Doctoral Dissertation

Extension of Blind Quantum Computation

Yuichi Sano

Department of Nuclear Engineering
Kyoto University

Abstract

Quantum computers, which can perform calculations that classical computers cannot, are in great demand by many people nowadays. Quantum computers are too expensive for them to own, and therefore, users may want to delegate their calculations to quantum cloud servers offered by developers. However, the quantum server is not necessarily honest. Thus, the security of the delegation protocol is an important issue. If the users can encrypt the input/output and the computation algorithm, the quantum server cannot know anything about the user's calculations. An art called blind quantum computation protocol has been developed to achieve this purpose. In this thesis, we aim to improve the blind quantum computation protocols so that users with lower computational power can enjoy higher security.

We first try to extend single-server blind quantum computation protocols. In single-server protocols, a user interacts with a single quantum server. The single-server blind quantum computation protocols proposed thus far are based on a measurement-based quantum computation. In contrast, our new protocol employs a circuit-based quantum computation. We construct the protocol by combining gate teleportation and quantum one-time pad. Since the protocol is circuit-based, various discussions based on the quantum circuit model, such as error correction codes, can be directly applied. We also show the potential to reduce the number of qubits used compared to existing protocols.

We then focus on extending the multi-server blind quantum computation protocols. In those protocols, a user delegates computations to multiple servers. These protocols prevent server fraud by allowing multiple quantum servers to monitor each other and, as a result, require users to have only classical capabilities. In the known protocols, however, classical communication between servers is forbidden. We relax this restriction and develop a blind quantum computation protocol that is secure even if some servers communicate classically after the computation. Our results allow a quantitative assessment of the useful period of the blind quantum computation protocol under certain assumptions.

Finally, we examine how far the restrictions of multi-server blind quantum computation protocols can be relaxed. The restrictions imposed on quantum servers in multi-server blind quantum computation protocols are quite strong and are desirable to be weakened as much as possible. We show that it is difficult to go beyond our results above by comparing the limitation with that of the user's restriction about single-server blind quantum computation protocols. The results show a link between single-server and multi-server blind quantum calculations.

List of Papers

This thesis is based on the following papers:

1. (Reproduced from [1], with permission from Springer Nature)

   Yuichi Sano, “Equivalence of single-server and multiple-server blind quantum computation protocols”. *Quantum Information Processing* **22**, 61, 2023.

2. (Reproduced from [2], with permission from Springer Nature)

   Yuichi Sano, “Multi-server blind quantum computation protocol with limited classical communication among servers”. *Quantum Information Processing* **21**, 88, 2022.

3. ([3])

   Yuichi Sano, “Blind Quantum Computation Using a Circuit-Based Quantum Computer”. *Journal of the Physical Society of Japan* **90**, 12, 2021.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

As quantum theory differs significantly from classical physics, the various applications in quantum computation and quantum cryptography, where it is applied, also differ significantly from their classical counterparts. However, while such differences are well-known today as many meaningful applications have been found, they were not necessarily obvious in the 1920s and 1930s when quantum theory was formulated [4–6]. It was in the 1980s that quantum theory was recognized to have the potential to apply computer science and cryptography. In 1982, Feynman pointed out the usefulness of quantum computers by claiming that a computer based on quantum theory is needed to simulate quantum systems [7]. In 1983, Wiesner proposed quantum money as an effective use of quantum states in the field of cryptography [8], and the basic idea of this quantum money led to Bennett and Brassard's proposal for quantum key distribution (QKD) in 1984 [9]. Following these groundbreaking ideas, various useful quantum algorithms and quantum cryptographic protocols were proposed.

Quantum algorithms were shown to be capable of solving various problems efficiently. After Feynman's proposal, Deutsch and Jozsa found the first important algorithm [10]. In 1985, Deutsch published an idea of a quantum Turing machine and the algorithmic basis of Deutsch's algorithm [11]. The advent of the Deutsch–Jozsa algorithm, a generalization of Deutsch's algorithm, revealed that there are problems that can be efficiently solved by a quantum computer but not by a deterministic classical computer. Then, in 1993, Bernstein and Vazirani clearly formalized the notion of a quantum Turing machine. In addition, they proposed an algorithm, called the Bernstein–Vazirani algorithm, which enables us to separate the class of problems that a stochastic classical computer can solve [12]. Then, in 1994, Shor published prime factorization algorithm [13, 14], one of the monumental works of quantum algorithms inspired by the quantum Fourier transform [12] and Simon's algorithm [15]. Shor's algorithm made a huge public impact as it can crack the RSA cryptosystem, a widely used public-key cryptosystem, if it will be physically implemented [16]. Then in 1996, another algorithm that made quantum computers famous, Grover's algorithm, was developed [17]. Grover's algorithm is a search algorithm that finds the required data from an unstructured database. Although it is a quadratic speedup, it achieves an optimum computational complexity that is strictly unattainable on a classical computer [18]. Furthermore, since 1997, a series of quantum simulation algorithms, as predicted by Feynman, have been proposed to simulate quantum systems [19–22]. In the 2000s, the quantum walk-search algorithm [23, 24], a generalization of Grover's algorithm, and the Harrow-Hassidim-Lloyd (HHL) algorithm [25], an efficient inverse matrix computation algorithm, and their applications were proposed. In recent years, quantum singular value transformation (QSVT) was found. This theory shows that major quantum algorithms such as Shor's algorithm, Grover's algorithm, quantum simulation, and the HHL algorithm can be understood in a unified way using singular value transformations of matrices [26, 27].

Another major pillar of applications is quantum cryptography. At first glance, quantum states and cryptographic protocols seem to have nothing to do with each other. However, quantum states have interesting properties regarding their information, such as the no-cloning theorem [28–30], the preparation uncertainty relation [31,32], and the information disturbance theorem [33–35]. These properties prohibit the complete extraction of information from a single quantum state, and therefore using quantum states to exchange information is considered resistant to attack from adversaries. As mentioned above, the first quantum cryptographic protocol is quantum money [8], which uses the no-cloning theorem to create money tokens in a quantum state. The next protocol proposed was the BB84 protocol [9] for QKD, one of the most successful applications of quantum theory. The E91 protocol using EPR pairs was then proposed by Ekert [36]. The protocol was shown to be equivalent to the BB84 protocol by transforming it in a proper way. Therefore, the E91 protocol which is easy to analyze offers security as strong as the BB84. A device-independent QKD (DIQKD) was proposed based on these QKD, which has the remarkable property of guaranteeing safety no matter how the devices are implemented [37–40]. Classical cryptography techniques such as leftover hash lemma have also been applied in quantum cryptography [41]. However, classical cryptography itself has not been used to preserve the information-theoretic security of quantum cryptography. In recent years, combining classical cryptography and quantum states has led to the development of cryptographic protocols that can not be performed using classical cryptography alone under computational security [42–49]. Therefore, quantum information is becoming increasingly important not only in quantum key distribution but also in the field of cryptography.

Finally, the history of the development of blind quantum computation is reviewed. As mentioned above, quantum computation and cryptography are applications based on quantum theory but have developed separately. Bling quantum computation is a sort of their combination. As classical computers became widespread, the importance of delegating calculations was highlighted. Users with no or little computing power delegate their calculations to a server with high computing power, and the server performs the calculations on their behalf. This scheme is the same concept as what we now call cloud computing services. The problem with this scheme is the confidentiality of the user's information [50–52]. The earliest user security idea in quantum computers was parallel to classical computers [53]. That is, it focuses on how the user can hide the input against the quantum server.

In 2005, Childs proposed blind quantum computation protocols with stronger security than classical homomorphic cryptography [54]. A *blind quantum computation protocol* is a quantum delegated computation in which the quantum server can learn neither only the user's input data nor the calculation algorithm, except for $L_{\mathrm{P}}(X)$, the size of the calculation. We define a blind quantum computation protocol in the following:

Definition 1.1 (Blind Quantum Computation Protocol)**.** Consider a delegation protocol P that can perform quantum computations executed by any polynomial-size quantum circuit. We denote the quantum circuit, an input to the protocol, by $X \in \{0,1\}^*$. Let $L_{\mathrm{P}}(X)$ be defined as the size of the quantum circuit executed by the server when the input to the protocol P is X. We call the delegation protocol P as a blind quantum computation protocol if it satisfies the following conditions:

1. For any X_1 and X_2 such that $L_{\mathrm{P}}(X_1) = L_{\mathrm{P}}(X_2)$, the probability distributions of the classical information obtained by the server during the protocol P for each input are identical.

2. For inputs such that the probability distributions of the classical information obtained by the server during the execution of protocol P are identical, the quantum states obtained by the server are indistinguishable.

Childs proposed protocol is for a user with a *weak* quantum computer to perform universal quantum computation with *blindness*. His ideas to hide the input ....from the quantum server are similar to quantum money and QKD. Note that Childs protocol requires changing the encryption key at every quantum gate to hide the algorithm, thus needs a large amount of quantum memory.

The next protocols proposed were the Aharonov, Ben-Or, and Eban protocols based on quantum authentication schemes (QAS) [55]. Their protocol has reduced the quantum memory requirement of Childs protocol to three qubits. Nevertheless, users still need quantum memory to maintain multiple qubits for encryption.

Then Broadbent, Fitzsimons, and Kashfi proposed a blind quantum computation protocol based on a measurement-based quantum computation (BFK protocol) [56]. The measurement-based quantum computation is completely parallel to circuit-based quantum computation, with resources as quantum computation in the measurement part [57,58]. The BFK protocol does not require quantum memory from the user. That is, the user can run the blind quantum computation protocol as long as they have the ability to create a single qubit and execute any single qubit gate and the quantum communication capability to send the qubit to the server. A protocol was subsequently proposed by Morimae and Fujii (MF protocol), which replaced state preparation with measurement [59]. Moreover, surprisingly, it has been shown that user capacity is likely to be unable to be further reduced [60,61]. However, no known blind quantum computation protocols using circuit-based quantum computation could be performed with the same user capabilities as the BFK protocol.

Another important blind quantum computation protocol is the blind quantum computation protocol with multiple quantum servers [62, 63]. A multi-server blind quantum computation (MBQC) protocol is one in which a classical user interacts with multiple quantum servers to delegate a computation. MBQC protocols do not require any quantum ability from the user, unlike the single-server blind quantum computation (SBQC) protocols. Therefore, MBQC protocols are more accessible to most users who can perform only classical computations. However, for those MBQC protocols to be secure, multiple quantum servers must share entangled quantum states. In addition, they are prohibited from communicating with each other at all. This restriction on the server of the MBQC protocols is qualitatively different from the capabilities required of users of the SBQC protocols. In fact, while the users can easily check their own capabilities, they cannot confirm if the servers honestly follow the no-communication rule. Whereas the limits of the user capabilities of the SBQC protocols are known, the limits of the server constraints of the MBQC protocols have not been known yet.

1.2 Our Work

Our work has solved the open problems described in the previous section:

1. we have developed a circuit-based blind quantum computation protocol that users can run without quantum memory,
2. we have developed the MBQC protocol with relaxed restrictions imposed on quantum servers,
3. we have proved the limits of the MBQC protocol constraints and their relation to SBQC protocol.

1.2.1 Extension of Single-server Blind Quantum Computation

We have extended a circuit-based blind quantum computation protocol for the protocol that users can use without quantum memory [3]. We prove the following theorem in Chapter 3:

Theorem 1.2 (Informal)**.** Protocol 1 (in p. 60) is a blind quantum computation protocol in which a user without any quantum memory can delegate computations to a single server.

Whereas users do not need quantum memory in measurement-based blind quantum computation protocols such as the BFK and MF protocols, all circuit-based blind quantum computation protocols proposed thus far require users to have quantum memory. Childs protocol requires quantum memory for users, as it can not be implemented without informing the server of T gates, which is essential for universal quantum computation. We have proposed a method whereby gate teleportation and additional quantum states could be used as resources to execute T gates without informing the server. Furthermore, using *circuit like brickwork states*, a method of constructing quantum circuits inspired by the BFK protocol, information on the structure of quantum circuits can be hidden from the quantum server. Using these techniques, we developed a circuit-based blind quantum computation protocol that users can use without quantum memory.

1.2.2 Extension of Multiple-server Blind Quantum Computation

We have extended the MBQC protocol, making it safe for some quantum servers to communicate after the computation is finished [2]. We prove the following theorem in Chapter 4:

Theorem 1.3 (Informal)**.** Protocol 2 (in p. 73) is a multiple-server blind quantum computation protocol in which some servers are allowed to communicate after computation.

The MBQC protocols in previous studies prohibit any communication between servers. Restrictions on this server continue to be imposed after the delegated calculations have been completed. As this very strong restriction is difficult to impose, the protocol may leak some information in its practical use. We have proposed the MBQC protocol that relaxes this restriction. First, we extend the original two-server MBQC protocol to N-server one. To further eliminate information about the structure of the quantum circuit, we encrypted the quantum circuit using a *circuit like brickwork states.* Next, we developed a new quantum gate encryption method called *dummy gate*, which allows information to be leaked from any $N-1$ server out of N servers. Finally, we have extended the X gate encryption method for hiding output for N servers. These techniques ensure that our protocols remain blind even when $N-1$ servers in N servers perform classical/quantum communication after computation. Note, however, that any server-to-server communication is still prohibited during the calculation.

1.2.3 Limitation of Blind Quantum Computation

We have proved the limits on the communication restrictions imposed on the servers of MBQC protocols [1]. We prove the following theorem in Chapter 5:

Theorem 1.4 (Informal)**.** The MBQC protocols that allow communication between servers after computation are equivalent to the SBQC protocols that classical users can execute.

As mentioned above, it is known that there is likely to be no SBQC protocol that is executable by classical users [60, 61]. This means that there may not even be an MBQC protocol with which any servers can communicate classically with each other after computation/during computation, so this theorem represents a limitation of MBQC protocols.

1.3 Structure of this thesis

This thesis is organized as follows: In Chapter 2, we review the fundamentals of quantum theory and quantum computation, furthermore introduce some blind quantum computation protocols. In Chapter 3, we propose our new circuit-based blind quantum computation protocol based on [3]. In Chapter 4, we propose a more accessible MBQC protocol based on [2], which would relax the restrictions imposed on quantum servers. In Chapter 5, we analyze the limitations of the MBQC protocol limits based on the discussion in [1]. In Chapter 6, we summarize our results and discuss future works.

Chapter 2

Preliminaries

This chapter provides the preliminaries needed for later chapters. First, we review the axioms of the quantum theory underlying quantum computation and discuss some of the properties that can be derived from them. Then, we introduce the fundamentals of quantum computation, central to our research, and conclude with an overview of blind quantum computation.

2.1 Quantum Theory

We introduce the fundamental postulates that constitute the quantum theory, providing the basis for our understanding of quantum computing. We consider only finite-dimensional quantum systems in this thesis.

2.1.1 State

First, let us begin with a quantum state.

Postulate 2.1 (State)**.** An isolated quantum system has a Hilbert space $\mathcal{H}$, and its quantum state is represented as a unit vector $|\psi\rangle$ of the Hilbert space.

The simplest but the most important physical system is a qubit system. In quantum computation, the basis for qubit systems is the following computational basis:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{2.1}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.2}$$

Any single qubit state can be expressed as follows:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \tag{2.3}$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. This condition is equivalent to the state being a unit vector. This is because being a unit vector means that its norm (induced by inner product) equals one.

We call α and β amplitudes/probability amplitudes. We discuss why we call them *probability amplitudes* after describing quantum measurements.

As seen from Equation (2.3), quantum states can become *superposition* states. In other words, classical bits/states have only deterministic values, whereas qubits/states do not necessarily have deterministic values. This property is important in quantum computation and quantum cryptography.

2.1.2 Evolution

We then consider how the state changes.

Postulate 2.2 (Evolution)**.** The evolution of a closed quantum system is described by unitary transformations of states. That is, the relationship between the state $|\psi\rangle$ at time t_1 and the state $|\psi'\rangle$ at time t_2 of the same system is written by a unitary operator U which depends only on times t_1 and t_2 as follows:

$$|\psi'\rangle = U(t_1, t_2) |\psi\rangle. \tag{2.4}$$

A unitary operator describing the evolution of a quantum state is a complex square matrix U satisfying $UU^\dagger = U^\dagger U = I$. In other words, a unitary operator is necessary for a closed quantum system to remain in a quantum state after its evolution, i.e., it is a unit vector. This is obvious as unitary operators map a basis to another.

We present the Pauli and Hadamard gates as examples of the specific time evolution of the qubit system. The Pauli gates/matrices are defined by the following matrices:

$$
\begin{aligned}
I &\equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, && (2.5)\\
X &\equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, && (2.6)\\
Y &\equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, && (2.7)\\
Z &\equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. && (2.8)\\
& && (2.9)
\end{aligned}
$$

The Pauli gates are orthogonal in the Hilbert-Schmidt inner product and form a basis for the 2×2 complex matrices. Let us examine the time evolution induced by each operator. Firstly, identity operator I represents the constant operation, i.e., the state does not change:

$$
\begin{aligned}
I\left|\psi\right\rangle =&I(\alpha\left|0\right\rangle + \beta\left|1\right\rangle)\\
=&\alpha\left|0\right\rangle + \beta\left|1\right\rangle\\
=&\left|\psi\right\rangle.
\end{aligned}
\tag{2.10}
$$

The Pauli X gate can be considered an operation corresponding to a classical bit flip that swaps $\left|0\right\rangle$ and $\left|1\right\rangle$:

$$
\begin{aligned}
X\left|\psi\right\rangle =&X(\alpha\left|0\right\rangle + \beta\left|1\right\rangle)\\
=&\alpha\left|1\right\rangle + \beta\left|0\right\rangle.
\end{aligned}
\tag{2.11}
$$

The Pauli Z gate is an operation called a phase flip, which has no counterpart in classical bit manipulation:

$$
\begin{aligned}
Z\left|\psi\right\rangle =&Z(\alpha\left|0\right\rangle + \beta\left|1\right\rangle)\\
=&\alpha\left|0\right\rangle - \beta\left|1\right\rangle.
\end{aligned}
\tag{2.12}
$$

The Pauli Y gate can then be considered a continuous execution of the Z and X gates, except for the global phase, from $Y = iXZ$.

The Hadamard gate H is a unitary operator represented by the following matrix:

$$
H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.13}
$$

The Hadamard gate can be understood as one that creates a superposition state. Executing on the $|0\rangle$ state of the Hadamard gate leads to the $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state, which is an invariant state of the Pauli X gate. Similarly, executing on the $|1\rangle$ state of the Hadamard gate leads to the $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ state, which is the intrinsic state of the Pauli X gate.

2.1.3 Measurement

In this subsection, we consider measurements to obtain information on quantum systems.

Postulate 2.3 (Quantum measurement)**.** Quantum measurement is described by a set of measurement operators $\{M_n\}$, which satisfies the following *completeness equation*:

$$\sum_n M_n^\dagger M_n = I. \tag{2.14}$$

The index n of the measurement operator M_n represents a measurement output. When a measurement $\{M_n\}$ is performed on a quantum state $|\psi\rangle$, the probability that the measurement output is n is given by

$$p(n) = \langle\psi| M_n^\dagger M_n |\psi\rangle . \tag{2.15}$$

The state after the measurement given that the outcome is n is

$$\frac{M_n |\psi\rangle}{\sqrt{\langle\psi| M_n^\dagger M_n |\psi\rangle}}. \tag{2.16}$$

In classical physics, the output of a measurement is determined before the measurement is made, whereas in quantum theory, the measurement changes the state, and the output is stochastic. The measurement operator satisfies the completeness equation consistent with that measurement is a stochastic act of obtaining the output, as follows:

$$\begin{aligned}\sum_n p(n) =& \sum_n \langle\psi| M_n^\dagger M_n |\psi\rangle \\ =& \langle\psi| \sum_n M_n^\dagger M_n |\psi\rangle \\ =& \langle\psi| I |\psi\rangle \\ =&1.\end{aligned} \tag{2.17}$$

One example of a measurement in a qubit system is a measurement using the computational basis. The measurement operators for measurement by computational basis are the following two operators:

$$M_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.18}$$

$$M_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.19}$$

It is easy to see that these operators are Hermitian and satisfy the completeness equation. We consider the case where state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is measured on the computational basis. From Postulate 2.3, the probability of obtaining output 0 is

$$\begin{aligned} p(0) =& \langle\psi| M_0^\dagger M_0 |\psi\rangle \\ =& (\overline{\alpha}\langle 0| + \overline{\beta}\langle 1|)|0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle) \\ =& |\alpha|^2. \end{aligned} \tag{2.20}$$

Similarly, the probability of obtaining output 1 is $p(1) = |\beta|^2$. Therefore, the probability amplitude squared is the probability. This is why we call it the *probability amplitude.* In addition, the states after each measurement are

$$\frac{M_0|\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}|0\rangle, \tag{2.21}$$

$$\frac{M_1|\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|}|1\rangle. \tag{2.22}$$

These states are equal to $|0\rangle$ and $|1\rangle$ except for the global phase. Thus, it can be seen that the quantum states are *projected* by M_0 and M_1 to $|0\rangle$ and $|1\rangle$, respectively.

Projective Measurement

We generalize measurements that project to a computational basis to *projective measurements.*

Definition 2.4 (Projective Measurement)**.** A projective measure is described by an observable M, which is Hermitian on the target state space. The observable has a spectral decomposition

$$M = \sum_m m P_m, \tag{2.23}$$

where P_m is the projector on to the eigenspace of M with eigenvalue m. The set of measurement operators is $\{P_m\}$.

A projective measurement is clearly part of quantum measurement. We show later that projective measurement is equivalent to quantum measurement.

The measurement probabilities and the state after measurement by the projector P_m are as follows:

$$p(m) = \langle\psi| P_m |\psi\rangle , \tag{2.24}$$

and

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}}. \tag{2.25}$$

These equations are equal to Equations (2.15) and (2.16), using the projector's properties $P_m^\dagger = P_m$ and $P_m^2 = P_m$.

A measurement using the aforementioned computational basis $\{M_0, M_1\}$ is a projective measurement. Because the measurement operator M_0 is also a projector as follows:

$$M_0^\dagger = (|0\rangle \langle 0|)^\dagger = |0\rangle \langle 0| = M_0, \tag{2.26}$$

$$M_0^2 = (|0\rangle \langle 0|)^2 = |0\rangle \langle 0| = M_0. \tag{2.27}$$

Similarly, the measurement operator M_1 is a projector.

POVM Measurement

A projective measurement gives us an intuitive understanding of measurement, but we can also consider more general measurements. For example, what measurement should be considered when distinguishing between states $|0\rangle$ and $|+\rangle$? It can be seen that these states cannot be completely distinguished by any binary projection measurement. Suppose that a binary projection measurement $\{P_0, P_+\}$ can distinguish between two states. The assumption can be rewritten as follows:

$$\langle 0| P_0 |0\rangle = 1, \tag{2.28}$$

$$\langle +| P_+ |+\rangle = 1. \tag{2.29}$$

However, no projective measurement satisfies these equations and a completeness equation simultaneously.

How about considering a three-output measurement that has a third output called *indistinguishable* as an alternative? In other words, the measurement is such that the state is always $|0\rangle$ when the output is 0, the state is always $|+\rangle$ when the output is +, and the state can be either when the output is *indistinguishable*. The measurement operators corresponding to these measurement outputs are as follows:

$$\begin{aligned} M_0 \equiv& \lambda \left|-\right\rangle \left\langle-\right|, & (2.30)\\ M_+ \equiv& \lambda \left|1\right\rangle \left\langle 1\right|, & (2.31)\\ M_{indistinguishable} \equiv& \sqrt{I - M_0^\dagger M_0 - M_+^\dagger M_+}, & (2.32)\end{aligned}$$

where λ is a complex number satisfying $0 \leq |\lambda|^2 \leq \frac{\sqrt{2}}{1-\sqrt{2}}$. The probability of obtaining output 0 on the state $|+\rangle$ is

$$\begin{aligned} p(0) =& \left\langle+\right| M_0^\dagger M_0 \left|+\right\rangle \\ =& \frac{\sqrt{2}}{1-\sqrt{2}} \left\langle+\right|\left|-\right\rangle \left\langle-\right|\left|+\right\rangle \\ =& 0. \end{aligned} \tag{2.33}$$

Similarly, the probability of obtaining output + on the state $|0\rangle$ is zero. This measurement is not a projective measurement. Thus, quantum measurements include measurements that are not projective measurements. Such general measurements are called *POVM measurements*, and the operators $E_m \equiv M_m^\dagger M_m$ corresponding to the measurement outputs are called *POVM elements*. From the completeness equation, it follows that

$$\sum_m E_m = \sum_m M_m^\dagger M_m = I. \tag{2.34}$$

A POVM measurement can be said to focus only on measurement probability. In other words, when considering an operator that returns a non-negative value less than or equal to 1 for any state, such an operator is a POVM operator. The set of POVM operators $\{E_m\}$ such that the sum of probabilities is one is called a POVM.

2.1.4 Composite System

Until now, quantum systems consisting of a single system have been considered. How would the physics of a composite system consisting of several systems be described?

Postulate 2.5 (Composite system)**.** The state space of a composite system is described by the tensor product of the individual state spaces. In a composite system consisting of n quantum systems, when the quantum state of each state space is $|\psi_i\rangle$, the quantum state of the composite system is described by $|\psi_1\rangle \otimes |\psi_2\rangle \cdots \otimes |\psi_n\rangle$. Such a state is called a *separable state*. In addition, there are quantum states where multiple states are *entangled*, as follows:

$$|\Psi\rangle_{1,2,\ldots,n} = \sum_{i,j,k} q_{i,j,k} \, |i\rangle_1 \, |j\rangle_2 \ldots |k\rangle_n \, , \tag{2.35}$$

where $q_{i,j,k}$ denotes a probability amplitude, $|i\rangle_1$ denotes a state of system 1, $|j\rangle_2$ denotes a state of system 2, and $|k\rangle_n$ denotes a state of system n. Such a state is called a *entangled state*.

In Postulate 2.5, we write the state of the composite system as $|\psi_1\rangle \otimes |\psi_2\rangle \cdots \otimes |\psi_n\rangle$, but as a more compact notation we also use $|\psi_1\psi_2 \cdots \psi_n\rangle$.

The Bell states are an important example of composite qubit systems. The Bell states are the four maximum entanglement states on the two qubits system $\mathbb{C}^2 \times \mathbb{C}^2$ as follows:

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{2.36}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{2.37}$$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{2.38}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{2.39}$$

These four Bell states form a complete orthonormal system of a two-qubit system. In addition, the measurement using the Bell basis $\{|\Psi^+\rangle \langle\Psi^+| \, , |\Psi^-\rangle \langle\Psi^-| \, , |\Phi^+\rangle \langle\Phi^+| \, , |\Phi^-\rangle \langle\Phi^-|\}$ are called the Bell measurement.

One of the most significant results in quantum theory is that Bell states are quantum states that violate Bell's inequality [64–67]. The violation of Bell's inequality by Bell states indicates that local realism cannot describe quantum theory. In contrast, classical physics is based on local realism, highlighting yet another difference between quantum theory and classical physics.

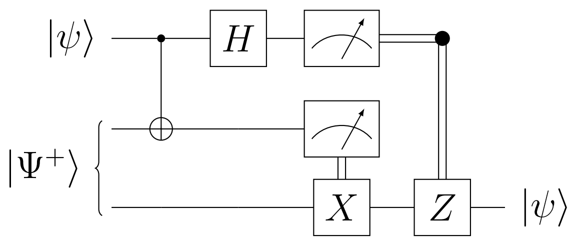


Figure 2.1: Quantum circuit of quantum teleportation

An important use of the Bell states is *quantum teleportation* [68–70]. Quantum teleportation is a protocol that allows Alice and Bob, who share the EPR pair $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, one of the Bell states, to communicate quantum states using only classical communication. Quantum teleportation plays an essential role in blind quantum computation, quantum cryptography, and quantum repeater/quantum communication [71, 72]. The quantum circuit representing quantum teleportation is shown in Figure 2.1.

Definition 2.6 (Quantum teleportation)**.** Alice has a quantum state $|\psi\rangle \in \mathbb{C}^2$, and Alice and Bob share the EPR state $|\Psi^+\rangle$. We call the following protocol quantum teleportation:

Step 1. Alice makes the Bell measurement to the qubit she has of the EPR pair and the quantum state she wants to send.

Step 2. Alice sends the measurement output to Bob.

Step 3. Bob executes one of the Pauli gates according to the measurement output.

The bell state also plays other vital roles in the superdense coding plotocol [73] and the E91 protocol [36].

2.1.5 Indirect Measurement Model

We show how we implement any measurement using a projective measurement and *indirect measurement model*. First, a new external system, the *ancilla* system, is added to the measured system. The following unitary operators can be consisted for a measurement $\{M_m\}_m$ that we want to perform:

$$U |\psi\rangle_t |0\rangle_a = \sum_m M_m |\psi\rangle_t |m\rangle_a \,, \tag{2.40}$$

where the index t denotes the system to be measured and the index a denotes the ancilla system, and $\{|m\rangle_a\}_m$ is CONS of the ancilla system. Then, a projective measurement P is performed on the created state. Using the projector $P_m^a \equiv |m\rangle \langle m|_a$ for the ancillary system, the projector $P_m^{ta} \equiv I_t \otimes |m\rangle \langle m|_a$ for the whole system can be constructed. The measurement probabilities $p(m)$ using the projection measurement $\{P_m^{ta}\}$ subjected to the projectors P_m are calculated as follows:

$$\begin{aligned} p(m) &= \langle\psi|_t \langle 0|_a U^\dagger P_m^{ta} U |\psi\rangle_t |0\rangle \\ &= \sum_{m',m''} \langle\psi| M_{m'}^\dagger \langle m'| (I_t \otimes |m\rangle \langle m|) M_{m''} |\psi\rangle |m''\rangle \\ &= \langle\psi| M_m^\dagger M_m |\psi\rangle . \end{aligned} \tag{2.41}$$

This probability is equal to Equation (2.15). In addition, the post-measurement state is

$$\frac{P_m U |\psi\rangle |0\rangle}{\sqrt{\langle\psi|_t \langle 0|_a U^\dagger P_m U |\psi\rangle_t |0\rangle_a}} = \frac{M_m |\psi\rangle |m\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{2.42}$$

By ignoring the ancilla system, it can be written as follows:

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{2.43}$$

This state is equal to Equation (2.16). Therefore, any measurement, including a POVM measurement, can be performed using a projective measurement and an ancilla system.

2.1.6 Density Operator

The representation of a quantum state in Postulate (2.3) we call a *pure state*. However, specific state preparations cannot be represented by using pure states only. For example, consider preparing state $|0\rangle$ with probability 1/2 and state $|1\rangle$ with probability 1/2. Any binary projective measurement for this state will take a random value 0/1 at the output. No matter what α and β we choose, we can not represent this state by $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$. Let us introduce the concept of a *density operator*.

Postulate 2.7 (Density Operator)**.** A quantum state ensemble $\{p_i, |\psi_i\rangle\}$, which prepare states $|\psi_i\rangle$ with probability p_i, is represented as a density operator $\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$.

Intuitively, a density operator is an expression of stochastic state preparation. As will be discussed below, at the same time, the density operator is a description of a quantum state that lacks information. A pure state can be considered a special case of a density operator $\rho = |\psi\rangle \langle\psi|$, i.e., a state $|\psi\rangle$ is prepared with probability 1.

A density operator is equal to a positive semi-definite matrix and trace 1. The trace of a density operator ρ is 1:

$$\begin{aligned} \mathrm{tr}[\rho] =& \mathrm{tr}[\sum_i p_i |\psi_i\rangle \langle\psi_i|] \\ =& \sum_i p_i \mathrm{tr}[|\psi_i\rangle \langle\psi_i|] \\ =& \sum_i p_i \langle\psi_i|\psi_i\rangle \\ =& \sum_i p_i \\ =& 1, \end{aligned} \tag{2.44}$$

and a density operator is a positive semi-definite matrix:

$$\begin{aligned} \langle\phi| \rho |\phi\rangle =& \langle\phi| (\sum_i p_i |\psi_i\rangle \langle\psi_i|) |\phi\rangle \\ =& \sum_i p_i \langle\phi| (|\psi_i\rangle \langle\psi_i|) |\phi\rangle \\ =& \sum_i p_i | \langle\phi|\psi_i\rangle |^2 \\ \geq& 0, \end{aligned} \tag{2.45}$$

where $|\phi\rangle$ is any quantum state. Conversely, a positive semi-definite matrix ρ is Hermitian and thus has a spectral decomposition $\rho = \sum_i \lambda_i |\psi_i\rangle \langle\psi_i|$ and $\lambda_i \geq 0$. In addition, from the conditions $\mathrm{tr}[\rho] = 1$, the properties of probability $\sum_i \lambda_i = 1$ also holds. Therefore, this positive semi-definite matrix ρ can be understood to be an ensemble of states $\{\lambda_i, |\psi\rangle\}$.

The extension of a quantum state to a density matrix would also require the extension of other postulates.

Postulate 2.8 (Evolution for density operator)**.** The relationship between the density operator ρ at time t_1 and the density operator ρ' at time t_2 is written by the unitary operator U which depends only on times t_1 and t_2 as follows:

$$\rho' = U\rho U^{\dagger}. \tag{2.46}$$

Postulate 2.9 (Measurement for density operator)**.** When a measurement $\{M_n\}$ is performed on a density operator ρ, the probability that the measurement output is n is given by

$$p(n) = \mathrm{tr}[M_n^{\dagger} M_n \rho]. \tag{2.47}$$

The density operator after the measurement is

$$\frac{M_n \rho M_n^{\dagger}}{\sqrt{\mathrm{tr}[M_n^{\dagger} M_n \rho]}}. \tag{2.48}$$

Postulate 2.10 (Composite system for density operator)**.** The state space of a composite system is described by the tensor product of the individual state spaces. A quantum state is called a separable state if it can be written as follows:

$$\rho = \sum_i p_i \rho^i \tag{2.49}$$

$$= \sum_i p_i (\rho_1^i \otimes \rho_2^i \otimes \cdots \otimes \rho_n^i), \tag{2.50}$$

where p_i denotes the probability of being in state ρ^i, and ρ_1^i, ρ_2^i and ρ_n^i denotes the state of the each system 1, 2, and n in the state ρ^i. A state that cannot be written in the form of Equation (2.50) is called an entangled state.

2.1.7 Bloch Ball

Here we introduce the Bloch ball, an intuitive notation for the quantum state of a single qubit. A pure state of a qubit system can be represented as a unit vector on a sphere, as follows:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle, \tag{2.51}$$

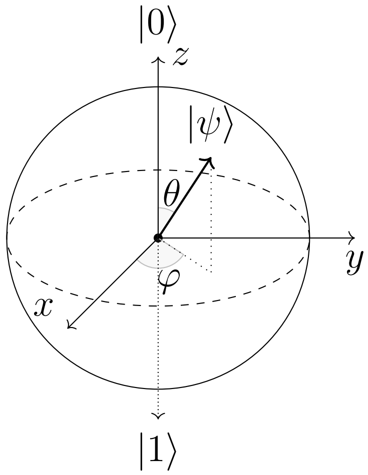


Figure 2.2: Bloch ball

where $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. Bloch ball is visualized in Figure 2.2. The intersections of the z-axis and the Bloch sphere are quantum states $|0\rangle$ and $|1\rangle$, the eigenstates of the Pauli Z. Similarly, in the x-axis and y-axis, the points of intersection with the sphere are the eigenstates of Pauli X and Pauli Y, respectively.

What does the interior of the Bloch ball have to do with a quantum state? The interior of the Bloch ball corresponds to a density operator. A density operator can be written using the Bloch vector $\vec{n} \in \mathbb{R}^3$ satisfying $\|\vec{n}\|^2 \leq 1$, as follows:

$$\rho = \frac{1}{2}(I + \vec{n} \cdot \vec{\sigma}), \tag{2.52}$$

where $\vec{\sigma} = (X, Y, Z)$. The center of the Bloch ball can be written as $\rho_o = \frac{I}{2}$ and is called the *maximally mixed state*.

2.1.8 Quantum One-time Pad

An important encryption method for various quantum cryptographic protocols, including blind quantum computation protocols, is the *quantum one-time pad* [9]. In classical cryptography, there is an encryption method called the one-time pad [74]. The one-time pad is an encryption method that "covers" n-bits message $m \in \{0,1\}^n$ with a key that is the same length as the message. First, Alice selects a random sequence of bits $k \in \{0,1\}^n$ as a key and shares the key with Bob. Alice computes the exclusive disjunction of the message and key, as follows:

$$m' = Enc(m, k) = m \oplus k. \tag{2.53}$$

Then Alice sends it to Bob. Bob computes the exclusive disjunction of the key with the ciphertext from Alice and the output is the original message, as follows:

$$\begin{aligned} Dec(m', k) =& m' \oplus k \\ =& m \oplus k \oplus k \\ =& m. \end{aligned} \tag{2.54}$$

If the key is generated with appropriate randomness, the following equation holds:

$$H(M) = H(M|C = m'), \tag{2.55}$$

where $H(M)$ is the Shannon entropy of the plaintext and $H(M|C = m')$ is the conditional Shannon entropy of the plaintext given the ciphertext $C = m'$. This means that even if Eve, the adversary, learns the ciphertext, the Shannon entropy of the plaintext is not reduced, meaning that it is independent of the ciphertext. That is, this cryptography is secure no matter what kind of computational ability Eve has, and such security is called information-theoretic security.

The quantum one-time pad is the quantum counterpart of the (classical) one-time pad. Alice wants to send one qubit $|\psi\rangle$ to Bob. Alice randomly generates two 1 bit keys $k_x \in \{0, 1\}$ and $k_z \in \{0, 1\}$, and shares them with Bob. Alice encrypts the quantum state as follows:

$$|\psi'\rangle = Enc(|\psi\rangle, k_x, k_z) = X^{k_x} Z^{k_z} |\psi\rangle. \tag{2.56}$$

Then Alice sends it to Bob. Since Bob knows the keys, he can decrypt the quantum state by executing quantum gates as follows:

$$\begin{aligned} Dec(|\psi'\rangle, k_x, k_z) =& Z^{k_z} X^{k_x} |\psi'\rangle \\ =& Z^{k_z} X^{k_x} X^{k_x} Z^{k_z} |\psi\rangle \\ =& |\psi\rangle. \end{aligned} \tag{2.57}$$

Suppose Eve, who does not know the keys, obtained the encrypted state. For Eve, the state is the following ensemble:

$$\begin{aligned}\rho =&\frac{1}{4}\sum_{x,z\in\{0,1\}} X^{k_x}Z^{k_z}\left|\psi\right\rangle\left\langle\psi\right| Z^{k_z}X^{k_x}\\ =&\frac{1}{4}\sum_{x,z\in\{0,1\}} X^{k_x}Z^{k_z}(\alpha\left|0\right\rangle+\beta\left|1\right\rangle)(\bar{\alpha}\left|0\right\rangle+\bar{\beta}\left|1\right\rangle)Z^{k_z}X^{k_x}\\ =&\frac{1}{2}(\left|0\right\rangle\left\langle 0\right|+\left|1\right\rangle\left\langle 1\right|)\\ =&\frac{1}{2}I.\end{aligned} \tag{2.58}$$

For any given initial state $|\psi\rangle$, Eve gets the maximally mixed state. That is, Eve cannot gain any information from the state $|\psi'\rangle$.

2.1.9 No-cloning Theorem

One significant result in quantum information is the *no-cloning theorem* [28–30].

Theorem 2.11 (No-cloning theorem)**.** There is no unitary operation to clone any quantum state. In other words, there is no unitary operation U that causes the following state changes:

$$U(\left|\psi\right\rangle\left|0\right\rangle)=\left|\psi\right\rangle\left|\psi\right\rangle \tag{2.59}$$

for any state $|\psi\rangle$.

Proof. Suppose that such a unitary operation U exists. The inner product of copying another two states $|\psi\rangle$ and $|\phi\rangle$ is

$$\begin{aligned}\left\langle\psi\right|\left\langle 0\right|\left|\phi\right\rangle\left|0\right\rangle =&\left\langle\psi|\phi\right\rangle\\ =&\left\langle\psi\right|\left\langle 0\right|U^\dagger U\left|\phi\right\rangle\left|0\right\rangle\\ =&\left\langle\psi\right|\left\langle\psi\right|\left|\phi\right\rangle\left|\phi\right\rangle\\ =&(\left\langle\psi|\phi\right\rangle)^2.\end{aligned} \tag{2.60}$$

However, for this equation to hold, the two states must have an inner product of 0 or 1. This means that the two states must be the same or orthogonal. This is against the assumption. Therefore, there is no unitary operation that clones an arbitrary quantum state. □

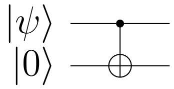


Figure 2.3: Quantum circuit for copying $|0\rangle$ and $|1\rangle$

From the above proof, it can be said that only sets of states that are orthogonal to each other can be cloned. For example, if it is known that the state is either state $|0\rangle$ or $|1\rangle$, it can easily be replicated using a $CNOT$ gate, as shown in Figure 2.3. It can be replicated as follows:

$$CNOT(|0\rangle|0\rangle) = |0\rangle|0\rangle, \tag{2.61}$$

$$CNOT(|1\rangle|0\rangle) = |1\rangle|1\rangle. \tag{2.62}$$

In contrast, if we try to copy a state $|+\rangle$ using a $CNOT$ gate, we get the following:

$$CNOT(|+\rangle|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle). \tag{2.63}$$

As $|+\rangle|+\rangle \neq \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$, the state $|+\rangle$ is not cloned.

The no-cloning theorem can be considered with several extensions. For example, by adding an ancilla system, we can consider the following copy:

$$U|\psi_1\rangle|0\rangle|a_0\rangle = |\psi_1\rangle|\psi_1\rangle|a_1\rangle, \tag{2.64}$$

where $|a_0\rangle$ and $|a_1\rangle$ denotes quantum states of the ancilla system. Even such a copy is impossible. Furthermore, one can consider the extension to the case where quantum states are mixed states, but even it is also known that in this case, cloning (broadcasting) is not possible [75–77].

There are pros and cons to the fact that quantum states cannot be replicated. In quantum cryptography, the no-cloning theorem provides strong security. For example, Eve, the adversary, can not deceive Alice or Bob using replicated quantum states. It is also not possible to obtain much information from replicated quantum states. In contrast, storing the states under calculation is impossible in quantum computation. It is also impossible to make them more resistant to noise by copying quantum states. These are easy in classical computers.

2.2 Quantum Computation

We give an overview of the foundations of quantum computation in this section.

2.2.1 Quantum Gate

We saw that the counterpart to a classical *bit* is a *qubit* in Subsection 2.1.1. The smallest computational element in classical computation is logic gates. Similarly, the smallest computational element in quantum computation is quantum gates.

We have already seen several quantum gates in Subsection 2.1.2. The Pauli gates are typical quantum gates. A generalized quantum gate of the Pauli gate is the *rotation* gate. A generalization of the Pauli X gate is the following X-axis rotation gate:

$$\begin{aligned} R_x(\theta) \equiv& e^{-i\theta X/2} \\ =& \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X \\ =& \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}. \end{aligned} \tag{2.65}$$

This x-axis rotation gate corresponds to the rotational operation of the angle θ around the x-axis of the Bloch sphere. Similarly, generalizations of the Pauli Y and Z gates are the following y-axis and z-axis rotation gates:

$$\begin{aligned} R_y(\theta) \equiv& e^{-i\theta Y/2} \\ =& \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Y \\ =& \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \end{aligned} \tag{2.66}$$

$$\begin{aligned} R_z(\theta) \equiv& e^{-i\theta Z/2} \\ =& \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z \\ =& \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \end{aligned} \tag{2.67}$$

In addition, these gates can be extended to rotational operations on any axis. The rotation gate for any given axis rotation is as follows:

$$\begin{aligned} R_{\vec{n}}(\theta) &\equiv \exp(-i\theta\vec{n}\cdot\vec{\sigma}) \\ &= \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}(n_xX + n_yY + n_zZ), \end{aligned} \tag{2.68}$$

where $\vec{n} = (n_x, n_y, n_z)$ is Bloch vector and $\vec{\sigma} = (X, Y, Z)$. In other words, this rotation gate represents the rotational operation of the Bloch sphere around the $\vec{n}$-axis.

There are other essential quantum gates besides the Pauli gates. They are the H gate (Hadamard gate), the T gate ($\frac{\pi}{8}$ gate), and the S gate (phase gate). Recall that the H gate is Equation (2.13). The T gate is represented by the following matrix:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \tag{2.69}$$

and the S gate is represented by the following matrix:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{2.70}$$

Note that the T and S gates are rotations of angle $\frac{\pi}{2}$ and $\frac{\pi}{4}$ on the z-axis respectively. Therefore, the following equality holds, which we also use in our protocols:

$$T^2 = S, \tag{2.71}$$

$$S^2 = Z, \tag{2.72}$$

$$Z^2 = I. \tag{2.73}$$

It is also known that any single-qubit gate can be constructed from rotation gates around the y-axis and z-axis [78].

Theorem 2.12. Suppose U is a single-qubit gate. There exist real numbers α, β, γ, and δ such that

$$U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta). \tag{2.74}$$

Proof. U is a unitary operator, hence its rows and columns are orthonormal. Therefore, it can be written as the following matrix:

$$\begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)}\cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)}\cos\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)}\sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)}\sin\frac{\gamma}{2}. \end{bmatrix} \tag{2.75}$$

This matrix is obviously equal to the matrix in Equation (2.74). □

There is the following relationship between the Pauli gates, H gate, and S gate:

$$\begin{aligned} HXH =& Z, & (2.76)\\ HYH =& -Y, & (2.77)\\ HZH =& X, & (2.78)\\ SXS^{\dagger} =& Y, & (2.79)\\ SYS^{\dagger} =& -X, & (2.80)\\ SZS^{\dagger} =& Z. & (2.81) \end{aligned}$$

The above quantum gates are single-qubit gates acting on a single-qubit system. There are also quantum gates that act on multiple quantum states. The $CNOT$ gate and the CZ gate are essential in quantum computation as two-qubit gates. The $CNOT$ gate, already introduced in Subsection 2.1.9, is the following matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.82}$$

and the CZ gate is the following matrix:

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{2.83}$$

$CNOT$ and CZ gates are special cases of controlled-U gates. The controlled-U gate refers to the first qubit and performs the unitary operator U on the second qubit. That is, if the first qubit is $|0\rangle$, the identity gate is executed on the second qubit, and if the first qubit is $|1\rangle$, the unitary gate U is executed on the second qubit. When $U = X$, a controlled-U gate is $CNOT$ gate, and when $U = Z$, a controlled-U gate is CZ gate. In addition, from Equation (2.76), the following relationship exists between the $CNOT$ and CZ gates:

$$\begin{aligned} CNOT &= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \\ &= |0\rangle\langle 0| \otimes HIH + |1\rangle\langle 1| \otimes HZH \\ &= I \otimes H(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z)I \otimes H \\ &= I \otimes H(CZ)I \otimes H \end{aligned} \tag{2.84}$$

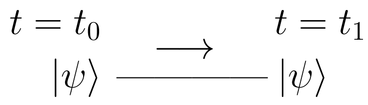


Figure 2.4: Quantum wire

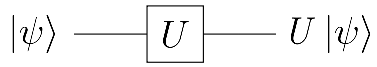


Figure 2.5: Single-qubit gate

The important three-qubit gate is the *Toffoli* gate. The *Toffoli* gate is the following matrix:

$$\mathit{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.85}$$

The *Toffoli* gate applies the X gate to the third qubit only when the states of the first and second qubits are both $|1\rangle$. This means *Toffoli* gate can also be interpreted as $CCNOT/CC\text{-}X$ gate.

2.2.2 Quantum Circuit

A combination of multiple quantum gates is necessary to perform complex operations on qubits. *Quantum circuit* is a simple framework for understanding the application of multiple quantum gates to multiple qubits. In addition, the quantum circuit can be considered the quantum counterpart of a classical electronic circuit.

We use the *quantum wire* for the passage of time in quantum states, as shown in Figure 2.4. On the quantum circuit, a quantum state enters the quantum wire from the left and progresses to the right over time. Figure 2.5 shows a quantum gate U for a quantum state $|\psi\rangle$.

How can a two-qubit gate be described in a quantum circuit model? In a quantum circuit model, the $CNOT$ gate is described in Figure 2.6. Similarly, the CZ gate is described in Figure 2.7. The second

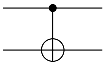


Figure 2.6: *CNOT* gate in a quantum circuit

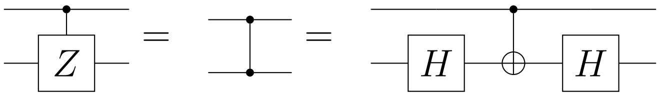


Figure 2.7: *CZ* gate in a quantum circuit

expression comes from the fact that, with the CZ gate, the first or the second qubit can be chosen as a *control* qubit. In addition, any single-qubit gate can be decomposed as $U = e^{i\alpha}AXBXC$, where A, B and C satisfy $ABC = I$, so that a controlled-U gate can be performed with the quantum circuit of Figire 2.8.

It is common in quantum circuits to use $|0\rangle$ as an initial quantum state. In quantum circuits, the measurement is the projective measurement with a computational basis (computational basis measurement). In the quantum circuit model, the computational basis measurement is shown in Figure 2.9. These rules are natural generalizations of classical circuits.

In quantum theory, we can use many different kinds of projective measurements, for example, the Bell measurement. In the quantum circuit model, Various projection measurements are performed using unitary gates and computational basis measurements. For example, if we want to measure X, we can use a H gate, as shown in Figure 2.10. This means that we interpret an output 0 as $|+\rangle = H|0\rangle$ and an output 1 as $|-\rangle = H|1\rangle$ from the computational basis measurement. Similarly, the Bell measurement can be performed with the quantum circuit in Figure 2.11. This can be seen from the following relationship between the computational basis and the Bell basis:

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = H \otimes I(CNOT)\,|00\rangle\,, \tag{2.86}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = H \otimes I(CNOT)\,|10\rangle\,, \tag{2.87}$$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = H \otimes I(CNOT)\,|01\rangle\,, \tag{2.88}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = H \otimes I(CNOT)\,|11\rangle\,. \tag{2.89}$$

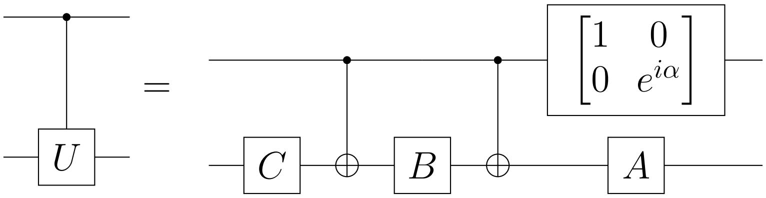


Figure 2.8: Controlled-U gate in a quantum circuit

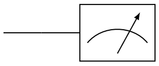


Figure 2.9: Computational basis measurement in a quantum circuit

Let us introduce two quantum gates using quantum circuits. The first one is the *Toffoli* gate. The *Toffoli* gate can be implemented in the quantum circuit of Figure 2.12 by combining the single-qubit gates and two-qubit gates. The other is the *SWAP* gate. The *SWAP* gate is a quantum gate that *swaps* states between two qubits as follows:

$$SWAP(|\psi\rangle_1 |\phi\rangle_2) = |\phi\rangle_1 |\psi\rangle_2 , \tag{2.90}$$

where the index 1 and 2 denote the system of these qubits. That is, the *SWAP* gate can be interpreted as an operation that swaps an "arrangement" of quantum states. The *SWAP* gate can be implemented by combining three *CNOT* gates, as shown in Figure 2.13.

2.2.3 Universal Gate Set

In classical circuits, we can perform any classical calculation if we have NAND gates. Are there any quantum gates that play a similar role in quantum circuits? Such a set of quantum gates exists and is called a *universal gate set*.

Decomposition into Two-level Unitary Operations

It is known that any unitary operation can be decomposed into two-level unitary operations [79]. As the first step, we decompose a 3×3 unitary operation into three 2-level unitary operations. Any 3×3

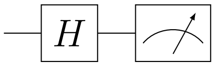


Figure 2.10: Quantum circuit for the X measurement

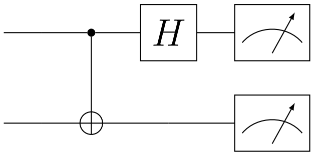


Figure 2.11: Quantum circuit for the Bell measurement

unitary operation can be written as the following matrix:

$$U = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & j \end{bmatrix}. \tag{2.91}$$

If $b \neq 0$, we set U_1 as the following matrix:

$$U_1 \equiv \begin{bmatrix} \frac{\bar{a}}{\sqrt{|a|^2+|b|^2}} & \frac{\bar{b}}{\sqrt{|a|^2+|b|^2}} & 0 \\ \frac{b}{\sqrt{|a|^2+|b|^2}} & \frac{-a}{\sqrt{|a|^2+|b|^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.92}$$

If $b = 0$, we set U_1 as the identity matrix, as follows:

$$U_1 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.93}$$

When calculating the matrix product of such U_1 and U, it becomes the following matrix:

$$U_1 \cdot U = \begin{bmatrix} a' & d' & g' \\ 0 & e' & h' \\ c' & f' & j' \end{bmatrix}. \tag{2.94}$$

Similarly, if $c' \neq 0$, we set U_2 as the following matrix:

$$U_2 \equiv \begin{bmatrix} \frac{\bar{a}'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{\bar{c}'}{\sqrt{|a'|^2+|c'|^2}} \\ 0 & 1 & 0 \\ \frac{c'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{-a'}{\sqrt{|a'|^2+|c'|^2}} \end{bmatrix}. \tag{2.95}$$

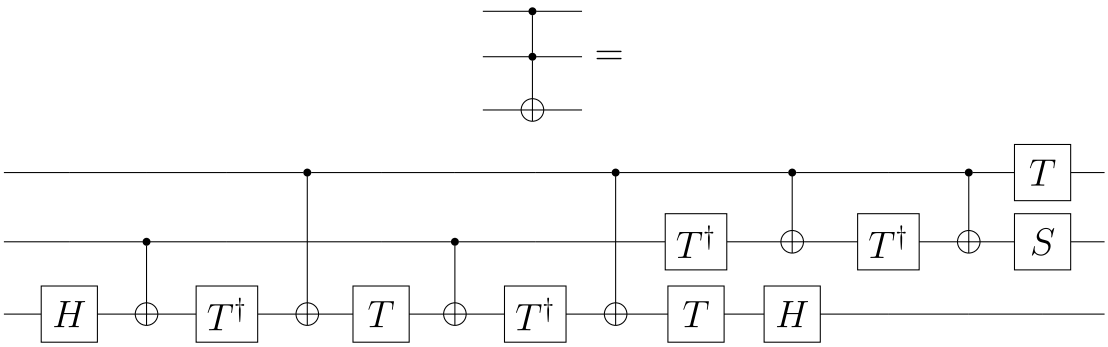


Figure 2.12: Quantum circuit for the *Toffoli* gate

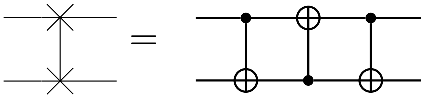


Figure 2.13: Quantum circuit for the *SWAP* gate

If $c' = 0$, we set U_2 as the following matrix:

$$U_2 \equiv \begin{bmatrix} \bar{a}' & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.96}$$

When calculating the matrix product of such U_1, U_2 and U, it becomes the following matrix:

$$U_2 \cdot U_1 \cdot U = \begin{bmatrix} 1 & d'' & g'' \\ 0 & e'' & h'' \\ 0 & f'' & j'' \end{bmatrix}. \tag{2.97}$$

As U_1, U_2, and U are unitary matrices, $U_2 \cdot U_1 \cdot U$ is also a unitary matrix. Given this property, we can conclude that $d'' = 0$ and $g'' = 0$. Then, we set U_3 as the following matrix:

$$U_3 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{e}'' & \bar{f}'' \\ 0 & \bar{h}'' & \bar{j}'' \end{bmatrix}. \tag{2.98}$$

When calculating the matrix product of such U_1, U_2, U_3 and U, it becomes the identity matrix, as follows:

$$U_3 \cdot U_2 \cdot U_1 \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.99}$$

Therefore, the following equation holds immediately:

$$U = U_1^\dagger \cdot U_2^\dagger \cdot U_3^\dagger. \tag{2.100}$$

Since U_1, U_2, and U_3 are all two-level unitary operations, the 3×3 unitary operation U is decomposed into two-level unitary operations. This decomposition can be easily extended to $d \times d$ unitary operations for any dimensions d. Specifically, let us consider the following $d \times d$ unitary matrix:

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1d} \\ 0 & u_{22} & & \\ \vdots & & \ddots & \vdots \\ 0 & & & \\ u_{j1} & & & \\ \vdots & & & \\ u_{d1} & \cdots & & u_{dd} \end{bmatrix}. \tag{2.101}$$

In other words, it is a unitary matrix where the elements from the first row to the j-1th row in the first column are zero. Then we set a $d \times d$ unitary operation U_j as the following matrix:

$$U_j \equiv \begin{bmatrix} \frac{u\bar{}_{11}}{\sqrt{|u_{11}|^2+|u_{j1}|^2}} & 0 & \cdots & 0 & \frac{u\bar{}_{j1}}{\sqrt{|u_{11}|^2+|u_{j1}|^2}} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & & \vdots \\ \vdots & & \ddots & & & & & \\ 0 & & & & & & & \\ \frac{u_{j1}}{\sqrt{|u_{11}|^2+|u_{j1}|^2}} & 0 & \cdots & & \frac{-u_{11}}{\sqrt{|u_{11}|^2+|u_{j1}|^2}} & & \cdots & 0 \\ 0 & & \cdots & & & \ddots & & \\ \vdots & & & & & & 1 & 0 \\ 0 & & \cdots & & & & 0 & 1 \end{bmatrix}. \tag{2.102}$$

When calculating the matrix product of such U_j and U, it becomes the following matrix:

$$U_j \cdot U = \begin{bmatrix} u'_{11} & u'_{12} & \cdots & u'_{1d} \\ 0 & u'_{22} & & \\ \vdots & & \ddots & \vdots \\ 0 & & & \\ 0 & & & \\ u'_{j+11} & & & \\ \vdots & & & \\ u'_{d1} & \cdots & & u'_{dd} \end{bmatrix}. \tag{2.103}$$

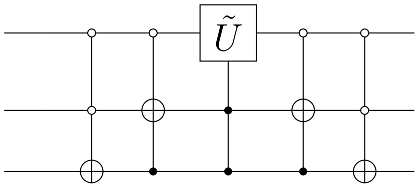


Figure 2.14: Quantum circuits for performing U

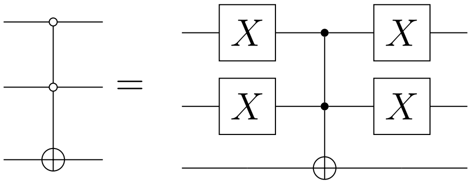


Figure 2.15: *Toffoli* gate refer $|0\rangle$ instead of $|1\rangle$

In this way, for any $d \times d$ matrix, we can construct a unitary matrix such that all the elements from the second row onward in the first column become zero when multiplied with that matrix.

Decomposition into *CNOT* gate and single-qubit gate

It is known that any two-level unitary operations can be performed using single-qubit gates and *CNOT* gates [80]. For a 3-qubit state, we consider a two-level unitary operation U that acts only on states $|000\rangle$ and $|111\rangle$. This unitary operation U can be written in the following matrix:

$$U = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix}. \tag{2.104}$$

If we consider that the non-trivial operation of U is an operation on a single qubit $\tilde{U}$, it can be written with the following unitary matrix:

$$\tilde{U} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}. \tag{2.105}$$

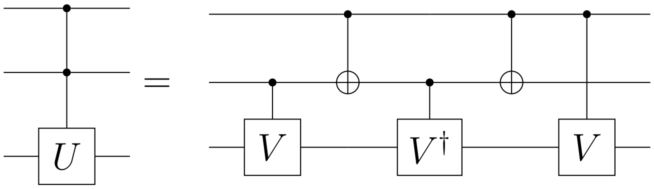


Figure 2.16: Quantum circuit for a *CC-U* gate. V is any unitary operation satisfying $U = V^2$

We introduce a method to apply U to states $|000\rangle$ and $|111\rangle$ by transferring state $|000\rangle$ to state $|011\rangle$. Specifically, the method can be implemented by using a quantum circuit, as shown in Figure 2.14. We have already shown that the *Toffoli* gate can consist of *CNOT* gates and single-qubit gates in Figure 2.12. From the Toffoli gate, as shown in Figure 2.15, we can create a controlled-controlled-X gate that changes the reference state from $|1\rangle$ to $|0\rangle$. Additionally, the *CC-U* gate can also be constructed from single-qubit gates and *CNOT* gates, as shown in Figure 2.16.

In the above, we have seen a specific circuit for the case of three qubits. Generally, it can be said that a circuit needs to be prepared that exchanges the basis so that the target on which U acts becomes a single qubit. Such a circuit is classical and easy to implement on a quantum circuit [78].

Approximation of Single-qubit Gate

It was shown that two-level unitary gates can be implemented with single qubit gates and *CNOT* gates [81]. Is it possible to implement an arbitrary single-qubit gate? In fact, it is known that if we can implement some single-qubit gates, we can construct any single-qubit gate with arbitrary accuracy.

First, let us consider the approximation of unitary gates. The *error* when unitary gate V is executed instead of unitary gate U is defined as follows:

$$E(U,V) \equiv \max_{|\psi\rangle} \|(U - V)\,|\psi\rangle\,\|, \tag{2.106}$$

where the maximum is over all quantum states $|\psi\rangle$. The error is important because it sets an upper limit on the difference in the probability of measurement outputs for states obtained by executing the two gates. Let M be the POVM elements and the respective measurement probabilities be $P_U = \langle\psi| U^\dagger M U |\psi\rangle$ and $P_V = \langle\psi| V^\dagger M V |\psi\rangle$. The limit is given as follows:

$$\begin{aligned}
|P_U - P_V| =& |\, \langle\psi| U^\dagger M U |\psi\rangle - \langle\psi| V^\dagger M V |\psi\rangle \,| \\
=& |\, \langle\psi| U^\dagger M |\Delta\rangle + \langle\Delta| M V |\psi\rangle \,| \\
\leq& |\, \langle\psi| U^\dagger M |\Delta\rangle \,| + |\, \langle\Delta| M V |\psi\rangle \,| \\
\leq& \|\, |\Delta\rangle \,\| + \|\, |\Delta\rangle \,\| \\
\leq& 2E(U, V),
\end{aligned} \tag{2.107}$$

where $|\Delta\rangle \equiv (U - V)|\psi\rangle$. Also, the following inequality holds for the error:

$$E(U_1 U_2 \dots U_m, V_1 V_2 \dots V_m) \leq \sum_{i=1}^{m} E(U_i, V_i). \tag{2.108}$$

This means that the error of the entire unitary gates in a quantum circuit is capped at the sum of the errors of the individual quantum gates. The inequality in Equation (2.108) is evident because the following inequality holds when $m = 2$:

$$\begin{aligned}
E(U_1 U_2, V_1 V_2) =& \|(U_2 U_1 - V_2 V_1)|\psi\rangle\,\| \\
=& \|(U_2 U_1 - V_2 U_1)|\psi\rangle + (V_2 U_1 - V_2 V_1)|\psi\rangle\,\| \\
\leq& \|(U_2 - V_2)U_1|\psi\rangle\,\| + \|V_2(U_1 - V_1)|\psi\rangle\,\| \\
\leq& E(U_2, V_2) + E(U_1, V_1).
\end{aligned} \tag{2.109}$$

Then, we show that by using the H and T gates, an arbitrary single-qubit gate can be approximated with arbitrary accuracy under the error. We can construct the following quantum gates using the H and T gates (and the $S = T^2$ gate):

$$R_z\left(\frac{-\pi}{4}\right) = T^\dagger, \tag{2.110}$$

$$R_x\left(\frac{\pi}{4}\right) = HTH, \tag{2.111}$$

$$R_x\left(\frac{-\pi}{4}\right) = HT^\dagger H, \tag{2.112}$$

$$R_y\left(\frac{\pi}{4}\right) = SHTHS^\dagger, \tag{2.113}$$

$$R_y\left(\frac{-\pi}{4}\right) = SHT^\dagger HS^\dagger. \tag{2.114}$$

Let us consider the following rotation gate, which is a combination of the H and the T gates:

$$\begin{aligned} T^\dagger HTH =& R_z\left(\frac{-\pi}{4}\right) R_x\left(\frac{\pi}{4}\right) \\ =& \left(\cos\frac{\pi}{8}I + i\sin\frac{\pi}{8}Z\right)\left(\cos\frac{\pi}{8}I - i\sin\frac{\pi}{8}X\right) \\ =& \cos^2\frac{\pi}{8}I - i\left(\cos\frac{\pi}{8}(X-Z) - \sin\frac{\pi}{8}Y\right)\sin\frac{\pi}{8}. \end{aligned} \tag{2.115}$$

This rotation gate can be interpreted as a quantum gate that rotates by an angle θ, which is defined by $\cos\frac{\theta}{2} \equiv \cos^2\frac{\pi}{8}$, around the $\vec{n}$-axis on the Bloch sphere, where $\vec{n} \equiv (\cos\frac{\pi}{8}, -\sin\frac{\pi}{8}, -\cos\frac{\pi}{8})$.

It is known that this θ is an irrational multiple of 2π [81]. By repeatedly applying $R_{\vec{n}}(\theta)$, it is possible to implement the rotation gate $R_{\vec{n}}(\alpha)$ with an arbitrary angle α to any accuracy. This can be understood by dividing 2π into a finite number of intervals based on the desired error δ and then applying the pigeonhole principle, along with the fact that θ is an irrational multiple of 2π.

Let us consider the following new quantum gate:

$$\begin{aligned} & R_{\vec{m}}(\theta) = \\ & R_y\left(\frac{\pi}{4}\right) R_z\left(\frac{-\pi}{2}\right) R_y\left(\frac{-\pi}{4}\right) R_z\left(\frac{-\pi}{4}\right) R_x\left(\frac{\pi}{4}\right) R_y\left(\frac{\pi}{4}\right) R_z\left(\frac{\pi}{2}\right) R_y\left(\frac{-\pi}{4}\right), \end{aligned} \tag{2.116}$$

where $\vec{m} \equiv (\frac{1}{2}\sin\frac{\pi}{8}, \cos\frac{\pi}{8}, -\frac{1}{2}\sin\frac{\pi}{8})$. Using this rotation gate $R_{\vec{m}}(\theta)$, just as in the discussion for the $\vec{n}$-axis, any rotation of an arbitrary angle around the $\vec{m}$-axis can be executed with any desired accuracy. The $\vec{n}$-axis and $\vec{m}$-axis are easily seen to be orthogonal. In Theorem 2.12, we showed that if rotation gates of any angle can be executed on two orthogonal axes (specifically, the y-axis and z-axis in the theorem), then any single-qubit gate can be implemented. Thus, by using T and H, we can implement any single-qubit gate with arbitrary accuracy regarding the error.

For simplicity in our proof, we chose orthogonal vectors $\vec{n}$ and $\vec{m}$. It is known that if $\vec{n}$ and $\vec{m}$ are not parallel, they can be combined to reproduce a rotation about any axis [82, 83]. In subsequent discussions, we use $R_{\vec{n}} = THTH$ and $R_{\vec{m}} = HTHT$. Since $\vec{n} = (\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})$ and $\vec{m} = (\cos\frac{\pi}{8}, -\sin\frac{\pi}{8}, \cos\frac{\pi}{8})$, therefore $\vec{n}$ and $\vec{m}$ are not parallel.

Universal Gate Sets and Gottesman—Knill Theorem

Based on the above discussion, it is evident that $\{CNOT, H, T\}$ is a *universal gate set* that can approximate any unitary operation. In other words, in order to perform any quantum computation, it suffices to implement only this gate set.

While we adopted the $CNOT$ gate as the two-qubit gate in the universal gate set, it is not necessarily required. It is clear from Equation 2.84 that gate set $\{CZ, H, T\}$, which adopts the CZ gate instead of the $CNOT$ gate, is also a universal gate set. Additionally, it is known that the gate set $\{\mathit{Toffoli}, H\}$ is also a universal gate set [84].

Conversely, is there a gate set that is not a universal gate set? A notable example of a non-universal gate set is the gate set consisting of *Clifford gates*.

Definition 2.13 (Clifford gate)**.** A unitary operator U that converts any Pauli gate into (possibly another) Pauli gate is called a *Clifford gate*. More precisely, for any P_1 that is a Pauli gate or a tensor product of Pauli gates, UP_1U' is also a Pauli gate or a tensor product of Pauli gates.

We have already seen several Clifford gates. For example, the H gate, S gate, and the $CNOT$ gate. However, the T gate is not a Clifford gate, as follows:

$$TXT^\dagger = \frac{X - Y}{\sqrt{2}}. \tag{2.117}$$

Quantum circuits consisting only of Clifford gates can be simulated in polynomial time by a classical computer. The fact is known as the Gottesman–Knill theorem [85].

Theorem 2.14 (Gottesman—Knill Theorem)**.** A quantum computer composed of the following elements can be simulated by a classical computer in polynomial time:

- the initial quantum state is only $|0\rangle$,
- the quantum gates are only Clifford gates,
- the quantum measurement is only computational basis measurements.

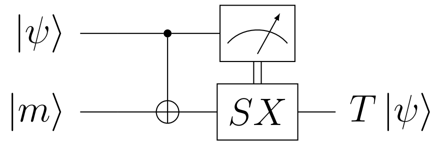


Figure 2.17: Quantum circuit for T gate by using the magic state.

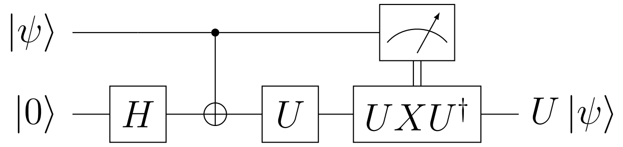


Figure 2.18: Quantum circuit for gate teleportation

Proof. When a Clifford gate executes on the eigenstates of the Pauli gates or tensor products of the eigenstates, it maps those quantum states to the eigenstates of the Pauli gates or their tensor products. On a classical computer, we consider a classical simulation where the eigenstates of the six types of Pauli gates are "updated" for each Clifford gate in the quantum circuit. Given the number of qubits n and the number of Clifford gates $poly(n)$, the simulation time can be computed in polynomial time, specifically at most $6 \times n \times poly(n)$. Therefore, a quantum computer composed of the above elements can be simulated on a classical computer in polynomial time. □

The T gate or the *Toffoli* gate differentiates quantum computation from classical one. But this does not mean that the T gate or the *Toffoli* gate are necessary for quantum computation. In fact, if we can avail a magic state $|m\rangle = \frac{1}{\sqrt{2}(|0\rangle+e^{i\pi/4}|1\rangle)}$ as an initial state or if we can perform arbitrary quantum measurements, any quantum computation can be performed only with Clifford gates.

Let us introduce a method to implement the T gate using magic states and Clifford gates [86,87]. Using the quantum circuit as shown in Figure 2.17, we can execute one T gate by consuming one magic state for any quantum state. The quantum circuit in Figure 2.17 can be seen as a special case of *gate teleportation* in Figure 2.18.

2.2.4 Measurement-based Quantum Computation

In this subsection, let us introduce measurement-based quantum computation [88]. Measurement-based quantum computation is a computational method proposed by Raussendorf and colleagues [57, 58]. The relationship between Circuit-based and measurement-based quantum computations is similar to that of the *Schrödinger picture* and the *Heisenberg picture.*

Measurement-based quantum computation uses $|+\rangle$ as the initial state and performs Clifford gates as quantum gates. As previously mentioned, we cannot perform arbitrary quantum computations only with these two elements. Let us assume that we can perform a measurement $\{P_0 \equiv (|0\rangle + e^{i\theta}|1\rangle)(\langle 0| + e^{-i\theta}\langle 1|), P_1 \equiv (|0\rangle - e^{i\theta}|1\rangle)(\langle 0| - e^{-i\theta}\langle 1|)\}$ at any arbitrary angle θ. Then we prepare states $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|+\rangle$, and execute the CZ gate on these qubits, as follows:

$$CZ(|\psi\rangle_1 |+\rangle_2) = \alpha |0\rangle_1 |+\rangle_2 + \beta |1\rangle_1 |-\rangle_2 . \tag{2.118}$$

We perform the previously mentioned projection measurement $\{P_0, P_1\}$ on the first qubit of this quantum state and assume the measurement output is 0. This means the first qubit is projected onto $|0\rangle + e^{i\theta}|1\rangle$, and the quantum state becomes as follows:

$$(|0\rangle_1 + e^{i\theta}|1\rangle_1)(\alpha|+\rangle_2 + \beta e^{-i\theta}|-\rangle_2) = (|0\rangle_1 + e^{i\theta}|1\rangle_1) H R_z(\theta)|\psi\rangle_2 . \tag{2.119}$$

Therefore, focusing only on the second qubit is equivalent to executing the quantum gate $M(\theta) \equiv HR_z(\theta)$ on the state $|\psi\rangle$. By combining this gate $M(\theta)$ with a specific θ, we can create the following quantum gate:

$$M(0) = H, \tag{2.120}$$

$$M(0)J\left(\frac{\pi}{4}\right) = HHT = T. \tag{2.121}$$

In contrast, if the measurement output is 1, the quantum state becomes the following state:

$$\begin{aligned}(\alpha|+\rangle_2 - \beta e^{-i\theta}|-\rangle_2) =& XHR_z(\theta)|\psi\rangle_2 \\ =& XM(\theta)|\psi\rangle_2 .\end{aligned} \tag{2.122}$$

Such the X gate is called a *byproduct operator*. This byproduct operator can be disregarded using the adjusted measurement $\{P_0' \equiv (|0\rangle + e^{-i\phi}|1\rangle)(\langle 0| + e^{i\phi}\langle 1|), P_1' \equiv (|0\rangle - e^{-i\phi}|1\rangle)(\langle 0| - e^{i\phi}\langle 1|)\}$, as follows:

$$X^b M(-\phi) X M(\theta)|\psi\rangle = X^b Z M(\phi) M(\theta)|\psi\rangle , \tag{2.123}$$

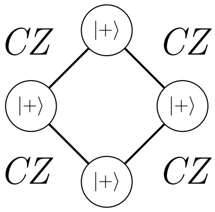


Figure 2.19: Graph state

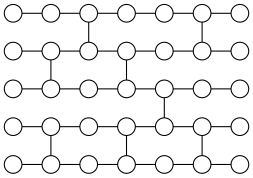


Figure 2.20: Circuit-like graph state

where b is the output of the adjusted measurement. This new Z gate is also a byproduct operator. These byproduct operators can be ignored when we perform the computational basis measurement.

In measurement-based quantum computation, a *graph state* is used instead of quantum circuits. As shown in Figure 2.19, the graph state is such a graph that has vertices in $|+\rangle$ and has edges in CZ gates. In measurement-based quantum computation, by using circuit-like graph states, as shown in Figure 2.20, programmable quantum computations similar to circuit-based quantum computation are possible.

2.3 Blind Quantum Computation Protocols

In this section, we provide an overview of blind quantum computation protocols and introduce several blind quantum computation protocols proposed in previous studies.

Unless otherwise specified, we refer to the delegator of the quantum computation as the *user* and the owner of the quantum server as the *server*. Let us reiterate the definition of the blind quantum computation protocol we defined in Introduction:

Definition 2.15 (Blind Quantum Computation Protocol)**.** Consider a delegation protocol P that can perform quantum computations executed by any polynomial-size quantum circuit. We denote the quantum circuit, an input to the protocol, by $X \in \{0,1\}^*$. Let

$L_{\mathrm{P}}(X)$ be defined as the size of the quantum circuit executed by the server when the input to the protocol P is X. We call the delegation protocol P as a blind quantum computation protocol if it satisfies the following conditions:

1. For any X_1 and X_2 such that $L_{\mathrm{P}}(X_1) = L_{\mathrm{P}}(X_2)$, the probability distributions of the classical information obtained by the server during the protocol P for each input are identical.
2. For inputs such that the probability distributions of the classical information obtained by the server during the execution of protocol P are identical, the quantum states obtained by the server are indistinguishable.

2.3.1 Childs Protocol

First, we introduce the earliest blind quantum computation protocol proposed by Childs [54].

Definition 2.16 (Childs Protocol)**.** A user has the following ability:

- Sufficiently large quantum memory.
- The ability to prepare the state $|0\rangle$.
- The ability to execute X and Z gates.

The user delegates his calculation to the server in the following procedure:

Step 1. The user selects one gate randomly from a gate set $\{H, T, CNOT\}$.

Step 2. If the user selected the gate to be executed, the user prepares a genuine input state for the gate. Otherwise, the user prepares a dummy state.

Step 3. The user encrypts the selected quantum state using a quantum one-time pad.

Step 4. The user sends the encrypted quantum state to the server.

Step 5. The user instructs the server to execute the chosen gate to the quantum state.

Step 6. The server executes the gate and then sends the output quantum state back to the user.

Repeat the above procedure until all gates in the quantum circuit the user wants to calculate are completed. However, when the T gate is selected, the following additional procedures are required after Step 6:

Step 1. If the original quantum one-time pad encryption key k_x is 1, then the user chooses the quantum state that has just returned; otherwise, selects a dummy qubit.

Step 2. The user encrypts the selected quantum state using a quantum one-time pad.

Step 3. The user sends the quantum state to the server.

Step 4. The user instructs the server to execute the S gate to the quantum state.

Step 5. The server executes the gate and then sends the quantum state back to the user.

This procedure is necessary because the T gate is non-commutative with the X gate.

Theorem 2.17. Childs Protocol is a blind quantum computation protocol.

Proof. Since all the quantum states received by the server are encrypted with a quantum one-time pad, the server cannot obtain information from the quantum states. When a quantum gate is executed to a quantum state encrypted with a quantum one-time pad, the encryption key changes as follows:

$$H(X^{k_x} Z^{k_z} |\psi\rangle) = X^{k_z} Z^{k_x} H |\psi\rangle , \tag{2.124}$$

$$S(X^{k_x} Z^{k_z} |\psi\rangle) = X^{k_x} Z^{k_x \oplus k_x} S |\psi\rangle , \tag{2.125}$$

$$T(X^{k_x} Z^{k_z} |\psi\rangle) = X^{k_x} Z^{k_x \oplus k_x} S^{k_x} T |\psi\rangle , \tag{2.126}$$

$$\begin{aligned} &CNOT(X^{k_x^1} Z^{k_z^1} |\psi\rangle_1 \otimes X^{k_x^2} Z^{k_z^2} |\psi\rangle_2) \\ &\quad = (X^{k_x^1} Z^{k_z^1 \oplus k_z^2} \otimes X^{k_x^1 \oplus k_x^2} Z^{k_z^2}) CNOT(|\psi\rangle_1 \otimes |\psi\rangle_2), \end{aligned} \tag{2.127}$$

where k_x, k_z, k_x^1, k_z^1, k_x^2 and k_z^2 are encryption keys for the quantum one-time pad.

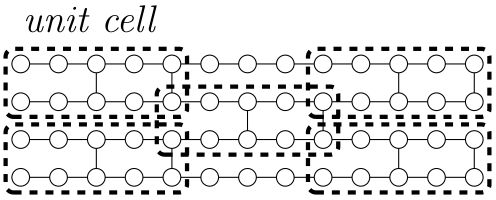


Figure 2.21: The brickwork state

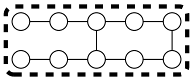


Figure 2.22: The unit cell in the brickwork state

It is easy to see that one can obtain a wanted state by employing a proper decryption key when the H gate, S gate, or $CNOT$ gate is executed. For instance, by operating $Z^{k_x \oplus k_x} X^{k_x}$ on the right-hand side of Equatuin 2.125, one obtains $S\left|\psi\right\rangle$. On the other hand, when the T gate is executed, if the encryption key is $k_x = 1$, the S gate is additionally executed to the quantum state. Since the user cannot execute the $S/S^{\dagger}$ gate, they need to cancel this additional gate. Hence, the user needs to perform the additional procedures. The user can execute quantum gates on the encrypted quantum state.

The user randomly decides which quantum gate to execute, so the server cannot obtain information about the quantum circuit from the quantum gate. Since the server cannot obtain information other than the size of the user's quantum circuit from the protocol, Childs protocol is a blind quantum computation protocol. □

2.3.2 BFK Protocol

For Childs protocol, it is necessary to have a quantum memory of the size required for the user's quantum computation. The BFK protocol, based on measurement-based quantum computation, was proposed to overcome this drawback [56].

First, let us define the *brickwork state* used in the BFK protocol.

Definition 2.18 (Brickwork state)**.** The graph state depicted in Figure 2.21 is defined as the brickwork state. Specifically, the brickwork state is composed of small elements called *unit cells*, as shown in Figure 2.22, alternatingly.

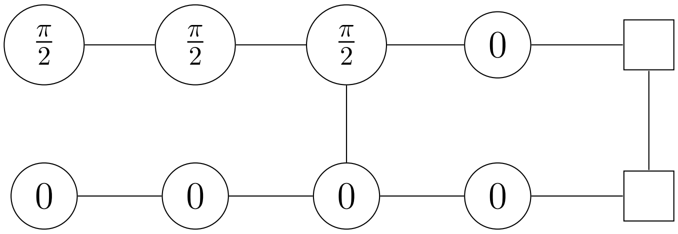


Figure 2.23: The $H \otimes I$ gate using a unit cell

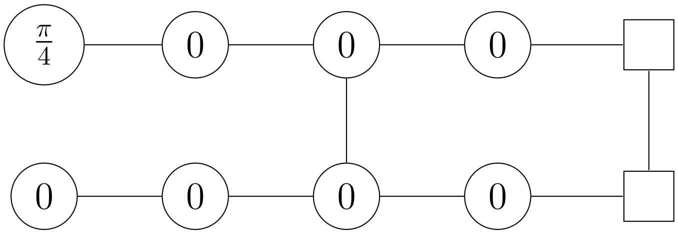


Figure 2.24: The $T \otimes I$ gate using the unit cell

By selecting the appropriate measurement angles, the unit cell can reproduce various quantum gates. For example, as shown in Figure 2.23, it is possible to reproduce the H gate (precisely, $H \otimes I$ gate). Similarly, by measuring at the angles depicted in Figures 2.24–2.26, the T gate, identity gate, and $CNOT$ gate can be executed.

Finally, we introduce the BFK protocol.

Definition 2.19 (BFK Protocol)**.** A user has the following ability:

- The ability to prepare the state $R_z(\theta)\left|+\right\rangle$, where θ can take values from the set $\{0, \frac{\pi}{4}, \frac{2\pi}{4}, \dots, \frac{7\pi}{4}\}$.

First, the user instructs the server to create the brickwork state using the following procedure:

Step 1. The user randomly selects an angle θ from the set $\{0, \frac{\pi}{4}, \frac{2\pi}{4}, \dots, \frac{7\pi}{4}\}$.

Step 2. The user creates a state $R_z(\theta)\left|+\right\rangle$ corresponding to the angle θ.

Step 3. The user sends that state to the server.

Step 4. The user repeats steps 1 through 3 until they send the required number of quantum states to the server.

Step 5. The user instructs the server to execute the CZ gates so that the sent states become the desired graph state.

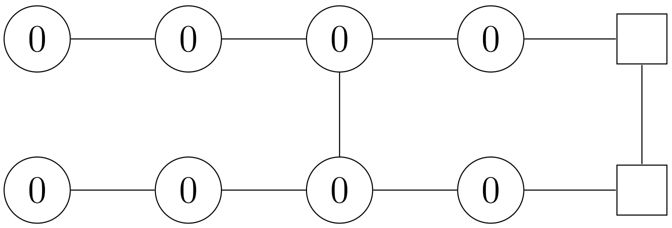


Figure 2.25: The identity gate using a unit cell

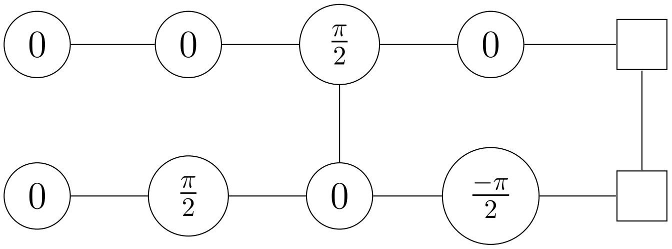


Figure 2.26: The *CNOT* gate using a unit cell

In the following, the angle of the graph state in the i-th column and j-th row is denoted as θ_{ij}.

Then, the user instructs the server to the brickwork state using the following procedure:

Step 1. The user determines the measurement angles ϕ_{ij} for each unit cell of the graph state so that it corresponds to the desired quantum gate.

Step 2. The user sends the measurement angle $\phi'_{ij} = \phi_{ij} - \theta_{ij} + r_{ij}\pi$ to the server, with $r_{ij} \in \{0, 1\}$ chosen randomly.

Step 3. The server performs the measurement $\{M_0^{ij} \equiv (|0\rangle + e^{i\phi'_{ij}} |1\rangle)(\langle 0| + e^{-i\phi'_{ij}} \langle 1|), M_1^{ij} \equiv (|0\rangle - e^{i\phi'_{ij}} |1\rangle)(\langle 0| - e^{-i\phi'_{ij}} \langle 1|)\}$ corresponding to the angle on the qubit in the i-th row and j-th column.

Step 4. The server sends the measurement output to the user.

Step 5. The user adjusts the measurement angle ϕ_{ij+1} based on the measurement output.

Step 6. Repeat steps 1 through 5 until all quantum gates have been executed.

Theorem 2.20. The BFK Protocol is a blind quantum computation protocol.

Proof. Since the measurement angle ϕ_{ij} is independent of the rotation gate's angle θ_{ij}, it is not possible to infer ϕ_{ij} from ϕ'_{ij}. Since the server is unaware of the actual measurement angles ϕ_{ij}, it cannot discern whether each unit cell is performing an H, T, or $CNOT$ gate. Therefore, the server cannot obtain information about the quantum circuit.

Because the measurement outputs are masked by r_{ij}, the server cannot determine whether the desired output is 0 or 1. In measurement-based quantum computation, remember that byproduct operators act on the quantum state during measurement. In other words, for the server, which does not know the desired measurement output, the byproduct operator is analogous to the encryption using an X gate and a Z gate in a quantum one-time pad. Hence, the server cannot obtain information about the quantum state.

Since the server cannot obtain information other than the size of the user's quantum circuit from the protocol, the BFK protocol is a blind quantum computation protocol. □

2.3.3 MF Protocol

The MF protocol is a blind quantum computation protocol where the measurement is performed by the user rather than the server [59]. If the user has the ability to measure instead of the ability to prepare quantum states, they would opt for the MF protocol.

Definition 2.21 (MF Protocol)**.** A user has the following ability:

- The ability to perform a specific measurement $\{P_0^\theta, P_1^\theta\}$.

These operators are given by:

$$P_0^\theta \equiv (|0\rangle + e^{i\theta}|1\rangle)(\langle 0| + e^{-i\theta}\langle 1|), \tag{2.128}$$

$$P_1^\theta \equiv (|0\rangle - e^{i\theta}|1\rangle)(\langle 0| - e^{-i\theta}\langle 1|), \tag{2.129}$$

where θ can take values from the set $\{0, \frac{\pi}{4}, \frac{2\pi}{4}, \dots, \frac{7\pi}{4}\}$.

The user instructs the server to the following procedure:

Step 1. The server creates a brickwork state of the size necessary for the user's calculation.

Step 2. The user determines the measurement angles ϕ_{ij} for each unit cell of the graph state so that it corresponds to the desired quantum gate.

Step 3. The server sends the qubit at the i-th row and j-th column to the user.

Step 4. The user performs the measurement $\{P_0^{\theta_{ij}}, P_1^{\theta_{ij}}\}$ on the received quantum state.

Step 5. The user adjusts the measurement angle ϕ_{ij+1} based on the measurement output.

Step 6. Repeat steps 1 through 5 until all quantum gates have been executed.

Theorem 2.22. The MF Protocol is a blind quantum computation protocol.

Proof. It is evident from the No-Signaling principle that the MF Protocol is a blind quantum computation protocol [89, 90]. □

2.3.4 Multi-server Protocol

Let us introduce a blind quantum computation protocol using multiple quantum servers [62]. Reichardt and colleagues' blind quantum computation protocol emerged from discussions on computational complexity.

First, we introduce the concept of *interactive proof systems* in computational complexity theory. An interactive proof system is a framework in which a server with unbounded computational power and a user with a classical computer engage in a dialogue to *verify* whether the solution presented by the server is indeed correct.

Definition 2.23 (IP [91])**.** A probabilistic polynomial-time verifier(user) sends messages back and forth with an unbounded prover(server). They can have polynomially many rounds of interaction. A language L belongs to IP if it satisfies the following conditions:

Completeness. If the answer is "yes"($w \in L$), the prover must be able to behave that the verifier accepts w with a probability of at least 2/3.

Soundness. If the answer is "no"($w \notin L$), regardless of the prover's behavior, the verifier must reject w with a probability of at least 2/3.

We can extend this system to consider a setup where multiple servers act as the provers.

Definition 2.24 (MIP [92])**.** A probabilistic polynomial-time verifier sends messages back and forth with some unbounded provers. The provers cannot communicate with each other. They can have polynomially many rounds of interaction. A language L belongs to MIP if it satisfies the following conditions:

Completeness. If the answer is "yes"($w \in L$), the provers must be able to behave that the verifier accepts w with a probability of at least 2/3.

Soundness. If the answer is "no"($w \notin L$), regardless of the provers' behavior, the verifier must reject w with a probability of at least 2/3.

Furthermore, we can also consider the case where the servers share entanglement.

Definition 2.25 (MIP* [93])**.** A probabilistic polynomial-time verifier sends messages back and forth with some unbounded provers. The provers cannot communicate with each other, but they share an arbitrary number of maximally entangled states. They can have polynomially many rounds of interaction. A language L belongs to MIP* if it satisfies the following conditions:

Completeness. If the answer is "yes"($w \in L$), the provers must be able to behave that the verifier accepts w with a probability of at least 2/3.

Soundness. If the answer is "no"($w \notin L$), regardless of the provers' behavior, the verifier must reject w with a probability of at least 2/3.

From here on, we denote MIP* with k servers as MIP*[k servers].

It is also possible to extend the model such that instead of the servers sharing entanglement, the user has a quantum computer.

Definition 2.26 (QMIP [94])**.** A *quantum* polynomial-time verifier sends messages back and forth with some unbounded provers. The provers cannot communicate with each other. They can have polynomially many rounds of interaction. A language L belongs to QMIP if it satisfies the following conditions:

Completeness. If the answer is “yes”($w \in L$), the provers must be able to behave that the verifier accepts w with a probability of at least $2/3$.

Soundness. If the answer is “no”($w \notin L$), regardless of the provers’ behavior, the verifier must reject w with a probability of at least $2/3$.

From here on, we denote QMIP with k servers as QMIP[k servers]. In fact, it is known that MIP* and QMIP are equivalent [62].

Theorem 2.27 (MIP* = QMIP [62])**.**

$$\mathrm{MIP}^* = \mathrm{QMIP}.$$

As a lemma in the proof of this equivalence, a two-server blind quantum computation protocol was proposed.

Lemma 2.28 (Two Server Blind Quantum Computation Protocol [62])**.**

$$\mathrm{MIP}^*[2 \text{ servers}] \geq \mathrm{QMIP}[0 \text{ server}] = \mathrm{BQP}.$$

Definition 2.29 (Two-server protocol(Informal))**.** A user has the following ability:

- The ability of classical computation.

Server A and Server B have the following conditions:

- The servers share an arbitrary number of maximally entangled states.
- The servers are prohibited from classical communication with each other.

We introduce the specific configuration of the two servers protocol. Two servers that share an entanglement do not necessarily follow a user’s instructions. Therefore, the user must verify that the server is performing the calculations correctly by delegating four circuit patterns to the server in Figure 2.27. First, the user creates a circuit for the calculation they wish to perform. A proper decomposition of the circuit makes each server impossible to obtain any information on the whole circuit unless the servers communicate with each other. The user performs the computation by repeatedly playing the following four *games*:

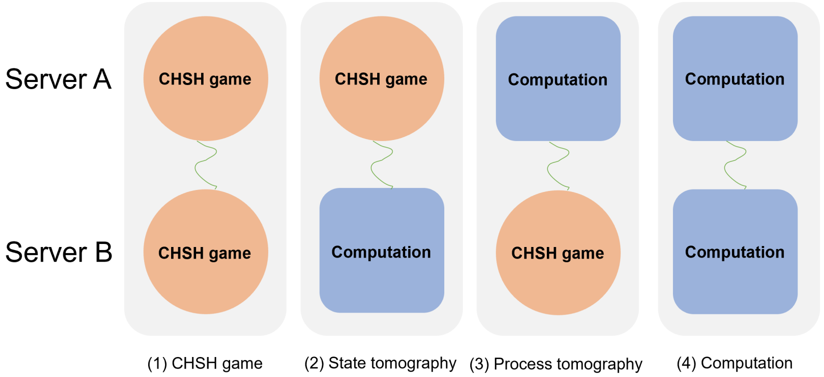


Figure 2.27: **Overview of the two server blind protocol.** (1) (2) (3)In these games, if the server's behavior deviates from the circuit as delegated by a user, the user obtains a wrong probability distribution as a measurement output and can notice the server's illegal behavior. (4)The servers should run the correct circuit for the calculation because they cannot distinguish games.

(1)**CHSH game.** Two servers perform the *CHSH game*, which is a game that verifies whether the measurement outputs satisfy the CHSH(Bell) inequality [65, 66, 95]. The user rejects and aborts the calculation unless both servers perform the correct measurements.

(2)**State tomography.** In the *state tomography*, Server B performs measurements for quantum computation, while Server A conducts measurements similar to those in the CHSH game. Server A must do the correct measurements because it cannot distinguish between the CHSH game and state tomography. Server B runs the correct circuit for computation, or else server A's measurement outputs will be incorrect.

(3)**Process tomography.** In the *process tomography*, Server A executes the quantum gate for quantum computation, while Server B conducts measurements similar to those in the CHSH game. Server B must do the correct measurements because it cannot distinguish between the CHSH game and process tomography. Server A runs the correct circuit for computation, or else server B's measurement outcomes will be incorrect.

(4)**Computation.** Server A cannot distinguish between state tomography and computation, and server B cannot distinguish between process tomography and computation, so the two servers should execute the correct circuit for computation.

In this way, the user can delegate any calculation to the two servers.

Theorem 2.30. The two-server protocol is a blind quantum computation protocol.

Proof. It is evident from Lemma 2.28 that the two-server protocol is a blind quantum computation protocol. □

2.3.5 Limitation of Single-server Blind Quantum Computation Protocol

We have introduced single-server blind quantum computation protocols for quantum users. Then, might there not be a single-server blind protocol for classical users? In fact, it is known that a blind quantum computation protocol for classical users might not exist [60, 61]. In this subsection, let us introduce those results.

The class of problems that quantum computers can efficiently solve is referred to as *BQP*.

Definition 2.31 (BQP)**.** A language L belongs to BQP if and only if there exists a polynomial-time uniform family of quantum circuits $\{Q_n \colon n \in \mathbb{N}\}$, such that

Completeness. for all x in L, $\Pr(Q_{|x|}(x) = 1) \geq \frac{2}{3}$,

Soundness. for all x not in L, $\Pr(Q_{|x|}(x) = 0) \geq \frac{2}{3}$,

where for all $n \in \mathbb{N}$, Q_n takes n qubits as input and outputs 1 bit.

The class of problems for which solutions can be verified in polynomial time on classical computers is referred to as *NP*.

Definition 2.32 (NP)**.** A language L belongs to NP if and only if and only if there exist polynomials p and q, and a deterministic Turing machine TM, such that

Completeness. for all x in L, there exists a string y of length $q(|x|)$ such that $TM(x, y) = 1$,

Soundness. for all x not in L and all strings y of length $q(|x|)$, $TM(x, y) = 0$, where for all x and y, the machine TM runs in time $p(|x|)$ on input (x, y).

The following results are known regarding a one-round blind quantum computation protocol for classical users.

Theorem 2.33 (One-round single-server blind quantum computation protocol for a classical user [61])**.** If a one-round single-server blind quantum computation protocol for classical users exists, then BQP $\subseteq$ NP.

However, at the same time, it is believed that BQP may not be a subset of NP [96].

Theorem 2.34. For any $T(n)$ which is $o(2^{n/2})$ relative to a random oracle with probability 1, BQTime($T(n)$) does not contain NP. Here, BQTime($T(n)$) denotes the class of problems solvable by a quantum computer operating in $T(n)$ time.

Because this result is derived using a random oracle, it does not necessarily imply that BQP is not a subset of NP. However, many in the field "believe" that BQP and NP are not subsets of one another.

Additionally, the following result is also known [60].

Definition 2.35 (MA/$\mathcal{O}(n^d)$)**.** A language L belongs to MA/n^d if and only if there exists polynomials p and q, a sequence $\{\alpha_n\}_{n\in\mathbb{N}}$ of strings with $\alpha_n \in \{0,1\}^d$, and a polynomial-time deterministic Turing machine TM, such that

Completeness. for all x in L, there exists a string y of length $q(|x|)$ such that $\Pr(TM(x, y, \alpha_n) = 1) \geq \frac{2}{3}$,

Soundness. for all x not in L and all strings y of length $q(|x|)$, $\Pr(TM(x, \alpha_n) = 0) \geq \frac{2}{3}$,

where for all x, y α_n, the machine TM runs in time $p(|x|)$ on input (x, y, α_n), and for some fixed constant d.

MA/$\mathcal{O}(n^d)$ can be understood as MA where a Turing machine may change for each bit size. If, hypothetically, a classical user could delegate quantum computation to a single server with unbounded computational capacity using a blind protocol, then BQP $\subseteq$ MA/$\mathcal{O}(n^d)$.

Theorem 2.36 (Unbounded single-server blind quantum computation protocol for a classical user)**.** If an unbounded single-server blind quantum computation protocol for classical users exists, then BQP $\subseteq$ MA/$\mathcal{O}(n^d)$.

However, a separation is known using an oracle [60].

Theorem 2.37. For each $d \in N$, there exists an oracle O_d such that BQP^{O_d} is not contained in (MA/$\mathcal{O}(n^d)$)^{O_d}.

Also, because this result is derived using an oracle, it does not necessarily imply that BQP is not a subset of MA/$\mathcal{O}(n^d)$.

Chapter 3

Extension of Single-server Blind Quantum Computation

This chapter is based on my paper [3].

3.1 Motivation and Our Work

When the public uses a universal quantum computer, it is assumed that it will be used as a quantum cloud server that exists in a few bases because the quantum computer is expensive. In this quantum cloud server, privacy will be an essential issue. Thus, a blind quantum computation protocol is needed so that each user can use the server without revealing the details of his or her calculations [54–56, 59, 97–100].

First, the blind protocol was proposed based on a quantum one-time pad [54]. Similar to a classical one-time pad [74], the quantum one-time pad uses the encryption key only once, therefore, the server cannot learn anything about the user's quantum state. However, this protocol needs multiple two-way quantum communications. In addition, the user is required to have a quantum memory on which a $SWAP$ gate can be executed. Then, Another protocol was proposed that requires neither quantum gates and two-way quantum communication nor quantum memory and $SWAP$ gates during its computation [97]. However, in this protocol while the input and output are encrypted, the calculation process is revealed to the server. This is a crucial drawback because an algorithm itself can constitute important information that should be kept secret. The blind protocols that do not use the quantum one-time pad have also been proposed [56, 59, 98].

In this chapter, we propose a new blind quantum protocol using rotation gates in addition to the quantum one-time pad. The abilities of the user required to perform the protocol are the ability to prepare the states $|+\rangle$, to execute T gates, to generate random numbers, and use a classical computer. These abilities of the user are equal to the abilities of a user in the previous protocols. In our protocol, the server does not need to execute non-clifford gates or do equivalent measurements. Hence, the capability required of the server in our protocol is equivalent to that of the MF protocol. We also compared the amount of qubits consumed in our protocol and the BFK protocol. We show that our protocol consumes fewer qubits than the BFK protocol except in special cases because our protocol uses fewer qubits per single-qubit gate. We also show that our protocol can be extended to fault-tolerant computation.

3.2 Technical Preliminaries

In this section, we describe the gate teleportation for an A_θ gate used to encrypt our protocol. For an overview of gate teleportation, see Subsection 2.2.3. For further details on the Quantum One-Time Pad, see Subsection 2.1.8.

3.2.1 Gate Teleportation

We explain the gate teleportation for an A_θ gate that is used for blindness in our protocol. The A_θ gate is defined by

$$A_\theta \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}. \tag{3.1}$$

When $\theta = \frac{\pi}{4}$, the A_θ gate is equivalent to the T gate.

For a given state $|\psi\rangle$, $A_\theta |\psi\rangle$ is obtained by using gate teleportation as shown in Figure 3.1, without directly executing the A_θ gate. Here a is the measurement output, and $|A_\theta\rangle$ is defined by

$$|A_\theta\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle). \tag{3.2}$$

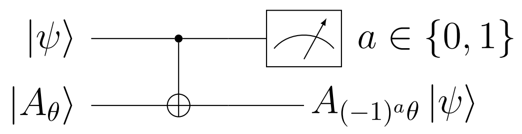


Figure 3.1: Executing A_θ gate by gate teleportation

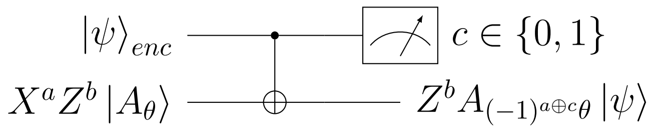

$|\psi\rangle_{enc}$
$X^a Z^b |A_\theta\rangle$
$c \in \{0,1\}$
$Z^b A_{(-1)^{a\oplus c}\theta} |\psi\rangle$

Figure 3.2: Key change at the A_θ gate using gate teleportation

3.3 The A_θ Gate

In this section, we explain the encryption techniques for the A_θ gate and its universality. First, we extend the quantum one-time pad to the A_θ gate. Next, we show how to modify the T gate and the A_θ gate in the quantum one-time pad. Finally, we explain universal quantum computation using the A_θ gate.

3.3.1 Quantum One-time Pad for the A_θ Gate

In this subsection, we show that the A_θ gate can be hidden using the quantum one-time pad. When $|A_\theta\rangle$ is encrypted using the quantum one-time pad, it is given by

$$|A_\theta\rangle_{\mathrm{enc}} = X^a Z^b |A_\theta\rangle . \tag{3.3}$$

When executing the gate teleportation by the state $|A_\theta\rangle_{\mathrm{enc}}$, the A_θ gate works as shown in Figure 3.2, since the Z gate commutes with the A_θ gate. Thus, it is possible to encrypt the A_θ gate using the quantum one-time pad. Even when the quantum state $|A_\theta\rangle$ is encrypted using only the Z gate, its ensemble is already a maximally mixed state. That is, such encryption using the X gate is not required. However, the X gate is used to hide the measurement output. It is used for the modification described in the following subsection.

3.3.2 T-like gate and single-qubit universal gate

It is known that an approximation of any single-qubit gate is achieved by using the T gate and the H gate [78, 81–83]. We represent gate blindness using non-parallel eight-axis rotation and a T-like gate. The T-like gate is defined as follows:

$$T = A_{\frac{i\pi}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, \tag{3.4}$$

$$T^3 = A_{\frac{i3\pi}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i3\pi}{4}} \end{bmatrix}, \tag{3.5}$$

$$T^\dagger = A_{\frac{-i\pi}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{bmatrix}, \tag{3.6}$$

$$(T^3)^\dagger = A_{\frac{-i3\pi}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i3\pi}{4}} \end{bmatrix}. \tag{3.7}$$

By combining the T-like gate with the H gate, it is possible to make rotations at eight axes that are not parallel. Table 3.1 shows these eight axes along with the gate combinations. In particular, note that the rotation axis of $T^\dagger H T^\dagger H$ is parallel to the rotation axis of $HTHT$. It is known that an arbitrary single-qubit gate can be approximated by the combination of $THTH$ and $HTHT$. The rotations of these two axes are not parallel. That is, they can achieve any rotation for the quantum state that the user desires. This is known thus a universal gate set for a single-qubit gate [78, 82, 83]. In the same way, any single-qubit gate can be approximated by the gate group shown in Table 3.1. Therefore, the server cannot discover which gate combination was chosen because he cannot know the received state $|A_\theta\rangle$. That is, the user can realize any single-qubit gate without it being known to the server.

3.4 Main Protocol

In this section, we describe our blind quantum computation protocol.

We assume a user has the ability to prepare a state $|A_\theta\rangle$ such that $\theta = \frac{n\pi}{4}(n = \{0, 1, \ldots, 7\})$, to execute the X and Z gates and to perform classical calculations. This user ability can be replaced by the ability to prepare the $|+\rangle$ and to execute the T gates. Also, we assume a server has a universal quantum computer. Note that in our protocol, the server is not required to have the capability to execute non-Clifford gates.

Table 3.1: Axes of rotation and the corresponding gate combinations

Gate	Axis of rotation
$THTH$	$(\cos\frac{\pi}{8},\sin\frac{\pi}{8},\cos\frac{\pi}{8})$
$THT^{\dagger}H$	$(-\cos\frac{\pi}{8},-\sin\frac{\pi}{8},\cos\frac{\pi}{8})$
$T^{\dagger}HTH$	$(\cos\frac{\pi}{8},-\sin\frac{\pi}{8},-\cos\frac{\pi}{8})$
$T^{\dagger}HT^{\dagger}H$	$(-\cos\frac{\pi}{8},\sin\frac{\pi}{8},-\cos\frac{\pi}{8})$
T^3HT^3H	$(\cos\frac{3\pi}{8},\sin\frac{3\pi}{8},\cos\frac{3\pi}{8})$
$T^3H(T^3)^{\dagger}H$	$(-\cos\frac{3\pi}{8},-\sin\frac{3\pi}{8},\cos\frac{3\pi}{8})$
$(T^3)^{\dagger}HT^3H$	$(\cos\frac{3\pi}{8},-\sin\frac{3\pi}{8},-\cos\frac{3\pi}{8})$
$(T^3)^{\dagger}H(T^3)^{\dagger}H$	$(-\cos\frac{3\pi}{8},\sin\frac{3\pi}{8},-\cos\frac{3\pi}{8})$

3.4.1 Circuit-based Protocol

We propose a circuit-based protocol, which we refer to as *Protocol 1*, which can also hide the position of the $CNOT$ gate by using a circuit based on the brickwork states proposed in the BFK protocol.

We define a *circuit like brickwork states* as follows.

Definition 3.1 (Circuit like brickwork states)**.** Circuit like brickwork states consist of a fixed number n of gates vertically and a fixed number $p(n)$, proportional to n, of gates horizontally. We can always make n and $p(n)$ an even number by adding ancilla gates. The circuit is created in the following steps:

Step 1. Each row starts with arbitrary unitary operator V which consist of m gates.

Step 2. Then, a unitary operator U_1 consisting of four gates is executed on the l-th row, where l is $\{l = 2k+1 | k = 0, 1\ldots, \frac{n}{2}-1\}$, and a unitary operator U_2 consisting of four gates is executed on $l+1$ th row.

Step 3. The user performs the CZ gate between the l-th row and the $l+1$ th row.

Step 4. A unitary operator U_3 consisting of four gates is executed on the l-th row, and a unitary operator U_4 consisting of four gates is executed on $l+1$ th row.

Step 5. The user performs arbitrary unitary operator V which consist of m gates on each row.

Step 6. The unitary operator U_2 consisting of four gates is executed on the l-th row, and the unitary operator U_1 consisting of four gates is executed on $l-1$ th row. Alice does not execute anything in the first and last row between steps 6 to 8.

Step 7. The user performs a CZ gate between the l-th row and the $l-1$ th row.

Step 8. The unitary operator U_4 consisting of four gates is executed on the l-th row, and the unitary operator U_3 consisting of four gates is executed on $l-1$ th row.

Step 9. Repeat steps 1 to 8 until the last column is reached.

The circuit like brickwork states is shown in Figure 3.

The unitary operator $\{U_1, \ldots, U_4\}$ that exists in front of each CZ gate can be changed to the identity gate or the $CNOT$ gate by changing it as shown in Figs.3.4–3.5. If $U_1 = I, U_2 = I, U_3 = I, U_4 = I$, the two CZ gates and the four unitary operators act as the identity gate. If $U_1 = R_z(\frac{\pi}{2}), U_2 = R_x(\frac{\pi}{2}), U_3 = I, U_4 = R_x(\frac{-\pi}{2})$, the two CZ gates and the four unitary operators act as the $CNOT$ gate. Each unitary operator can be made with a combination of A_θ gates and H gates as shown below:

$$I = H \cdot I \cdot H \cdot I = H \cdot A_0 \cdot H \cdot A_0, \tag{3.8}$$

$$R_z(\frac{\pi}{2}) = H \cdot I \cdot H \cdot S = H \cdot A_0 \cdot H \cdot A_{\frac{i\pi}{2}}, \tag{3.9}$$

$$R_x(\frac{\pi}{2}) = H \cdot S \cdot H \cdot I = H \cdot A_{\frac{i\pi}{2}} \cdot H \cdot I, \tag{3.10}$$

$$R_x(\frac{-\pi}{2}) = H \cdot S^\dagger \cdot H \cdot I = H \cdot A_{\frac{-i\pi}{2}} \cdot H \cdot I. \tag{3.11}$$

The number m of gates constituting an arbitrary unitary operator V between CZ gates is fixed. The reason is that to be a blind protocol, m needs to be fixed for any circuit and should not determined for each circuit.

This circuit can simulate any circuit by adjusting V and $\{U_1, \ldots, U_4\}$. This is because the $CNOT$ gates can be realized with the CZ gates and $\{U_1, \ldots, U_4\}$, and any single-qubit gates between the $CNOT$ gates can be realized with V. In a position where the $CNOT$ gate are not needed, the CZ gates can be converted to the identity gate as

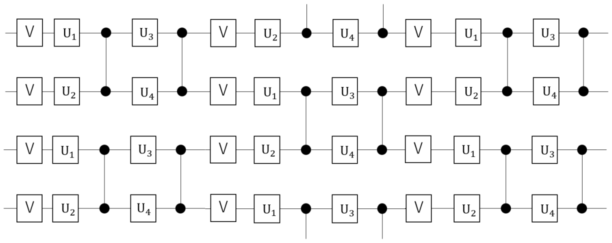

V U_1 U_3 V U_2 U_4 V U_1 U_3
V U_2 U_4 V U_1 U_3 V U_2 U_4
V U_1 U_3 V U_2 U_4 V U_1 U_3
V U_2 U_4 V U_1 U_3 V U_2 U_4

Figure 3.3: Circuit like brickwork states

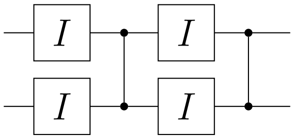


Figure 3.4: Combination of the CZ gates and single-qubit gates acting as the identity gate

in Figure 3.4. Therefore, instead of converting the user's circuit directly into the gate set $\{H, A_\theta, CNOT\}$, the user converts the user's circuit into the circuit like brickwork states and then converts the gate set $\{H, A_\theta, CNOT\}$ to hide position of the $CNOT$ gate. Also, the circuit like brickwork states by itself is not a blind protocol. However, since the server cannot obtain information concerning the single-qubit gates encrypted by the quantum one-time pad, he cannot determine whether the identity gate or the $CNOT$ gate in the circuit like brickwork states. When simulating the circuit like brickwork states, if the number of gates at V is less than m, the lack can be filled by reproducing the identity gates.

Definition 3.2 (Protocol 1)**.** A user has the following ability:

- The ability to prepare the state $R_z(\theta)\left|+\right\rangle$, where θ can take values from the set $\{0, \frac{\pi}{4}, \frac{2\pi}{4}, \ldots, \frac{7\pi}{4}\}$.

The user instructs the server to the following procedure:

Step 1. The user makes a quantum circuit for her calculation.

Step 2. The user reconstructs the original circuit to the circuit like brickwork states.

Step 3. The user decomposes the circuit by $\{H, A_\theta, CNOT\}$, where θ is limited to $\theta = \frac{n\pi}{4}(n = \{1, 3, 5, 7\})$.

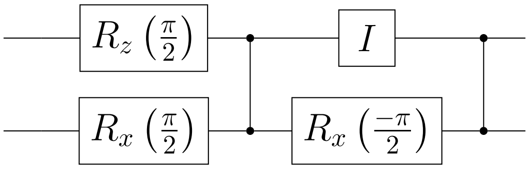

$R_z\left(\frac{\pi}{2}\right)$
I
$R_x\left(\frac{\pi}{2}\right)$
$R_x\left(\frac{-\pi}{2}\right)$

Figure 3.5: Combination of the CZ gates and single-qubit gates acting as the $CNOT$ gate

Step 4. The user prepares the qubits of the state $|A_\theta\rangle$ corresponding to the A_θ gate and the qubits needed to modify those gates and input states.

Step 5. The user generates the encryption keys and encrypts the qubits by the quantum one-time pad, then sends them to the server.

Step 6. After sending all the qubits, the user directs the server to do the calculations by classical communication. Specifically, when the user wants to execute H gate and CZ gate, the user has the server directly execute H gate and CZ gate. When the user wants to execute A_θ, she directs the server to execute A_θ by using gate teleportation of the $|A_\theta\rangle$. Then, the server sends the measurement output of the gate teleportation to the user and asks whether it is the desired result. If the result is not the desired one, the user directs the server to make additional modifications using $A_{2\theta}$ gate by gate teleportation and his Z gate.

Step 7. The server sends the last measurement to the user after the calculation is completed. When the last state is encrypted by X, the user flips the result and accepts it. When it is not, the user accepts the result.

The user can perform quantum computation while concealing the entire calculation process, including the position of the $CNOT$ gate.

Theorem 3.3. Protocol 1 is a blind quantum computation protocol.

Proof. The user encrypts the qubits by the quantum one-time pad, and the encryption key is selected randomly. That is, the server cannot obtain information about the qubits. The server performs the measurement for gate teleportation of A_θ gate, but the measurement output has a success probability of 1/2 regardless of the gate executed and the input quantum state. Therefore, the server does not obtain any information when executing the single-qubit gate. The $CNOT$ gate cannot be distinguished from the identity gate by using the circuit like brickwork states. That is, the user has the server execute the $CNOT$ gate without knowing where in the circuit it was executed. The server cannot learn anything about the output because the output qubits are still encrypted by the quantum one-time pad.

The server can only obtain information on the circuit size based on the calculation procedure. Therefore, Protocol 1 is a blind quantum computation protocol. □

Note that the size can be increased by sending dummy ancilla qubits.

The user's ability is preparing the $|+\rangle$ and executing the T gates. This ability is essentially equal to the ability required by the BFK protocol. The server's abilities are the ability to perform Clifford gates and to perform computational basis measurements. A quantum computer that has the ability to execute Clifford gates and to perform the computational basis measurements only can be efficiently simulated by a classical computer, as is known from the Gottesman-Knill theorem [85]. In the BFK protocol, besides the computational basis measurement, measurements corresponding to the T gate are required. In terms of computational capability, our protocol's server is inherently less capable than the server in the BFK protocol. The user can contract low-cost servers that have inferior abilities to our protocol.

3.4.2 Comparison of consumed qubits

An important previous study for this research is the BFK protocol [56]. Although the server's ability is different, the overall protocol is the same in that the user creates quantum states, the server uses those states to run circuits, and the user receives classical information. In this subsection, we will compare the number of qubits used in those protocols.

The BFK protocol consumed 8 qubits to execute any gate. However, since the BFK protocol can execute two single-qubit gates in one set of 8 qubits, we consider that the single-qubit gate consumes

Table 3.2: Number of qubits required for each gate

Protocol	H gate	T-like gate	$CNOT$ gate
BFK protocol	4	4	8
Protocol 1	0	1.5	12

4 qubits on average. Additional qubits are needed to execute the extra identity gates to form the Brickwork states. In Protocol 1, the number of qubits to perform a single-qubit gate varies depending on the measurement output. Since it takes an additional qubit to correct the result, it would take at least 1 qubit, up to 2 qubits, and an average of 1.5 qubits to execute the A_θ gate. And 8 A_θ gates are required to execute the $CNOT$ gate or the identity gate, therefore, 12 qubits are required to the $CNOT$ gate or the identity gate on average. Protocol 1 also needs to execute the additional identity gates to form circuit like brickwork states. We summarize the number of qubits consumed by each gate in Table 3.2.

The number of qubits consumed by the calculation depends on the number of these gates. The number of extra identity gates varies depending on the structure of the circuit the user wants to run, so the qubits consumed by those protocols cannot be simply compared. As a general case, we consider the case where an operator U, which can be approximated by n H gate and n A_θ gate, is executed between $CNOT$ gates. Since the BFK protocol requires 4 qubits for each of the H and A_θ gates,

$$N_{\mathrm{BFK}} = 2 \times 4 \times n + 8 = 8n + 8 \tag{3.12}$$

Then, let us consider $m = 4$ as the most inefficient case of Protocol 1. In this case, $HA_\theta HA_\theta$ is executed once between the CZ gates. In other words, one identity gate is needed for each two A_θ gates. It needs 12 qubits to convert the CZ gates into the identity gate on average, and the operator V between the CZ gates consumes 3 qubits to the two A_θ gates on average. Thus, it needs

$$N_{\mathrm{Protocol1}} = 1.5 \times n + 12 \times \frac{n}{2} = 7.5n. \tag{3.13}$$

Therefore, in the above case, the qubits consumed by Protocol 1 are $0.5n + 8$ qubits less than the qubits consumed by the BFK protocol. Protocol 1 can reduce the number of qubits it consumes by changing the value of m. In this case, we have only considered one row of quantum circuits. If we consider two or more rows, the qubits consumption per one row of the BFK protocol does not change, but the qubits consumption per one row of Protocol 1 decreases by 6 qubits per the identity gates. Hence, in general, Protocol 1 is superior to the BFK protocol in terms of the number of qubits it consumes. Note that Protocol 1 cannot directly execute H gates. Therefore, we estimate that the BFK protocol consumes fewer qubits for a circuit that executes a large number of alone H gates.

3.4.3 Fault tolerance

It is known that the ability to perform error correction in a universal quantum computer is an indispensable function since coherence is destroyed by external noise when manipulating a quantum state [78, 101–104]. It has also been shown that there is no universal gate set that is transversal (does not spread errors) [105, 106]. However, it is known that the H gates and the $CNOT$ gates can implement error correction codes in a transversal [107, 108]. For the T gate, this method is implemented only by transversal gates and gate teleportation. In this protocol, the gates used in the server's calculation are only the H gate and the $CNOT$ gate, and a non-transversal T-like gate can execute a logical T-like gate by preparing multiple similar A_θ state. Therefore, our protocol can be extended to fault-tolerant quantum computation.

3.5 Discussion

In this chapter, we proposed a blind quantum computation protocol using circuit-based quantum computation. In previous research, it has been discovered that the user's input and output can be concealed from the server using the quantum one-time pad. [97] However, previous techniques did not conceal the calculation process. In our protocol, blindness was satisfied using gate teleportation and expanding the T gate, which is important for universal quantum computation, to the A_θ gate. Protocol 1 uses the A_θ gates, circuit like brickwork states, and the quantum one-time pad. Protocol 1 requires the user to have the same abilities as the previous study and the server to have fewer abilities than the previous study [56]. In particular, the server does not require the ability to execute the non-clifford gates. We also have shown that Protocol 1 consumes fewer qubits than the protocols in the BFK protocol except in special cases. Additionally, we have shown that the method can be extended to fault-tolerant calculations by the same method for error correction using magic state.

Chapter 4

Extension of Multiple-server Blind Quantum Computation

This chapter is based on my paper [2].

4.1 Motivation and Our Work

Quantum computers have been actively studied with the expectation of providing higher computational capacity than classical computers. For example, Shor's algorithm uses the quantum Fourier transform to solve prime factorization and discrete logarithm problems exponentially faster than existing conventional algorithms [13]. Grover's algorithm is the fastest searching algorithm for an unordered database, at a speed that is thought to be impossible to achieve with classical computation [17]. In addition to specific algorithms, it is known that classical computers cannot sample as fast as quantum computers in the sampling problem [109, 110]. While quantum computers have such superiority, they will be enormously expensive compared with classical ones even if they become available in the future, as they will need some fine-tuned microscopic devices to use quantum effects. Consequently, it is anticipated that the server will possess the quantum computer, and users will delegate their quantum computations to the server.

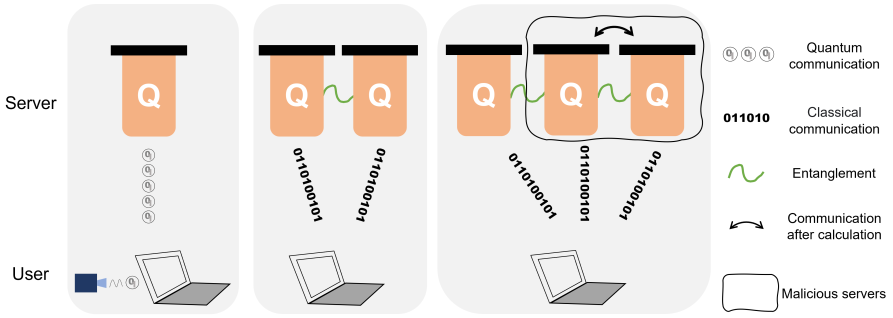
When quantum computers are used as cloud servers, user security is a concern. The user must send information about his/her calculations to the server to delegate the calculations. If the server is malicious, it may illegally obtain the user's information. Therefore, the user should use a blind quantum computation [3,54,56,59,62,63,98]. A blind quantum computation protocol securely encrypts inputs, outputs, and calculation processes of the calculations delegated by the user to the server. Blind quantum computation is expected to be an advantage of the new quantum computation because it is more powerful than fully homomorphic encryption, which is its classical analog [52].

At the moment, a blind quantum computation cannot be performed unconditionally; therefore, servers and users must be subjected to constraints. Blind quantum computation protocols applicable to a single quantum server have been most actively researched [3, 54, 56, 59, 98]. With the protocols of using the single server, the user does not have to impose any restrictions on the server, but the user must have quantum abilities. Currently, there is no known blind protocol that can be performed by the user having only abilities of classical computation and classical communication with a single server [111]. It is also not known about the possible or impossible existence of the blind protocol with a single server with a user who has only classical computation and classical communication abilities, but some negative results have been obtained [60, 61].

In contrast, blind protocols with two servers are available for users who can only perform classical computation and classical communication [62, 63]. These blind protocols are executed by a user who makes individual classical communication with multiple servers that share entanglement states. However, classical and quantum communication among servers is prohibited during and after the computation.

Given the widespread availability of classical computation and communication abilities, users are generally assumed to possess these capabilities. Therefore, these multiple-server blind quantum computation protocols are extremely convenient for users. However, the server is subjected to severe limitations; servers cannot communicate with each other in classical communication. In real, we cannot assume that servers cannot perform classical communication with each other, so we can assume that servers do not perform classical communication with each other according to the contract with the user.

There is no problem if the server honors the contract. However, we have required a blind protocol in case the server is malicious in



(a) Single server protocol **(b)** Two servers protocol with no server's communication **(c)** Multiple servers protocol with server's communication

Figure 4.1: **Server and user relationships for each protocol** (a)In a single-server protocol, a user must have the ability to generate or measure quantum states and classical computation. Additionally, the user and server must have quantum communication for exchanging quantum states. (b)In a two-server protocol, a user is only required to have the ability to perform classical computation and classical communication. Servers share entanglement, and communication between servers is always prohibited. (c)The relationship between servers and a user is the same as in the two-server protocols. In our protocol, some malicious servers can communicate with each other after calculation.

the first place. Additionally, servers that initially honor their contracts with users may suddenly breach those contracts. The current protocol does not allow users to estimate the magnitude of these risks.

In this chapter, we propose a blind protocol using multiple servers, in which some servers can classically communicate after the computation. First, we extend the protocol from two server cases discussed in previous studies. Next, we define a situation where some servers can do classical communication after a calculation. Finally, we then propose a method of encrypting the circuits used in the calculation so that the user can delegate the calculation by blind computation even if some servers are in classical communication after the calculation. Moreover, we show that if the user delegates his/her calculation to sufficiently many servers, the risk of their knowledge about the user's calculations can be controlled. An overview of the server-user relationship in previous studies' blind protocols and our target protocol is shown in Figure 4.1.

4.2 Multi-server Blind Quantum Computation Protocol with Limited Classical Communication among Servers

In this section, we propose a protocol that allows the user's calculations to remain blind even if some servers perform classical communication with other servers after the computation that the user delegates for servers. To do that, we first show that it is possible to perform the protocol on several servers, as extended from the two servers' protocol of the previous study [62]. Next, we show how to encrypt the circuit so that even if the server gets some information about the circuit, it cannot know anything about the calculation that the user delegated to it. For details on a blind quantum computation protocol, see Section 2.3; for the specifics of the two-server protocol, see Subsection 2.3.4.

4.2.1 Extension to Multiple Servers

We suggest a non-trivial method for increasing the number of servers by internally separating each of the protocol's two servers from the previous study [62]. One trivial way to increase the number of servers is to add virtual servers that do not participate in the calculation, but we will not consider this.

Theorem 4.1. In the two-server protocol (Definition 2.29), even when the roles of the two servers are each divided among multiple servers, the protocol remains blind.

Proof. We refer to the two servers used in the two-server protocol as server A and server B, respectively. We will split these servers into several groups. The set of servers that split server A is $\{A_a\}_a$, and the set of servers that split server B is $\{B_b\}_b$, where a and b are the numbers of server A and server B splits. Let $\{m_a^A\}_a$ and $\{m_b^B\}_b$ be the set of messages that a user sends to each server.

We use proof by contradiction. We assume that one of the servers can obtain information about the user's calculation using this server-splitting protocol. That is, we assume a protocol as described above to not be a blind quantum computation protocol. By assumption, one of the servers is getting information about the user's calculations from the messages $\{m_a^A\}_a$ and $\{m_b^B\}_b$. Let m_A and m_B be the sets of messages received by the original server A and server B. The server split is just a split of the internal workings of the original server, so $\{m_a^A\}_a$ can be created from m_A. The $\{m_b^B\}_b$ can be created in the same way. That is, the original server A and server B can easily simulate the servers' behavior after the split. Therefore, the original server A and server B can also obtain information about the user's calculation. However, this contradicts the fact that the protocol consisting of server A and server B is a blind quantum computation protocol.

Hence, the assumption is wrong, i.e., the protocol will remain a blind protocol even if the server is split such that the internal roles of server A and server B are split. □

Consequently, the two-server protocol has been successfully extended to a protocol employing a greater number of servers. As the number of servers used for computation grows, the circuits that use the computation become more fragmented, and each server knows less about the user's calculation. In reality, the amount of information each server knows is irrelevant if the servers do not use classical communication. However, each server must have a small amount of information if the servers perform classical communication with each other after the computation.

If some servers are allowed to communicate with other servers after the calculation, two servers can quickly learn the entire circuit if the protocol of the previous study [62]. However, if the number of servers participating in the calculation becomes huge, it becomes difficult to know everything completely. Of course, part of the circuit depends on the calculation, so the protocol is no longer a blind protocol under such an assumption. In the next subsection, we will encrypt the circuit so that servers can get some information about the circuit, but not the information that depends on the calculation of the user.

4.2.2 Main Protocol

In this subsection, we propose a blind quantum computation protocol that remains secure even if some servers communicate with each other after the computation. In the two servers, one server runs quantum gates on qubits, and another receives those qubits once and returns them to the first server. Thus, the entire quantum circuit for a calculation is realized by one of the two servers. From the information in the quantum circuit, the server can infer the input and output of the calculation and the calculation process. In other words, when the server is split up, if the server on the side running the quantum circuit shares information, information about the user's calculations will be leaked. Hence, we propose a method to encrypt the circuit so that even if the server knows some information about the circuit, the information does not depend on the user's calculation.

The set of gates required to perform an arbitrary quantum computation is called a universal gate set. In the two-server protocol, $\{CNOT, G\}$ is used as the universal gate set [112]. A G gate is defined by

$$G \equiv R_y\left(\frac{-\pi}{4}\right) = e^{i\frac{\pi}{8}Y}. \tag{4.1}$$

The G gate is an action that rotates $\pi/4$ radians around the Y axis. We adopt $\{H, T, CZ\}$ as a universal gate set for simplicity. These universal gate sets can approximate each other with polynomials, so the difference is not essentially significant.

In our new protocol, we utilize a *circuit like brickwork states* as employed in Protocol 1. For a detailed description of the circuit like brickwork states, see Subsection 3.1. Using the circuit like brickwork states, servers can know about the structure of the circuit but cannot get any information about the computation. Recall that the user can run any circuit by setting V and U_i in Figure 3.3 appropriately. It is known that any single qubit gate can be made from a combination of H gate and T gate [78]. Specifically, it approximates an arbitrary single-qubit gate by rotating of two axes on the Bloch ball, $HTHT$ and $HT^\dagger HT^\dagger$. Adding the identity gate $I = HIHI$ to these two sets allows the user to create any circuit by combining it with the circuit like brickwork states.

Next, we define *dummy gates* to eliminate information about the computation from these gate combinations.

Definition 4.2 (Dummy gates)**.** Let K be a constant. We define the elementary gates by:

$$D_1 = HT, \tag{4.2}$$

$$D_2 = HT^\dagger, \tag{4.3}$$

$$D_3 = HI. \tag{4.4}$$

A sequence of any K consecutive gates chosen from the set $\{D_1, D_2, D_3\}$ is defined as dummy gates.

As mentioned earlier, an arbitrary single-qubit gate consists of $HTHT$, $HT^\dagger HT^\dagger$, and $HIHI$. When a single-qubit gate consists of $K/2$ combinations of these three, the user adds the dummy gates consisting of K consecutive $\{D_1, D_2, D_3\}$. A server who knows nothing about the original single-qubit gate will not distinguish between those dummy gates and the original. By using those dummy gates, even if the server gets the information of $K/2$ consecutive gates from the user, the server cannot distinguish which gate is the original gate by running the dummy gates in parallel. By definition, dummy gates do not allow the server to obtain any information about the calculation, even if the server knows about a gate combination of less than $K/2$. For the encryption using dummy gates, the user needs to add, at most, 3^K gates. Since K is a constant, initially determined by the user, independent of the number of input bits, 3^K is also a constant. Therefore, adding dummy gates is efficient because it increases the number of gates by a constant factor independent of the number of input bits in the calculation.

Next, we introduce the procedure required to hide the output. The measurement required during the protocol is not dependent on the calculation, but the final measurement output, which is the output of the calculation, is dependent on the calculation. Therefore, we randomly perform an X gate or an identity gate at the end of each calculation. When the identity gate is executed to the quantum state, the user accepts the output directly; however, if the X gate is executed, the user accepts the output after flipping it. In this case, the probability of getting output either "0" or "1" as the server's measurement output is 50% each, and the actual output of the calculation cannot be known from the measurement output.

Then, we introduce a method to execute the X gate or the identity gate without revealing its implementation to the server. X axis rotation on the Bloch ball can be implemented as follows:

$$R_x(\frac{\pi}{4}) = H \cdot T \cdot H \cdot I. \tag{4.5}$$

If this $R_x(\frac{\pi}{4})$ is applied four times, it becomes the X gate, and if it is applied eight times, it becomes the identity gate. In other words, when the number of times $R_x(\frac{\pi}{4})$ is executed in a certain gate sequence is divided by 8, the remainder of 4 or 0 changes whether it is the X gate or the identity gate. Therefore, by having the server execute the $R_x(\frac{\pi}{4})$ gate multiple times at the end of the calculation, the user can conceal the output from servers that are unaware of the total number of $R_x(\frac{\pi}{4})$ gates executed.

Finally, we propose a protocol that summarizes the encryption of the circuit. In the following, we assume that the number of entire servers is $2N$ and that K servers ($N > \lfloor K/2 \rfloor$) do classical communication after the computation.

Definition 4.3 (Protocol 2)**.** The basic structure of the protocol is the same as the two-server protocol in Definition 2.29.

A user has the following ability:

- The ability of classical computation.

In this protocol, the two servers are divided into N servers each, i.e., the whole system will consist of $2N$ servers. The servers have the following conditions:

- The servers share an arbitrary number of maximally entangled states.
- The servers are prohibited from classical communication with each other.

We label each server as $\{A_1, A_2, \cdots, A_N\}$ and $\{B_1, B_2, \cdots, B_N\}$, then A_i receives quantum states from B_{i-1}, executes any gate, and passes quantum states to B_i. We also define that the server B_N sends a quantum state to server A_1. This protocol encrypts the circuit in the following process:

Step 1. The user restructures the circuit for the calculations by using the circuit like brickwork states.

Step 2. The user decomposes V and U_i, which compose the circuit like brickwork states, into $HTHT$, $HT^\dagger HT^\dagger$, and $HIHI$.

Step 3. The user adds dummy gates to the circuit so that the dummy gates are run in parallel for the gate sequence of $\lfloor K/2 \rfloor$ gates for the gates consisted in step 2.

Step 4. The user randomly executes the X gate or the identity gate using the gate sequence consisting of $4N$ gates that are $HTHI$ or $HIHI$ just before measuring the quantum state corresponding to the output.

The remaining delegation procedure is the same as the two-server protocol.

We show that Protocol 2 is a blind quantum computation protocol even when K servers of the $2N$ servers ($N > \lfloor K/2 \rfloor$) can do classical communication after the calculation.

Theorem 4.4. Protocol 2 is a blind protocol even when K servers of the $2N$ servers ($N > \lfloor K/2 \rfloor$) can do classical communication after the calculation.

Proof. Since we assume that all servers do not perform classical communication during the computation, from Theorems 2.30 and 4.1, the servers cannot get information about the user's calculation.

We consider what happens after the server finishes the computation delegated by the user and sends the output to the user. By assumption, the K servers can perform classical communication after the user's calculation is completed. It is shown from Theorems 2.30 and 4.1 that servers that do not perform classical communication with other servers after the computation is finished do not obtain information that depends on the user's calculations. Then, we describe the servers that do the classical communication with other servers. Since servers can do classical communication with each other, the protocol that users use to prevent servers from doing things differently from the user's instructions is no longer relevant to the server, and the server can directly get information about the user's circuit.

Even if the server can obtain information about the user's circuit, we show that it cannot get information that depends on the user's calculations. Since the circuit is built using the structure of the circuit like brickwork states, the server is unable to get information about the computation from the circuit structure. The server also gets information about the circuit's consecutive gates at most $\lfloor K/2 \rfloor$, but the circuit uses dummy gates parallel with the original gates. The server cannot distinguish between the dummy gates and the original gates because it does not know the original user's calculations. In other words, the gate information that the server obtained is all the possible gate combinations it could get. Hence, the server cannot obtain information that depends on the user's calculations from the gate information.

The input can be decomposed into $|0\rangle$ and gates without loss of generality; therefore, it can be hidden just like the gate. Since the measurement outputs during the computation process do not depend on the user's original calculation, the server cannot obtain information that depends on the user's calculation from them.

The server also gets some of the measurement outputs that correspond to the output of the calculation. However, recall that the user encrypted this output using the X gate or the identity gate randomly. The server cannot distinguish between the X gate and the identity gate without knowing all of the gates in the last $4N$ gate sequences. By the assumption the server knows only about $4\lfloor K/2 \rfloor$ gates out of $4N$, the server cannot know about the gates that encrypt the output. Therefore, the server's output is half "0" and half "1", and the server cannot decrypt it, so the output does not depend on the user's calculations.

The above result holds that even if the server performing the classical communication is less than K. The classical information available to the server does not depend on the user's original calculation.

The server's quantum state is identical to the two-server protocol. If those servers can get a quantum state that depends on the calculation, servers can also get the quantum at the time of the two servers. This contradicts Theorem 2.30. Therefore, the quantum state obtained by the server is independent of the user's original calculation.

Hence, Protocol 2 is a blind quantum computation protocol even if K servers of the $2N$ servers ($N > \lfloor K/2 \rfloor$) can do classical communication after the computation. □

4.2.3 Risk Estimation

Theorem 4.4 is based on the premise that after the computation, only K servers of the $2N$ servers ($N > \lfloor K/2 \rfloor$) perform classical communication after the computation. In reality, we can assume that a user has made a contract with all servers not to do classical communication with each other, but some of them have done so in violation of the contract after the computation. Assume that t is the average time between one server leaking information and the next, whether consciously or unconsciously. If a user chooses a sufficiently large K, the law of large numbers allows the user to estimate that the time it takes for the server to get the average user's information is $(K + 1)t$. Although t is considered fixed here, it is not necessarily fixed, and in practice, accurate model design for t is necessary. However, Protocol 2 allows the user to choose the parameter K, allowing him to adopt the risk of time leaking information that depends on the parameter K rather than the risk of time leaking information by a single server. Thus, users can estimate the risk that the previous study could not by using Protocol 2, and they can choose the number of servers according to the risk they are willing to accept.

4.3 Discussion

In this chapter, we proposed a blind quantum computation protocol with multiple servers that is effective even if some servers perform classical communication after the computation. Protocol 2 is an extension of the two-server protocol proposed in the previous study [62]. Firstly, we increased the number of servers by splitting the role into multiple servers inside the server for the two servers used in the two-server protocol. Next, we proposed a method of encrypting the circuit such that even if some circuit information is leaked, the server will not be able to determine which information is dependent on the user's original calculation. We then proposed Protocol 2, which summarized them and showed a blind protocol even in assumption. Finally, we have seen that the risk can be quantified to appropriately adjust the number of servers.

One of the disadvantages of our protocol is that it uses many more gates in the calculation than the two-server protocol. The amount it increases depends on how much risk the user is willing to accept. However, the increase in the number of gates fits into the polynomial size. Another disadvantage is that to use $2N$ servers, $2N$ quantum cloud servers sharing entanglements should exist in reality.

Chapter 5

Limitation of Blind Quantum Computation

This chapter is based on my paper [1].

5.1 Motivation and Our Work

Quantum computers are expected to become the next-generation computers because they can perform calculations that are considered impossible with classical computers. For example, Shor's algorithm [13] solves prime factorization problems in polynomial time using the quantum Fourier transform, and Grover's algorithm [17] is recognized as the quickest unordered database search. However, due to the sensitivity of quantum states to external noise, the physical implementation of quantum computers is hard and requires expensive technology. Hence, quantum computers will most likely be employed as cloud quantum servers rather than being owned by individual customers. In the context of such cloud quantum servers, a fundamental concern is a potential for the server to illegally obtain information about the computations delegated by the user. Therefore, users require a blind quantum computation protocol, allowing them to perform calculations without revealing their input, output, and processes [2,3,54–56,59,62,63,98,113].

To execute a blind quantum computation protocol, a user must encrypt the input, output, and processes of the delegated calculation. Childs showed that a user with quantum memory and the ability to execute an X gate and a Z gate could perform a blind quantum computation protocol via quantum communication with a quantum server [54]. Broadbent, Fitzsimons, and Kashefi proposed a blind quantum computation protocol in which users who do not have quantum memory create a specific quantum state, send it to a server, and do classical communication with the server [56]. Several more blind quantum computation protocols are also carried out by a user doing quantum communication with a single server [3, 59, 98, 113]. Protocols utilizing many servers have been proposed to ease the limitations on the user's abilities [2, 62, 63]. In these protocols, a user requires a classical computer and classical communication with multiple servers that share entangled qubits. These protocols are useful because the user does not need to have any quantum equipment. However, it is vital to note that classical and quantum communication is not allowed among multiple servers.

As mentioned above, thus far, several different blind quantum computation protocols have been proposed. However, it is uncertain if there is a single server protocol with users who only have classical capabilities and a multiple servers protocol that allows servers to freely communicate. The standard users are considered to have classical computation and communication abilities. In general, servers are considered to communicate freely with each other. Therefore, if they exist, these protocols would be the most user-friendly blind quantum computation protocols. Our goal is to investigate the link between these protocols.

In this chapter, we show that if there exists a single-server blind quantum computation (SBQC) protocol with users who have only classical abilities, there is a multi-server blind quantum computation (MBQC) protocol that allows servers to communicate with each other, and vice versa. We show specifically that if the SBQC protocol exists, it can be emulated with multiple servers, and that if the MBQC protocol exists, it can be simulated with a single server. We further show that these simulation approaches are not affected by the particular blind quantum computation protocol configuration. As a result, even investigating multi-server protocols can lead to the search for blind quantum computation protocols that employ a single server with users who only have classical abilities.

5.2 Blind Quantum Computation Protocol

In this section, we define an SBQC protocol with users who only have classical capabilities and an MBQC protocol that allows servers to freely communicate. For details on the definition of a blind quantum computation protocol, see Section 1.1. We assume in the following section that honest servers have the quantum computing power and malicious servers have unbounded computing power. Let n be the number of input bits.

5.2.1 SBQC Protocol

In this subsection, we define an SBQC protocol in which users can only use classical abilities.

Definition 5.1 (Single-server blind quantum computation protocol with a classical user)**.** A user has the following ability:

- The ability of classical computation.

If the delegating computation protocol, characterized by the following interactions between the user and the server, is a blind quantum computation protocol, we define it as a *single-server blind quantum computation protocol with a classical user*. The number of protocol steps is the polynomial-size $p(n)$.

Step 1. Send the first message to the server
The user sends a classical polynomial-size message m_1 to the server.

Step 2. Return a first message to the user
The server receives the user's message m_1 and performs quantum computation based on the message. The server sends to the user a classical polynomial-size message s_1, the content of which is determined by the server's calculation.

Step 3. Send the second message to the server
The user gets the message s_1 and performs classical computation based on the message. The user sends a classical polynomial-size message m_2 to the server, the size which relies of the user's

calculation.

⋮

Step $2i$. Return an i-th message to the user

The server receives the user's message m_i and performs quantum computation based on the message. The server sends a classical polynomial-size message s_i, which depends on the server's calculation, to the user.

Step $2i+1$. Send a $i+1$-th message to the server

The user receives the message s_i and performs classical computation based on the message. The user sends a classical polynomial-size message m_{i+1}, which depends on the user's calculation, to the server.

⋮

Step p(n). Calculation is complete

The user receives the last message s_l and obtains the result of the delegated calculation by decryption.

By this definition, an honest server has quantum computing power, so obviously, a user can delegate quantum computation to it.

5.2.2 MBQC Protocol

We define a multi-server blind quantum computation protocol that allows servers to communicate. We define separately when servers share entanglement with each other and when they do not.

Firstly, we define a multi-server blind quantum computation protocol that allows servers to communicate with each other during computation.

Definition 5.2 (Multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation)**.** A user has the following ability:

- The ability of classical computation.

The number of servers is polynomial-size $q(n)$. The servers have the following conditions:

- The servers do not share any maximally entangled states.
- The servers can do classical communication with each other always.

If the delegating computation protocol, characterized by the following interactions between the user and the servers, is a blind quantum computation protocol, we define it as a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation. The number of protocol steps is the polynomial-size $p(n)$.

Step 1. Send first messages to the servers
The user sends classical polynomial-size messages to all servers. Let $m_{1,j}$ be the message that the user sends to the j-th server.

Step 2. Return first messages to the user
The j-th server receives the user's message $m_{1,j}$ and performs quantum computation and classical communication with other servers based on the message. The j-th server sends a classical polynomial-size message $s_{1,j}$, which depends on the server's calculation, to the user.

Step 3. Send second messages to the server
The user gets the messages $\{s_{1,j}\}_j$ and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, which depends on the user's calculation. Let $m_{2,j}$ be the message that the user sends to the j-th server.

$\vdots$

Step $2i$. Return i-th messages to the user
The j-th server gets the user's message $m_{i,j}$ and performs quantum computation classical communication with other servers based on the message. The j-th server sends a classical polynomial-size message $s_{i,j}$, which depends on the server's calculation, to the user.

Step $2i+1$. Send $i+1$-th messages to the server
The user receives the messages $\{s_{i,j}\}_j$ and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, which depends on the

user's calculation. Let $m_{i+1,j}$ be the message that the user sends to the j-th server.

$\vdots$

Step p(n). Calculation is complete

The user receives the last messages $\{s_{l,j}\}_j$ from the servers and obtains a result about the delegated calculation by decryption.

Definition 5.3 (Multiple-servers with entanglement blind quantum computation protocol that allows servers to do quantum communicate with each other during computation)**.** A user has the following ability:

- The ability of classical computation.

The number of servers is polynomial-size $q(n)$. The servers have the following conditions:

- The servers share an arbitrary number of maximally entangled states.
- The servers can do classical communication with each other always.

When entanglement is sharing between servers in the server's computation step of the multiple-servers without entanglement blind quantum computation protocol that allows servers to communicate with each other during computation (Definition 5.2), it is defined as multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation.

These definitions just state that the blind protocols performed by the aforementioned processes, if they exist, will be referred to by the names provided in each definition, and they do not prove the existence of these protocols.

Regardless of the server's entanglement sharing, the user can delegate quantum computation to an honest server because the server has quantum computational abilities. In the protocol with shared entanglement, physical quantum communication is unnecessary since quantum teleportation can be performed using classical communication combined with entanglement.

Next, we define a multi-server blind quantum computation protocol that allows servers to communicate with each other after computation.

Definition 5.4 (Multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation)**.** A user has the following ability:

- The ability of classical computation.

The number of servers is polynomial-size $q(n)$. The servers have the following conditions:

- The servers do not share any maximally entangled states.
- The servers can do classical communication with each other after computation.

If the delegating computation protocol, characterized by the following interactions between the user and the servers, is a blind quantum computation protocol, we define it as a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation. The number of protocol steps is the polynomial-size $p(n)$.

Step 1. Send first messages to the servers
The user sends classical polynomial-size messages to all servers. Let $m_{1,j}$ be the message that the user sends to the j-th server.

Step 2. Return first messages to the user
The j-th server receives the user's message $m_{1,j}$ and performs quantum computation. The j-th server sends a classical polynomial-size message $s_{1,j}$, which depends on the server's calculation, to the user.

Step 3. Send second messages to the server
The user gets the messages $\{s_{1,j}\}_j$ and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, which depends on the user's calculation. Let $m_{2,j}$ be the message that the user sends to the j-th server.

$\vdots$

Step $2i$. Return i-th messages to the user

The j-th server gets the user's message $m_{i,j}$ and performs quantum computation. The j-th server sends a classical polynomial-size message $s_{i,j}$, which depends on the server's calculation, to the user.

Step $2i+1$. Send $i+1$-th messages to the server

The user receives the messages $\{s_{i,j}\}_j$ and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, which depends on the user's calculation. Let $m_{i+1,j}$ be the message that the user sends to the j-th server.

$\vdots$

Step p(n). Calculation is complete

The user receives the last messages $\{s_{l,j}\}_j$ from the servers and obtains a result about the delegated calculation by decryption.

Definition 5.5 (Multiple-servers with entanglement blind quantum computation protocol that allows servers to do quantum communicate with each other after computation)**.** A user has the following ability:

- The ability of classical computation.

The number of servers is polynomial-size $q(n)$. The servers have the following conditions:

- The servers share an arbitrary number of maximally entangled states.
- The servers can do classical communication with each other after computation.

When entanglement is sharing between servers in the multiple-servers without entanglement blind quantum computation protocol that allows servers to communicate with each other after computation (Definition 5.4), it is defined as multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation.

5.3 Equivalence of single server and multiple server blind quantum computation protocols

In this section, we show that if the SBQC protocol defined in the previous section exists, then there is the MBQC protocols defined in the previous section, and vice versa.

Theorem 5.6. If a single-server blind quantum computation protocol with a classical user exists, then a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation can be constructed from a single-server blind quantum computation protocol with a classical user. Furthermore, if a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation exists, then a single-server blind quantum computation protocol with a classical user can be constructed from a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation.

Proof. We first show that if there exists a single-server blind quantum computation protocol with a classical user, then there exists a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation. Assume there is a single-server blind quantum computation protocol with a classical user. The number of servers is polynomial-size $q(n)$. The user chooses one of those servers. This chosen server can be the first server without loss of generality. With the following procedure, we explore the scenario when a user delegates computation to multiple servers:

Step 1. Send first messages to the servers
The user sends classical polynomial-size messages to all servers. Let $m_{1,j}$ be the message that the user sends to the j-th server, and $m_{1,1}$ is m_1 and $j \neq 1$ message $m_{1,j}$ is a meaningless string.

Step 2. Return first messages to the user
The j-th server receives the user's message $m_{1,j}$ and performs

quantum computation and classical communication with other servers based on the message. The j-th server sends a classical polynomial-size message $s_{1,j}$, which depends on the server's calculation, to the user.

Step 3. Send second messages to the server

The user gets the message $s_{1,1}$, discards the messages from other servers, and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, which depends on the user's calculation. Let $m_{2,j}$ be the message that the user sends to the j-th server, and $m_{2,1}$ is m_2 and $j \neq 1$ message $m_{2,j}$ is a meaningless string.

$\vdots$

Step $2i$. Return i-th messages to the user

The j-th server receives the user's message $m_{i,j}$ and performs quantum computation classical communication with other servers based on the message. The j-th server sends a classical polynomial-size message $s_{i,j}$, which depends on the server's calculation, to the user.

Step $2i+1$. Send $i+1$-th messages to the server

The user receives the message $s_{i,1}$ and discards other server's messages, and performs classical computation based on the message. The user sends classical polynomial-size messages to all servers, the size of which depends on the user's calculation. Let $m_{i+1,j}$ be the message that the user sends to the j-th server, and $m_{i+1,1}$ is m_{i+1} and $j \neq 1$ message $m_{i+1,j}$ is a meaningless string.

$\vdots$

Step p(n). Calculation is complete

The user receives the last message $s_{l,1}$ from the first server and gets a result about the delegated calculation by decryption.

Note that m_i and s_i refer to messages in the single-server protocol.

This protocol delegates the computation to only one server out of multiple servers. The information gained by multiple servers during this protocol is the same as that obtained by a single server during the single-server blind quantum computation protocol with a classical user. If malicious servers can obtain information about the computation from this protocol, then the malicious server can also obtain information from the single-server protocol. This contradicts the assumption. Therefore, if there is a single-server blind quantum computation protocol with a classical user, there is a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation.

We then show that if there exists a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation, then there exists a single-server blind quantum computation protocol with a classical user. Assume there is a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation. We consider the scene where a user delegates computation to a single server using the following procedure:

Step 1. Send the first message to the server

The user sends a classical polynomial-size message $m_1 = \{m_{1,1}, \cdots, m_{1,q(n)}\}$ to the server.

Step 2. Return the first message to the user

The server receives the user's message m_1 and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_1 = \{s_{1,1}, \cdots, s_{1,q(n)}\}$, which depends on the server's calculation, to the user.

Step 3. Send a second message to the server

The user gets the message s_1 and performs classical computation based on the message. The user sends a classical polynomial-size message $m_2 = \{m_{2,1}, \cdots, m_{2,q(n)}\}$, which depends on the user's calculation, to the server.

$\vdots$

Step $2i$. Return a i-th message to the user

The server receives the user's message m_i and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_i = \{s_{i,1}, \cdots, s_{i,q(n)}\}$, which depends on the server's calculation, to the user.

Step $2i + 1$. Send a $i + 1$-th message to the server

The user receives the message s_i and performs classical computation based on the message. The user sends a classical polynomial-size message $m_{i+1} = \{m_{i+1,1}, \cdots, m_{i+1,q(n)}\}$, which depends on the user's calculation, to the server.

$\vdots$

Step p(n). Calculation is complete

The user receives the last message s_l from the server and gets a result about the delegated calculation by performing a classical calculation.

Note that $m_{i,j}$ and $s_{i,j}$ refer to messages in the multiple-servers protocol.

This protocol may be thought of as a single-server protocol simulating the multiple-servers protocol. If the server is honest, this simulation can be performed by a single server since it is just a quantum computation. Malicious servers can do classical communication during computation in the multiple-servers protocol. In other words, malicious servers might send all user messages to a single server and calculate them alone on that server against the user's intentions. Since the malicious server has unbounded computing power, there is no difference in computing power whether all calculations are alone on one server or multiple servers. From the given assumptions, the multiple-server blind quantum computation protocol satisfies security against such attacks by malicious servers. If the malicious single server can get calculation information from the aforementioned single-server protocol, then malicious servers can also get calculation information from the multiple-server protocol. This contradicts the assumption. Therefore, if a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation exists, so does a single-server blind quantum computation protocol with a classical user. □

Theorem 5.7. If a single-server blind quantum computation protocol with a classical user exists, then a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation can be constructed from a single-server blind quantum computation protocol with a classical user. Furthermore, if a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation exists, then a single-server blind quantum computation protocol with a classical user can be constructed from a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation.

Proof. The proof for the former follows an approach analogous to that presented in Theorem 5.6.

We show that if there exists a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation, then there exists a single-server blind quantum computation protocol with a classical user. Assume a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation. We consider the case where a user delegates computation to a single server using the protocol described below. The number of servers is polynomial-size $q(n)$.

Step 1. Send the first message to the server

The user sends a classical polynomial-size message $m_1 = \{m_{1,1}, \cdots, m_{1,q(n)}\}$ to the server.

Step 2. Return the first message to the user

The server receives the user's message m_1 and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_1 = \{s_{1,1}, \cdots, s_{1,q(n)}\}$, which depends on the server's calculation, to the user.

Step 3. Send a second message to the server

The user gets the message s_1 and performs classical computation based on the message. The user sends a classical polynomial-size message $m_2 = \{m_{2,1}, \cdots, m_{2,q(n)}\}$, which depends on the user's calculation, to the server.

⋮

Step $2i$. Return a i-th message to the user

The server receives the user's message m_i and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_i = \{s_{i,1}, \cdots, s_{i,q(n)}\}$, which depends on the server's calculation, to the user.

Step $2i+1$. Send a $i+1$-th message to the server

The user receives the message s_i and performs classical computation based on the message. The user sends a classical polynomial-size message $m_{i+1} = \{m_{i+1,1}, \cdots, m_{i+1,q(n)}\}$, which depends on the user's calculation, to the server.

$\vdots$

Step p(n). Calculation is complete

The user receives the last message s_l from the server and gets a result about the delegated calculation by decryption.

Note that $m_{i,j}$ and $s_{i,j}$ refer to messages in the multiple-servers protocol.

A single server can also easily prepare entanglement, making such protocols feasible. If the server is honest, this simulation can be performed by a single server since it is just a quantum computation. Malicious servers can do classical/quantum communication during computation in the multiple-servers protocol. In other words, malicious servers might send all user messages to a single server and calculate them alone on that server against the user's intentions. Since the malicious server has unbounded computing power, it makes no difference in computing power whether all computations are performed on a single server or multiple servers. From the given assumptions, the multiple-server blind quantum computation protocol satisfies security against such attacks by malicious servers. If the malicious single server can get calculation information from the aforementioned single-server protocol, then malicious servers can also get calculation information from the multiple-server protocol. This contradicts the assumption. Therefore, if a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation exists, so does a single-server blind quantum computation protocol with a classical user. □

Theorem 5.8. If a single-server blind quantum computation protocol with a classical user exists, then a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation can be constructed from a single-server blind quantum computation protocol with a classical user. Furthermore, if a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation exists, then a single-server blind quantum computation protocol with a classical user can be constructed from a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation.

Proof. The proof for the former follows an approach analogous to that presented in Theorem 5.6.

We then show that if there exists a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation, then there exists a single-server blind quantum computation protocol with a classical user. Assume there is a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation. We consider the scene where a user delegates computation to a single server using the following procedure:

Step 1. Send the first message to the server

The user sends a classical polynomial-size message $m_1 = \{m_{1,1}, \cdots, m_{1,q(n)}\}$ to the server.

Step 2. Return the first message to the user

The server receives the user's message m_1 and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_1 = \{s_{1,1}, \cdots, s_{1,q(n)}\}$, which depends on the server's calculation, to the user.

Step 3. Send a second message to the server

The user gets the message s_1 and performs classical computation based on the message. The user sends a classical polynomial-size message $m_2 = \{m_{2,1}, \cdots, m_{2,q(n)}\}$, which depends on the user's calculation, to the server.

$\vdots$

Step $2i$. Return a i-th message to the user

The server receives the user's message m_i and performs quantum computation based on the message. The server sends a classical polynomial-size message $s_i = \{s_{i,1}, \cdots, s_{i,q(n)}\}$, which depends on the server's calculation, to the user.

Step $2i+1$. Send a $i+1$-th message to the server

The user receives the message s_i and performs classical computation based on the message. The user sends a classical polynomial-size message $m_{i+1} = \{m_{i+1,1}, \cdots, m_{i+1,q(n)}\}$, which depends on the user's calculation, to the server.

$\vdots$

Step p(n). Calculation is complete

The user receives the last message s_l from the server and gets a result about the delegated calculation by performing a classical calculation.

Note that $m_{i,j}$ and $s_{i,j}$ refer to messages in the multiple-servers protocol.

If the server is honest, this simulation can be performed by a single server since it is just a quantum computation. Malicious servers can do classical communication during computation in the multiple-servers protocol. In other words, malicious servers might send all user messages to a single server and calculate them alone on that server against the user's intentions. Since the malicious server has unbounded computing power, there is no difference in computing power whether all calculations are alone on one server or multiple servers. From the given assumptions, the multiple-server blind quantum computation protocol satisfies security against such attacks by malicious servers.

For the single server to obtain information about the user's computation from the above protocol, it would necessitate sending messages that are not sent by multiple servers to the user. The user can reject any message that the multiple servers are not expected to send. The user accepts messages that the multiple servers might potentially send. However, if the single server can obtain information about the user's computation from such messages, multiple servers can also obtain the information. This contradicts the assumption. Therefore, if a multiple-servers without entanglement blind quantum computation protocol that allows servers to do classical communication with each other during computation exists, so does a single-server blind quantum computation protocol with a classical user. □

Theorem 5.9. If a single-server blind quantum computation protocol with a classical user exists, then a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation can be constructed from a single-server blind quantum computation protocol with a classical user. Furthermore, if a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation exists, then a single-server blind quantum computation protocol with a classical user can be constructed from a multiple-servers with entanglement blind quantum computation protocol that allows servers to do classical communication with each other after computation.

Proof. The proof follows an approach analogous to that presented in Theorem 5.8. □

5.4 Discussion

In this chapter, we have defined an SBQC protocol with a classical user. Also, we have defined MBQC protocols to allow servers to communicate with each other. We have shown that if there exists an SBQC protocol with users who have only classical abilities, there are MBQC protocols that allow servers to communicate with each other, and vice versa.

It is not known if a single-server blind quantum computation protocol with a classical user exists [60, 61, 111]. As a result, it is a significant open problem. Multi-server blind protocols are helpful but have received little attention. Our results imply that investigating multi-server blind protocols can reveal the existence of a single-server blind quantum computation protocol with a classical user.

Chapter 6

Summary

In this chapter, we summarize the results of this thesis and discuss future works.

6.1 Summary

We have shown the following results in this thesis:

1. we have proposed a circuit-based blind quantum computation protocol that users can perform without quantum memory,
2. we have proposed the MBQC protocol so that some servers can communicate with each other after computation,
3. we have shown the limits of the MBQC protocol constraints and their relation to the SBQC protocol.

In Chapter 3, we have proposed Protocol 1 (Definition 3.1), a circuit-based blind quantum computation protocol that does not require quantum memory for the user. For Protocol 1, we have proposed a novel encryption technique using gate teleportation to encrypt the T gate, which was not achievable in previous research [54, 55]. In Protocol 1, it suffices for the server to have the ability to execute only Clifford gates. In previous protocols where the user prepares the quantum state, the server has the ability to execute non-Clifford gates [54–56]. Therefore, Protocol 1 significantly relaxes the requirement. Also, this requirement is equivalent to the MF protocol, where the user performs measurements [59]. Consequently, Protocol 1 can be interpreted as being parallel to the MF protocol.

In Chapter 4, we have proposed Protocol 2 (Definition 4.3), a multiple-server quantum computation protocol which some servers can communicate with each other after computation. In Theorem 4.1, we initially extended the two-server protocol to accommodate more servers. For Protocol 2, we have proposed *dummy gates*, a novel encryption technique. We integrated dummy gates with *circuit like brickwork states* into the multiple-server protocol. In Protocol 2, the restriction that servers could not communicate with each other was partially relaxed. Additionally, when performing the multi-server protocol, a user can estimate the risk of user information leakage after computation.

In Chapter 5, we have shown that if there exists an SBQC protocol with users who have only classical abilities, there are MBQC protocols that allow servers to communicate with each other, and vice versa. First, we defined a procedure for an SBQC protocol and an MBQC protocol that is independent of the encryption method. Next, we have shown that if the SBQC protocol exists, it can be emulated with multiple servers, and that if the MBQC protocol exists, it can be simulated with a single server. It is known that an SBQC protocol with users with only classical abilities is highly unlikely [60, 61]. Therefore, there is highly likely no MBQC protocol with which all servers can communicate with each other.

6.2 Future Works

Future works are as follows:

1. Optimization of existing blind quantum computation protocols.

2. The development of blind quantum computation protocols with reduced computational complexity.

3. Proof of the existence or non-existence of a single-server blind quantum computation protocol with a user with only classical abilities.

Acknowledgments

I would like to express my profound gratitude, first and foremost, to my supervisor, Professor Takayuki Miyadera. His instruction has provided me with deep insights into various aspects of quantum theory, thereby significantly enriching my understanding of the field. He has consistently made himself available for consultations on my research, proposing exceptional solutions and offering insightful guidance. His unwavering support has been a fundamental pillar in the advancement of my research throughout my doctoral program. Without him, my work would not have been possible. I extend my gratitude also to Ryosuke Koga and Ikko Hamamura for their fruitful collaborations. Our engaging discussions allowed me to undertake more captivating research. I want to extend my deepest thanks to the current and former members of my laboratory. Special thanks go to Kenzo Ogure, Kazuki Yamaga, and Ryo Takakura for their meaningful assistance in my research. I would also like to thank my mentors for their varied input on my research during my internship. This research has been supported by JST SPRING Grant Number JPMJSP2110 and JSPS KAKENHI Grant Numbers JP23KJ1178.

I would also like to thank my many friends who have supported me in my student life. Among them, I would like to especially thank Yuske Edamoto, Hiroki Uchihara, Hideaki Kitamura, and Kiego Shimano for helping me with my application and slide production despite their busy work schedules. Also, I would like to express my deep gratitude to my gaming friends of ten years, who have shared in daily conversations and games with me. Thanks to them, I could take sufficient breaks from my research, which significantly contributed to maintaining my mental and emotional well-being.

Finally, I would like to thank my family. Thanks to my respectful parents and sister, I have been able to live without inconvenience and concentrate on my research.

References

[1] Y. Sano. Equivalence of single-server and multiple-server blind quantum computation protocols. *Quantum Inf Process*, 22:61, 2023.

[2] Y. Sano. Multi-server blind quantum computation protocol with limited classical communication among servers. *Quantum Inf Process*, 21:88, 2022.

[3] Y. Sano. Blind quantum computation using a circuit-based quantum computer. *J. Phys. Soc. Japan*, 90(12):124001, 2021.

[4] W. Heisenberg. Über quantentheoretische umdeutung kinematischer und mechanischer beziehungen. *Z. Physik*, 33:879–893, 1925.

[5] E. Schrödinger. Quantisierung als eigenwertproblem. *Annalen der Physik*, 384(4):361–376, 1926.

[6] J. VonNeumann. *Mathematische Grundlagen der Quantenmechanik*. Springer, 1932.

[7] R. P. Feynman. Simulating physics with computers. *Int J Theor Phys*, 21:467–488, 1982.

[8] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

[9] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *IEEE International Conference on Computers Systems and Signal Processing*, pages 175–179, 1984.

[10] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London.*

Series A: Mathematical and Physical Sciences, 439(1907):553–558, 1992.

[11] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[12] E. Bernstein and U. Vazirani. Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 11–20, 1993.

[13] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science(FOCS '94)*, pages 124–134, 1994.

[14] D. Coppersmith. An approximate fourier transform useful in quantum factoring. *IBM Research Report* (RC 19642), 1994.

[15] D. R. Simon. On the power of quantum computation. In *Proceedings 35th Annual Symposium on Foundations of Computer Science(FOCS '94)*, pages 116–123, 1994.

[16] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[17] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, page 212–219, 1996.

[18] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[19] D. S. Abrams and S. Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, 79:2586–2589, 1997.

[20] B. M. Boghosian and W. Taylor. Simulating quantum mechanics on a quantum computer. *Physica D*, 120(1):30–42, 1998.

[21] A. T. Sornborger and E. D. Stewart. Higher-order methods for simulations on quantum computers. *Phys. Rev. A*, 60:1956–1965, 1999.

[22] C. Zalka. Simulating quantum systems on a quantum computer. *Proc. R. Soc. Lond. A.*, 454:313–322, 1998.

[23] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing(STOC '03)*, page 59–68, 2003.

[24] A. M. Childs, L. J. Schulman, and U. Vazirani. Quantum algorithms for hidden nonlinear structures. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 395–404, 2007.

[25] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009.

[26] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing(STOC '19)*, page 193–204, 2019.

[27] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, 2021.

[28] J. L. Park. The concept of transition in quantum mechanics. *Found Phys*, 1:23—33, 1970.

[29] W. Wootters and W. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

[30] D. Dieks. Communication by epr devices. *Phys. Lett. A*, 92(6):271–272, 1982.

[31] E. H. Kennard. Zur quantenmechanik einfacher bewegungstypen. *Z. Physik*, 44(6):326–352, 1927.

[32] H. P. Robertson. The uncertainty principle. *Phys. Rev.*, 34:163–164, 1929.

[33] C. A. Fuchs and A. Peres. Quantum-state disturbance versus information gain: Uncertainty relations for quantum information. *Phys. Rev. A*, 53:2038–2045, 1996.

[34] C. A. Fuchs. Information gain vs. state disturbance in quantum theory. *Fortschritte der Phys*, 46:535–565, 1998.

[35] T. Heinosaari and T. Miyadera. Qualitative noise-disturbance relation for quantum measurements. *Phys. Rev. A*, 88:042117, 2013.

[36] A. K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, 1991.

[37] D. Mayers and A. Yao. Quantum cryptography with imperfect apparatus. In *Proceedings 39th Annual Symposium on Foundations of Computer Science(FOCS '98)*, pages 503–509, 1998.

[38] J. Barrett, L. Hardy, and A. Kent. No signaling and quantum key distribution. *Phys. Rev. Lett.*, 95:010503, 2005.

[39] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani. Device-independent security of quantum cryptography against collective attacks. *Phys. Rev. Lett.*, 98:230501, 2007.

[40] U. Vazirani and T. Vidick. Fully device-independent quantum key distribution. *Phys. Rev. Lett.*, 113:140501, 2014.

[41] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing(STOC '89)*, page 12–24, 1989.

[42] S. Aaronson and P. Christiano. Quantum money from hidden subspaces. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing(STOC '12)*, page 41–60, 2012.

[43] A. Broadbent and R. Islam. Quantum encryption with certified deletion. In *Theory of Cryptography*, pages 92–122, 2020.

[44] M. Zhandry. Quantum lightning never strikes the same state twice. or: Quantum money from cryptographic assumptions. *J Cryptol*, 34:6, 2021.

[45] T. Hiroka, T. Morimae, R. Nishimaki, and T. Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In *Advances in Cryptology – ASIACRYPT 2021*, pages 606–636, 2021.

[46] A. Coladangelo, J. Liu, Q. Liu, and M. Zhandry. Hidden cosets and applications to unclonable cryptography. In *Advances in Cryptology – CRYPTO 2021*, pages 556–584, 2021.

[47] A. Broadbent, S. Jeffery, S. Lord, S. Podder, and A. Sundaram. Secure software leasing without assumptions. In *Theory of Cryptography*, pages 90–120, 2021.

[48] T. Yamakawa and M. Zhandry. Verifiable quantum advantage without structure. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science(FOCS '22)*, pages 69–74, 2022.

[49] T. Morimae and T. Yamakawa. Quantum commitments and signatures without one-way functions. In *Advances in Cryptology – CRYPTO 2022*, pages 269–295, 2022.

[50] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[51] J. Feigenbaum. Encrypting problem instances. In *Advances in Cryptology – CRYPTO 1985*, pages 477–488, 1986.

[52] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing(STOC '09)*, page 169–178, 2009.

[53] P. Arrighi and L. Salvail. Blind quantum computation. *Int. J. of Quantum Information*, 4(5):883–898, 2006.

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

[55] D. Aharonov, M. Ben-Or, E. Eban, and U. Mahadev. Interactive proofs for quantum computations. *arXiv:1704.04487*, 2017.

[56] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science(FOCS '09)*, pages 517–526, 2009.

[57] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, 2001.

[58] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68:022312, 2003.

[59] T. Morimae and K. Fujii. Blind quantum computation protocol in which alice only makes measurements. *Phys. Rev. A*, 87:050301, 2013.

[60] S. Aaronson, A. Cojocaru, A. Gheorghiu, and E. Kashefi. Complexity-theoretic limitations on blind delegated quantum computation. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132, pages 6:1–6:13, 2019.

[61] T. Morimae and T. Koshiba. Impossibility of perfectly-secure one-round delegated quantum computing for classical client. *Quantum Info. Comput.*, 19(3–4):214–221, 2019.

[62] B. W. Reichardt, F. Unger, and U. Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of chsh games. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, page 321–322, 2013.

[63] M. McKague. Interactive proofs for BQP via self-tested graph states. *Theory of Computing*, 12(3):1–42, 2016.

[64] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, 1935.

[65] J. S. Bell. On the einstein podolsky rosen paradox. *Phys. Phys. Fiz.*, 1:195–200, 1964.

[66] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, 1969.

[67] A. Aspect. Proposed experiment to test the nonseparability of quantum mechanics. *Phys. Rev. D*, 14:1944–1951, 1976.

[68] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, 1993.

[69] D. Boschi, S. Branca, F. De Martini, L. Hardy, and S. Popescu. Experimental realization of teleporting an unknown pure quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 80:1121–1125, 1998.

[70] D. Bouwmeester, J. W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger. Experimental quantum teleportation. *Nature*, 390:575–579, 1997.

[71] H. J. Briegel, W. Dür, J. I. Cirac, and P. Zoller. Quantum repeaters: The role of imperfect local operations in quantum communication. *Phys. Rev. Lett.*, 81:5932–5935, 1998.

[72] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Rev. Mod. Phys.*, 83:33–80, 2011.

[73] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69:2881–2884, 1992.

[74] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.

[75] H. Barnum, C. M. Caves, C. A. Fuchs, R. Jozsa, and B. Schumacher. Noncommuting mixed states cannot be broadcast. *Phys. Rev. Lett.*, 76:2818–2821, 1996.

[76] M. Piani, P. Horodecki, and R. Horodecki. No-local-broadcasting theorem for multipartite quantum correlations. *Phys. Rev. Lett.*, 100:090502, 2008.

[77] S. Luo, N. Li, and X. Cao. Relation between "no broadcasting" for noncommuting states and "no local broadcasting" for quantum correlations. *Phys. Rev. A*, 79:054305, 2009.

[78] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, USA, 10th edition, 2011.

[79] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, 1994.

[80] D. P. DiVincenzo. Two-bit gates are universal for quantum computation. *Phys. Rev. A*, 51:1015–1022, 1995.

[81] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan. On universal and fault-tolerant quantum computing: a novel basis and a new constructive proof of universality for shor's basis. In *40th Annual Symposium on Foundations of Computer Science(FOCS '99)*, pages 486–494, 1999.

[82] F. Lowenthal. Uniform finite generation of su(2) and sl(2, r). *Can. J. Math.*, 24(4):713–727, 1972.

[83] F. Lowenthal. Uniform finite generation of the rotation group. *Rocky Mt J Math*, 1(4):575–586, 1971.

[84] Y. Shi. Both toffoli and controlled-not need little help to do universal quantum computing. *Quantum Info. Comput.*, 3(1):84–92, 2003.

[85] D. Gottesman. The heisenberg representation of quantum computers. In *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1998.

[86] D. Gottesman and I. L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402:390–393, 1999.

[87] X. Zhou, D. W. Leung, and I. L. Chuang. Methodology for quantum logic gate construction. *Phys. Rev. A*, 62:052316, 2000.

[88] T. Morimae. *Ryoushi Keisan Riron Ryoushi Konpyuuta no Genri.* Morikita Publishing Co., Ltd., JAPAN, 2017.

[89] A. Peres and D. R. Terno. Quantum information and relativity theory. *Rev. Mod. Phys.*, 76:93–123, 2004.

[90] M. Pawłowski, T. Paterek, D. Kaszlikowski, V. Scarani, A. Winter, and M. Żukowsk. Quantum information and relativity theory. *Nature*, 461:1101–1104, 2009.

[91] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing(STOC '85)*, page 291–304, 1985.

[92] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing(STOC '88)*, page 113–131, 1988.

[93] R. Cleve, P. Hoyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity*, pages 236–249, 2004.

[94] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *J Comput Syst Sci*, 66(3):429–450, 2003.

[95] B. S. Cirel'son. Quantum generalizations of bell's inequality. *Lett Math Phys*, 4:93–100, 1980.

[96] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[97] A. Broadbent. Delegating private quantum computations. *Can J Phys*, 93:941–946, 2015.

[98] M. Hayashi and T. Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *Phys. Rev. Lett.*, 115:220502, 2015.

[99] X. Tan and X. Zhou. Universal half-blind quantum computation. *Ann. Telecommun.*, 72(9-10):589–595, 2017.

[100] W. Liu, Z. Chen, J. Liu, Z. Su, and L. Chi. Full-blind delegating private quantum computation. *Comput. Mater. Contin.*, 56:211–223, 2018.

[101] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, 1995.

[102] D. P. DiVincenzo and P. W. Shor. Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.*, 77(15):3260–3263, 1996.

[103] J. Preskill. Fault-tolerant quantum computation. *Introduction to Quantum Computation and Information*, page 213–269, 1998.

[104] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv.*, 52(6):1191–1249, 1997.

[105] X. Chen, H. Chung, A. W. Cross, B. Zeng, and I. L. Chuang. Subsystem stabilizer codes cannot have a universal set of transversal gates for even one encoded qudit. *Phys. Rev. A*, 78(1):012353, 2008.

[106] B. Eastin and E. Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.*, 102:110502, 2009.

[107] S. Bravyi and A. Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, 2005.

[108] A. Fowler, S. Devitt, and C. Jones. Surface code implementation of block code state distillation. *Sci Rep*, 3:1939, 2013.

[109] T. Morimae and T. Koshiba. Impossibility of perfectly-secure one-round delegated quantum computing for classical client. *Quantum Info. Comput.*, 19(3–4):214–221, 2019.

[110] T. Morimae and S. Tamaki. Fine-grained quantum computational supremacy. *Quantum Info. Comput.*, 19(13–14):1089–1115, 2019.

[111] J. F. Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Inf.*, 3(1):1–11, 2017.

[112] Y. Shi. Both toffoli and controlled-not need little help to do universal quantum computing. *Quantum Info. Comput.*, 3(1):84–92, 2003.

[113] J. Fitzsimons and E. Kashefi. Unconditionally verifiable blind computation. *Phys. Rev. A*, 96:012303, 2017.