Comparison on security of single server and multiple servers blind quantum protocols

Yuichi Sano*1

1Department of Nuclear Engineering, Kyoto University, Nishikyo-ku, Kyoto 615-8540, Japan

June 11, 2021

Abstract

A user who delegates quantum computation to others hopes to keep information on the computation secret. Several different blind quantum computation protocols which attain this goal have been proposed thus far. While in single server protocols a user needs to manipulate quantum states, multiple server protocols do not require a user to have any quantum equipment although they prohibit any communication among servers even after the computation. It is not a straightforward task for a user to decide the most proper protocol among them as each protocol has its own merits and defects. In this paper we introduce a notion called usability which enables us to compare the different protocols. In particular we consider a blind protocol with multiple servers which allows classical communication during and after the computation, and compare it with a single server protocol. As a result we find that their usability are partially equal to each other.

1 Introduction

Quantum computers are expected to be the next-generation computers because they can perform calculations that are considered impossible with classical computers. For example, Shor’s algorithm [1] solves prime factorization problems in polynomial time using the quantum Fourier transform, and Grover’s algorithm [2] is known as the fastest unordered database search. However, due to the fragility of the quantum effects against external noise, the physical implementation of quantum computers is hard and needs expensive technology. For this reason, the quantum computers, not being owned by each user, will most likely be used as servers in cloud services. An important issue in such service is

*sano.yuichi.77v@st.kyoto-u.ac.jp
that a server provider may illegally obtain information on a calculation which is delegated by a user. Therefore, we need a security, which is called blind quantum computation protocol, that allows us to perform calculations without revealing the contents of the calculations [3–11]. The protocol encrypts the inputs, outputs, and computational algorithms of the computations. An analogy in classical computation is the homomorphic encryption [12]. However, the homomorphic encryption only encrypts input and output, and does not hide the calculation algorithm. Thus the blind quantum computation protocol is considered to provide stronger security than the homomorphic encryption. Childs has shown that a user who has the ability to use quantum memories and create qubits, i.e., a user who has a small quantum computer, can perform blind quantum calculations by communicating with a server that has a universal quantum computer [4]. Broadbent, Fitzsimons and Kashefi proposed a protocol using the ability to create a specific quantum state, send it to a server and classical communication, without quantum memory [6]. There are several other known blind quantum computation protocols that can be executed by quantum communication with a quantum single server [7–9]. To relax the requirements on user’s ability, protocols using multiple servers have been proposed [10,11]. In these protocols a user needs to have only a classical computer and communicates with multiple servers who share entangled qubits. Those protocols are useful because the user does not need to have any quantum equipment. It is important to note, however, that no classical and quantum communication is allowed between multiple servers.

As mentioned above, so far several different blind quantum computation protocols have been proposed. Each protocol has its merit and defect depending on its security attained and its requirements to be satisfied. Therefore, user will choose the most secure and least cost protocol by comparing their own capabilities and the restriction of the server. However, it is not possible to simply compare the safety of a single server’s protocol and multiple server’s protocol. The single server promises to be Blindness even after the computation is finished. On the other hand, in the case of multiple servers, if the servers do classical communicate with each other after the calculation, the blindness is broken and the user’s information is not kept secret. Therefore, it is not possible to simply compare these two. The purpose of this study is to provide a reasonable standard for comparing the security of the single server protocols and multiple servers protocols.

In this paper, we first define a notion usability in order to compare blind quantum computation protocols. Then, we consider the case where the restrictions of the blind protocol with multiple servers is relaxed. We show that a multiple servers protocol that allows classical communication even during computation is equal in the usability to a single server protocol. We then show that a multiple servers protocol allowing communication among servers after the computation, when the problem used for computation is restricted to PSPACE, is equal in the usability to a single server protocol in which a user can use only classical computers and communication. These results are an important criterion for comparing single server’s protocols with multiple server’s protocols.
2 Preliminaries

In this chapter, we describe blind protocol needed to show the main result, and define a new security for blind protocols. We also cite some important results on computational complexity theory, which is necessary for the proof of the theorem.

2.1 Blind quantum computation protocol

In this section, we first describe a blind protocol. Blind protocol, first proposed by Childs [4], is a security feature that hides not only the input and output but also the computation algorithms from the server. This means that when calculated using the Blind protocol, the server does not even know what calculation the user has done. Homomorphic encryption [12], known as a similar on classical computation, encrypts only input and output. Thus, the Blind protocol is more powerful than homomorphic encryption, but it is only found in quantum computation. Of course, quantum computation trivially includes classical computation, so classical computation can also use blind protocols as part of quantum computation. Broadbent, Fitzsimons and Kashefi give the following definition of the Blind protocol [6].

**Definition 1 (Blindness [6 Definition 2]).** Let P be a quantum delegated computation on input X and let L(X) be any function of the input. We say that a quantum delegated computation protocol is blind while leaking at most L(X) if, on user’s input X, for any fixed Y = L(X), the following two hold when given Y :

1. The distribution of the classical information obtained by Server in P is independent of X.
2. Given the distribution of classical information described in 1, the state of the quantum system obtained by Server in P is fixed and independent of X.

In this paper, let the condition of definition [4] is called Blindness, and let the protocol that satisfy Blindness is called a Blind protocol. This definition refers to that the server only gets the information that the server gets by any calculation, such as the size of the circuit, and not the information that depends on the calculation.

In this study, we focus on the situation in which the protocol can be executed, rather than the specific protocol configuration. For example, the Childs’s protocol is available when the user can have quantum memory and to execute SWAP gates, and the user and server are in quantum communication. In this way, we classify the availability situation of each protocol as each “usability”. This means that protocols that can be used under the same conditions are included in the same usability.
Definition 2 (Usability). The conditions used to execute the blind protocol, i.e., the user’s ability, the server’s ability, restrictions on the server, and the ability of the user and server to communicate, etc., are called “usability”. For usability, the amount of computation time and the amount of resources are not important as long as they are of equal order. In other words, the same condition protocols that are computation time $O(n)$ and $O(n^3)$ are the same usability since both are polynomial time. But the same condition protocols that are computation time $O(n)$ and $O(e^n)$ are the different usability.

We define the case where usability is equivalent even if different conditions.

Definition 3 (Equivalence of usability). We define equivalent of usability of protocol A and B as that if protocol A exists, protocol B also exists, and if protocol B exists, protocol A also exists.

We define the inclusion of usability.

Definition 4 (Inclusion of usability). Consider the usability of protocol A and B. That the usability of protocol A is included in the usability of protocol B, we mean that if protocol exists A, then protocol B exists. We denote it as

$$B \supset A.$$  

(1)

Usability can be used to compare blind protocols. Suppose that there are two protocols A and B that are not simply comparable. If there is a protocol C such that

$$A \supset C,$$  

(2)

and the usability of protocol C is equal to the usability protocol of B, then

$$A \supset B$$  

(3)

holds. In this case, we can say that B is more usable than A.

2.2 Computational complexity theory

In this study, we use computational complexity theory to measure the difficulty of computation. In this section, we introduce some theorems used in the main result.

Blindness is not directly part of computational complexity theory, but it is known to be closely related. For example, in showing $\text{MIP}^* = \text{QMIP}$ [10], Reichardt, Unger, and Vazirani showed that blind quantum computation protocol is possible with two servers and user who can only use classical communication and classical computer. Previous research [14][15] have also used approach using computational complexity to study the possibility that there may not be a blind quantum computation protocol with a single server and user who are only capable of classical communication and classical computation.

In this paper, we consider that the server should have unlimited computational resources following the interactive proof system in computational complexity. In this case, the relationship between a server and a user who can only
use a classical communication and a classical computer, is exactly equivalent to
the complexity class IP. IP is a complexity class of problems where a server with
infinite computing resources and a user with a classical computer can verify the
solution by using polynomial times classical communication. It is known that IP
is equal to PSPACE, a complexity class of problems that deterministic Turing
machines can solve using polynomial-sized memory.

**Theorem 5** ([16]).

\[ \text{IP} = \text{PSPACE} \]

And as lemmas to show this theorem, it is known that the TQBF decision
problem is PSPACE-complete and PSPACE is included in IP [17].

**Lemma 6.** TQBF is a PSPACE-complete problem.

**Lemma 7.**

\[ \text{PSPACE} \subseteq \text{IP} \]

About TQBF, there is an algorithm that uses an interactive proof system
to accept with probability 1 if a QBF $\psi$ is true, and with probability $O(1/n)$
at most if it is false [17]. Since TQBF is a PSPACE-complete problem, any
problems contained in PSPACE can be solved by polynomial-time reduction
from it to TQBF.

## 3 Equivalence of usability between single server’s
protocol and multiple server’s protocol

In this chapter, we show that the usability of a protocol with a single server
is equal to the usability of a certain protocol with multiple servers. Here, user
are only available for classical communication and classical computing in both
protocols. First, we compare the protocols of multiple servers that allow classical
communication between servers even during computation with the protocol of a
single server. Next, we consider multiple servers where classical communication
between servers is not possible during the calculation, but after the calculation,
classical communication between servers is possible.

### 3.1 Multiple servers capable of classical communication
during computation

In this section, we consider a blind protocol with multiple servers that allow classical
communication between servers even during computation. First, we
define the condition of the single server to be compared.

**Definition 8** (Protocol $\mathcal{S}$). We define protocol $\mathcal{S}$ to be a blind protocol of the
usability consisting of the following conditions.

**User ability** Classical computer.
Communication format  Classical communication.

Server ability  Unlimited computational resources.

Server restriction  Single server.

Computation time  Polynomial.

Protocol $\mathcal{S}$ is considered to be the most convenient condition for blind protocols. However, there is currently no known Blind protocol that can run under protocol $\mathcal{S}$ [20].

The next step is to define the usability for multiple servers that can classically communicate with each other during the calculation.

Definition 9 (Protocol $\mathcal{M}_{\text{during}}$). We define protocol $\mathcal{M}_{\text{during}}$ to be a blind protocol of the usability consisting of the following conditions.

User ability  Classical computer.

Communication format  Classical communication.

Server ability  Unlimited computational resources.

Server restriction  Multiple servers that can classically communicate with each other during the calculation.

Computation time  Polynomial.

We show that the usability of these two protocols is equal.

Theorem 10. Usability of protocol $\mathcal{S}$ and protocol $\mathcal{M}_{\text{during}}$ are equal.

Proof. Let $N = \mathcal{O}(\text{poly}(n))$ be the number of servers for the number of input bits $n$, since the computation time is polynomial.

We first show that if there exists protocol $\mathcal{S}$, then there exists protocol $\mathcal{M}_{\text{during}}$. The user executes protocol $\mathcal{S}$ only with one of the $N$ servers with which classical communication is possible during the computation, and does not communicate with the remaining $N-1$ servers. In this case, the information obtained by the $N$ servers does not increase compared to the single server protocol, so it is a blind protocol. Therefore, if there exists protocol $\mathcal{S}$, then there exists protocol $\mathcal{M}_{\text{during}}$.

Next, we show that if there exists protocol $\mathcal{M}_{\text{during}}$, then there exists protocol $\mathcal{S}$. Assume that protocol $\mathcal{M}_{\text{during}}$ exists. Let $m_j$ be the message that the $j$-th server receives from the user, and let $s_j$ be the message that the $j$-th server sends to the user. Following condition of protocol $\mathcal{M}_{\text{during}}$, even if the server side behaves in such a way that only one server performs the quantum computation and the other $N-1$ servers communicate the classical information received from the user to the server that performs the quantum computation, it is still a blind protocol. We assume that the server that is performing quantum computation without loss of generality is the first server. This is shown as a specific protocol.
1. The user sends messages \(\{m_1, \ldots, m_N\}\) to each server.

2. The first server receives messages \(\{m_2, \ldots, m_N\}\) from each server and performs quantum computation.

3. The first server then sends messages \(\{s_2, \ldots, s_N\}\) to each server from the calculation results.

4. The user receives messages \(\{s_1, \ldots, s_N\}\) from each server.

5. Repeat steps 1.~4. until the calculation is complete.

We show a protocol that simulates this protocol on a single server.

1. User send a message \(\{m_1, \ldots, m_N\}\) to a single server.

2. A single server perform the same quantum computation as the first of the multiple servers.

3. The User receives the message \(\{s_1, \ldots, s_N\}\) from the result of the calculation.

4. Repeat steps 1.~3. until the calculation is complete.

Suppose that this protocol \(\mathcal{S}\) does not satisfy blindness and is not a blind protocol. However, in protocol \(\mathcal{M}_{\text{during}}\), which is assumed to be a blind protocol, the message that the first server gets and the message that it computes and sends is the same as in the single server. Therefore, if the blindness is not satisfied even in protocol \(\mathcal{S}\), then the blindness is not satisfied even in protocol \(\mathcal{M}_{\text{during}}\). It contradicts the assumption. The protocol that makes protocol \(\mathcal{M}_{\text{during}}\) simulate a single server is a blind protocol. Therefore, if there exists protocol \(\mathcal{M}_{\text{during}}\), then there exists protocol \(\mathcal{S}\).

This result shows that the usability of the two protocols is equivalent. 

Protocol \(\mathcal{S}\) is one of the final objectives in the study of blind protocols. In addition it turns out that the same objective can be reached by studying protocols with multiple servers.

Although protocol \(\mathcal{M}_{\text{during}}\) did not consider the sharing of entanglements between servers, even if entanglements were shared, the same argument can be used to say that protocol \(\mathcal{S}\) and usability are equal.

### 3.2 Multiple servers capable of classical communication after calculation

In this section, we compare the protocol \(\mathcal{S}\) and the protocol using multiple servers, which allows classical communication between servers after the calculation. Unlike the previous section, servers cannot communicate classically with each other during the computation, so it makes a difference whether they share entanglement or not. First, we define the condition of the multiple server’s protocol.
Definition 11 (Protocol $M_{\text{after}}(C)$). We define protocol $M_{\text{after}}(C)$ to be a blind protocol of the usability consisting of the following conditions.

**User ability** Classical computer.

**Communication format** Classical communication.

**Server ability** Unlimited computational resources.

**Server restriction** Multiple servers that can not classically communicate with each other during the calculation, but can classically communicate with each other after the calculation.

**Computation time** Polynomial.

**Complexity Class** The complexity class of the computation used by the user in the protocol is $C$. To be precise, it is the computation class $C$ that is used to interact with the server if the server behaves as per the protocol. $C$ can be said to be a class of problems that can be used to solve BQP judgment problems while satisfying blindness in such situations. In reality, the server may behave in a manner that deviates from the protocol. Therefore, this protocol define that the server cannot deviate from the algorithm by distributing the algorithm to multiple servers.

Definition 12 (Protocol $M^*_{\text{after}}(C)$). We define protocol $M^*_{\text{after}}(C)$ to be a blind protocol of the usability consisting of the following conditions.

**User ability** Classical computer.

**Communication format** Classical communication.

**Server ability** Unlimited computational resources and sharing entangled qubits between servers.

**Server restriction** Multiple servers that can not classically communicate with each other during the calculation, but can classically communicate with each other after the calculation.

**Computation time** Polynomial.

**Complexity Class** The complexity class of the computation used by the user in the protocol is $C$. To be precise, it is the computation class $C$ that is used to interact with the server if the server behaves as per the protocol. $C$ can be said to be a class of problems that can be used to solve BQP judgment problems while satisfying blindness in such situations. In reality, the server may behave in a manner that deviates from the protocol. Therefore, this protocol define that the server cannot deviate from the algorithm by distributing the algorithm to multiple servers.
In other words, protocol \( M_{\text{after}}(C) \) and \( M^*_{\text{after}}(C) \) are protocols in which a decision problem such as solving BQP with blindness is in class \( C \), and the blindness is still satisfied even if the server communicates classically after the computation. Since servers cannot communicate classically with each other during computation, situation of protocol \( M_{\text{after}}(C) \) is equal to complexity class MIP, situation of protocol \( M^*_{\text{after}}(C) \) is equal to complexity class MIP* . MIP is a class of problems that can be verified by multiple servers qubits and user communicating with classical communication. MIP* is a class of problems that can be verified by multiple servers who share entangled and user communicating with classical communication. It is known that MIP is equal to the complexity class NEXP and MIP* is equal to the complexity class RE \([18,19]\). Therefore, if Complexity Class \( C \) is not specified, the values are protocol \( M_{\text{after}}(\text{NEXP}) \) and protocol \( M^*_{\text{after}}(\text{RE}) \), respectively.

If we restrict the complexity class of problems used for those protocols to PSPACE, respectively, we show that usability of protocol \( S \) are equal to those usability.

**Theorem 13.** Usability of protocol \( S \) and protocol \( M_{\text{after}}(\text{PSPACE}) \) are equal.

**Proof.** Let \( N = \mathcal{O}(\text{poly}(n)) \) be the number of servers for the number of input bits \( n \), since the computation time is polynomial.

We first show that if there exists protocol \( S \), then there exists protocol \( M_{\text{after}}(\text{PSPACE}) \). The user executes protocol \( S \) only with one of the \( N \) servers with which classical communication is possible during the computation, and does not communicate with the remaining \( N-1 \) servers. In this case, the information obtained by the \( N \) servers does not increase compared to the single server protocol, so it is a blind protocol. Also, since protocol \( S \) has the same conditions as IP, the complexity class of the problem used is at most PSPACE. Therefore, if there exists protocol \( S \), then there exists protocol \( M_{\text{after}}(\text{PSPACE}) \).

Next, we show using the following method that if there exists protocol \( M_{\text{after}}(\text{PSPACE}) \), then there exists protocol \( S \). We define the following hacking function.

**Definition 14 (Hacking function).** For any quantum circuit \( C(n) \) with \( n \) qubit inputs that the user wants to execute, let the hacking function \( H_i(\cdot;\cdot) \) be the following function that the server can execute. Let \( m_i \) be a message from the user to the server, \( s_i \) be a message from the server to the user, \( l \) be the last round number of the protocol’s classical communication. And let \( \text{Inf(NO)} \) be a message indicating that no information could be obtained that would break Blindness. Also, let \( \text{Inf}(C(n)) \) be the information such that Blindness is not satisfied for \( C(n) \). If the user can choose \( m_i \) randomly, we assume that the user sends the most desirable \( m_i \) for the server.

If the server cannot break the Blindness and retrieve the information, when \( i \in \{0,\ldots,l-1\} \)

\[
H_i(m_1,\ldots,m_i; s_1,\ldots,s_{i-1}) = s_i,
\]  

(4)
when $i = l$

$$H_l(m_1, \ldots, m_l; s_1, \ldots, s_{l-1}) = \text{Inf}(\text{NO}).$$  \hspace{1cm} (5)$$

If the server can break the blindness and obtain the circuit information,

when $i \in \{0, \ldots, l - 1\}$

$$H_i(m_1, \ldots, m_i; s_1, \ldots, s_{i-1}) = s_i,$$ \hspace{1cm} (6)

when $i = l$

$$H_l(m_1, \ldots, m_l; s_1, \ldots, s_{l-1}) = \text{Inf}(\text{C}(n)).$$ \hspace{1cm} (7)

By using such a hacking function, we can consider the output of the last round of the hacking function equal to whether or not the server breaks the Blindness of the protocol.

We show that if protocol $M_{\text{after}}(\text{PSPACE})$ exists, then there exists protocol $S$ using proof by contradiction. For this purpose, we first assume that protocol $M_{\text{after}}(\text{PSPACE})$ exists but there is no protocol $S$.

We consider simulating protocol $M_{\text{after}}(\text{PSPACE})$ on a single server. In other words, we send all the messages sent to $N$ servers to a single server, and simulate the computation of protocol $M_{\text{after}}(\text{PSPACE})$ in the single server. The information obtained by a single server by following completely the protocol is equal to the information obtained by multiple servers by classical communication after the computation. Therefore, Blindness is not satisfied due to the behavior of the single server during the computation.

Since TQBF is PSPACE-complete problem, we can do polynomial-time reduction from the decision problem used in the protocol to TQBF. For a language $L$, which is a TQBF, we know that the user and the server can interact in such a way that for a string $x$, if $x \in L$, $x$ is accepted with probability 1, and if $x \notin L$, $x$ is accepted with probability $O(1/n)$ \cite{17}. In IP, TQBF is determined by the user by solving a function that recursively represents the Boolean value of the defined QBF. The server sends a hint to the user to solve the function in each phase, and the user sends a random value if he accept the hint, or the calculation ends there if he reject it. The user randomly selects a value from a finite field of at least $n^4$ in size and sends it to the server as a message during the calculation. Please refer to \cite{17} for specific methods.

When a single server perform the TBQF that is made by a polynomial-time reduction, if the server does not have access to information about the user’s original computation, i.e., if it is a blind protocol, it can also perform the blind computation on multiple servers, as described above. Thus, by using the hacking function in the process of proving the TBQF, the server can get information about the user’s original computation from the assumption. In other words, there exists $\{m_1, \ldots, m_l\}$ and $\{s_1, \ldots, s_{l-1}\}$ such that

$$H_l(m_1, \ldots, m_l; s_1, \ldots, s_{l-1}) = \text{Inf}(\text{C}(n)).$$ \hspace{1cm} (8)
On the other hand, in protocol $M_{\text{after}}(\text{PSPACE})$, the servers communicate classically with each other after the calculation, so that they can learn about the problems in PSPACE used for the calculation. If servers know the problem, servers can do polynomial time reduction from the problem in PSPACE to the TBQF problem. The servers cannot communicate about the TBQF problem with the user, so they cannot know the user’s response to any messages from servers to the user. However, recall that the message from the user to servers when proving the TBQF is a random value from a finite field of at least $n^4$ in size. This means that the server can choose the message $\{m_1, \ldots, m_l\}$ from the user as the server likes. Thus, by using this $m$ and the hacking function, servers can calculate about $\{s_1, \ldots, s_{l-1}\}$. Therefore, by assumption, the servers can calculate $H_l(m_1, \ldots, m_l; s_1, \ldots, s_{l-1}) = \text{Inf}(C(n))$, (9) by obtaining $\{m_1, \ldots, m_l\}$ and $\{s_1, \ldots, s_{l-1}\}$. This means that blindness can also be broken in protocol $M_{\text{after}}(\text{PSPACE})$. But this result contradicts the assumption. Hence, if there exists protocol $M_{\text{after}}(\text{PSPACE})$, then there exists protocol $S$.

Above results shows that the usability of the two protocols is equivalent.

The same is true for protocol $M^*_{\text{after}}(\text{PSPACE})$, since sharing entanglement is not important in this proof.

**Theorem 15.** Usability of protocol $S$ and protocol $M^*_{\text{after}}(\text{PSPACE})$ are equal.

**Proof.** Proof of this theorem can also be shown by the same argument as Theorem 13.

This shows that there is a condition with usability equal to protocol $S$ for multiple servers that can communicate classically after the calculation. This result does not necessarily give a strict upper bound. We think that it is important to investigate $M_{\text{after}}(C)$ such that PSPACE $\subset C$. Unlike in protocol $M_{\text{during}}$, we can consider a looser restriction in protocol $M_{\text{after}}(\text{PSPACE})$ and $M^*_{\text{after}}(\text{PSPACE})$. For example, $M^*_{\text{after}}(\text{RE})$ allows user to use undecidable problem such as halting problem. With such a powerful tool, it may be possible to delegate blind quantum computations. However, such a problem is computationally infeasible for real servers, and it seems even more infeasible for protocol $S$.

4 Discussion

In this paper, usability is defined to compare different blind protocols. We showed that protocol $S$, which user use only classical computation and classical communication and interact with a single server, has the same usability as a protocol with multiple servers with some restrictions imposed. In particular, it is an important result that the usability of protocol $M_{\text{after}}(\text{PSPACE})$
and $\mathcal{M}_{after}^*$(PSPACE) where there is no classical communication between servers during computation is equal to the usability of protocol $\mathcal{S}$.

The contributions that can be indirectly derived from the results of this study are shown below. First, the results of our study provide an important criterion for comparing protocols with different conditions. Next, we found that studying blind protocols with multiple servers leads to studying blind protocols with a single server. And finally, it provides a tool in proving whether the absence of blind protocols such as protocol $\mathcal{S}$ or not.

In this paper, we compared the extreme protocols of multiple servers and single server, but by using usability, it is possible to compare even other conditions. As more protocols are proposed in the future, comparisons between protocols will become more important.

Currently, there is no known protocol of condition that corresponds to protocol $\mathcal{S}$, $\mathcal{M}_{after}$(PSPACE) and $\mathcal{M}_{after}^*$(PSPACE). A known protocol of single server requires user to have quantum capabilities [6–9]. Also, the known protocol of multiple servers prohibits classical communication between servers during and after a calculation. We think that the protocol of multiple servers is worth more research.

It is still not clear whether a blind protocol such as protocol $\mathcal{S}$ is feasible in the first place. Previous research [14,15] has known collateral evidence that protocol $\mathcal{S}$ is not feasible, but no definite conclusion has been reached. Theorem 13 and Theorem 15 in this study can be used to investigate that. For example, if we show that $\mathcal{M}_{after}^*$(RE) does not exist, then from the inclusion relation of usability, we show that protocol $\mathcal{S}$ does not exist either.

Acknowledgment

We would like to thank Takayuki Miyadera for many helpful comments, and we are grateful to Kazuki Yamaga for important advice.

References

[1] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring”. Proceedings 35th Annual Symposium on Foundations of Computer Science, 124-134, 1994.

[2] L. K. Grover, “A fast quantum mechanical algorithm for database search”. In Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing, 212–219, 1996.

[3] M. Abadi, J. Feigenbaum, and J. Kilian, On hiding information from an oracle. Journal of Computer and System Sciences, 39, 21-50, 1989.

[4] A. M. Childs, Secure assisted quantum computation. Quantum Information and Computation, 5, 456-466, 2005.
[5] D. Aharonov, M. Ben-Or, and E. Eban, Interactive proofs for quantum computations. arXiv:1704.04487

[6] A. Broadbent, J. Fitzsimons and E. Kashefi, Universal Blind Quantum Computation. In Proceedings of the 50st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009), 517-526, 2009.

[7] T. Morimae and K. Fujii, Blind quantum computation protocol in which Alice only makes measurements. Physical Review A, 87, 050301, 2013.

[8] M. Hayashi and T. Morimae, Verifiable Measurement-Only Blind Quantum Computing with Stabilizer Testing. Physical Review Letters, 115, 220502, 2015.

[9] Y. Sano, “Blind Quantum Computation Using a Circuit-Based Quantum Computer”. arXiv:2006.06255, 2020.

[10] 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, 321–322, 2013.

[11] M. McKague, “Interactive Proofs for BQP via Self-Tested Graph States”. Theory of Computing, 12, 1, 2016.

[12] C. Gentry, “Fully homomorphic encryption using ideal lattices”. In Proceedings of the forty-first annual ACM symposium on Theory of computing, 169–178, 2009.

[13] J. Fitzsimons and E. Kashefi, Unconditionally verifiable blind computation. Physical Review A, 96, 012303, 2017.

[14] T. Morimae and T. Koshiba, “Impossibility of perfectly-secure one-round delegated quantum computing for classical client”. Quantum Information and Computation, 19, 0214-0221, 2019.

[15] S. Aaronson, A. Cojocaru, A. Gheorghiu, and E. Kashefi, “Complexity-Theoretic Limitations on Blind Delegated Quantum Computation”. 46th International Colloquium on Automata, Languages, and Programming, 6, 2019.

[16] S. Adi, “IP = PSPACE”. Journal of the ACM, 39, 869–877, 1992.

[17] M. Sipser, “Introduction to the Theory of Computation”. ACM Sigact News, 27, 27-29, 1996.

[18] L. Babai, L. Fortnow, and C. Lund, “Nondeterministic exponential time has two-prover interactive protocols”. Computational Complexity, 1, 3-40, 1991.
[19] Z. Ji, A. Natarajan, T. Vidick, J. Wright, and H. Yuen, “MIP* = RE”. arXiv:2001.04383 2020.

[20] J. F. Fitzsimons, Private quantum computation: an introduction to blind quantum computing and related protocols. npj Quantum Inf 3, 23, 2017.