teleportation between light and matter. *Nature* **2006**, *443*, 557–560. [[CrossRef](#)]
30. Olmschenk, S.; Matsukevich, D.N.; Maunz, P.; Hayes, D.; Duan, L.M.; Monroe, C. Quantum Teleportation Between Distant Matter Qubits. *Science* **2009**, *323*, 486–489. [[CrossRef](#)]
31. Chou, C.W.; De Riedmatten, H.; Felinto, D.; Polyakov, S.V.; Van Enk, S.J.; Kimble, H.J. Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature* **2005**, *438*, 828–832. [[CrossRef](#)]
32. Peng, T.; Harrow, A.W.; Ozols, M.; Wu, X. Simulating large quantum circuits on a small quantum computer. *Phys. Rev. Lett.* **2020**, *125*, 150504. [[CrossRef](#)]
33. Bravyi, S.; Smith, G.; Smolin, J.A. Trading classical and quantum computational resources. *Phys. Rev. X* **2016**, *6*, 021043. [[CrossRef](#)]
34. Broadbent, A.; Fitzsimons, J.; Kashefi, E. Universal blind quantum computation. In Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, Atlanta, GA, USA, 25–27 October 2009; pp. 517–526.
35. Aharonov, D.; Ben-Or, M.; Eban, E.; Mahadev, U. Interactive proofs for quantum computations. *arXiv* **2017**, arXiv:1704.04487.
36. Zeuner, J.; Pitsios, I.; Tan, S.H.; Sharma, A.N.; Fitzsimons, J.F.; Osellame, R.; Walther, P. Experimental quantum homomorphic encryption. *Npj Quantum Inf.* **2021**, *7*, 25. [[CrossRef](#)]
37. Barz, S.; Kashefi, E.; Broadbent, A.; Fitzsimons, J.F.; Zeilinger, A.; Walther, P. Demonstration of blind quantum computing. *Science* **2012**, *335*, 303–308. [[CrossRef](#)]
38. Ma, S.; Zhu, C.; Nie, M.; Quan, D.; Pei, C. Secure delegated quantum computation based on Z-rotation encryption. *Europhys. Lett.* **2022**, *137*, 38001. [[CrossRef](#)]
39. Eisert, J.; Jacobs, K.; Papadopoulos, P.; Plenio, M.B. Optimal local implementation of nonlocal quantum gates. *Phys. Rev. A* **2000**, *62*, 052317. [[CrossRef](#)]
40. Zhou, X.; Leung, D.W.; Chuang, I.L. Methodology for quantum logic gate construction. *Phys. Rev. A* **2000**, *62*, 052316. [[CrossRef](#)]
41. Gottesman, D. Theory of fault-tolerant quantum computation. *Phys. Rev. A* **1998**, *57*, 127. [[CrossRef](#)]
42. Calderbank, A.R.; Shor, P.W. Good quantum error-correcting codes exist. *Phys. Rev. A* **1996**, *54*, 1098. [[CrossRef](#)]
43. Rozpedek, F.; Schiet, T.; Thinh, L.P.; Elkouss, D.; Doherty, A.C.; Wehner, S. Optimizing practical entanglement distillation. *Phys. Rev. A* **2018**, *97*, 062333. [[CrossRef](#)]
44. Bernstein, E.; Vazirani, U. Quantum complexity theory. *SIAM J. Comput.* **1997**, *26*, 1411–1473. [[CrossRef](#)]
45. Fitzsimons, J.F. Private quantum computation: An introduction to blind quantum computing and related protocols. *Npj Quantum Inf.* **2017**, *3*, 23. [[CrossRef](#)]
46. Gheorghiu, A.; Kapourniotis, T.; Kashefi, E. Verification of quantum computation: An overview of existing approaches. *Theory Comput. Syst.* **2019**, *63*, 715–808. [[CrossRef](#)]
47. Ma, S.; Zhu, C.; Nie, M.; Quan, D. Efficient self-testing system for quantum computations based on permutations. *Chin. Phys. B* **2021**, *30*, 040305. [[CrossRef](#)]