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

Yuichi Sano

Received: 28 May 2022 / Accepted: 26 December 2022 / Published online: 18 January 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
Because quantum computers are expensive, it is envisaged that individuals who want to utilize them would do so by delegating their calculations to someone who has a quantum computer. When quantum computer users delegate computations to quantum servers, they wish to keep information about their calculations hidden from the servers. The protocol of delegating a calculation while hiding information about the calculation from the server is called blind quantum computation protocol. Prior research on single-server blind quantum computation protocol required users to have quantum capabilities. Prior research on multiple-server blind quantum computation protocols required users to have just classical capabilities but imposed limits on the server-to-server communication. There are no known single-server blind quantum computation protocols with a classical user and multiple-server blind quantum computation protocols that allows servers to communicate freely with each other. We show that the existence of these protocols is equivalence.

Keywords Quantum computation · Quantum blind computing · Quantum encryption · Security of quantum computation

1 Introduction
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 [1] solves prime factorization problems in polynomial time using the quantum Fourier transform, and Grover’s algorithm [2] 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. As a result, quantum computers will most likely be employed as servers in cloud services rather than being owned by individual customers. An essential concern with such cloud services is that a service provider may gain information on a calculation delegated by a user unlawfully. As a result, a form of security is required, namely blind quantum computation protocols, which would allow users to perform calculations without revealing their contents [3–13]. The inputs, outputs, and processes of these blind quantum computation protocols are encrypted. Childs showed that a user with quantum memory and the ability to manipulate qubits, i.e., a user with a mini quantum computer, may execute blind quantum computation via quantum communication with a server equipped with a universal quantum computer [4]. Broadbent, Fitzsimons, and Kashefi proposed a protocol that users, who do not have quantum memory, create a specific quantum state, send it to a server, and do classical communication with the server [6]. Several more blind quantum computation protocols are also carried out by a user doing quantum communication with a single server [7–10]. Protocols utilizing many servers have been proposed to ease the limitations on the user’s abilities [11–13]. 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 capabilities. 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 study, we show that if there exists a single-server blind quantum computation protocol with users who have only classical capabilities, there is a multi-server blind quantum computation protocol that enables servers to communicate freely with each other, and vice versa. We show specifically that if the single-server protocol exists, it can be emulated with multiple servers, and that if the multiple-server 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 of our findings, 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 computational capabilities.

2 Preliminaries

In this section, we describe a blind quantum computation protocol. Then, we define single server protocol with users who only have classical capabilities and a multiple-
server protocol that allows servers to freely communicate. Note that in the following, \( n \) represents the number of input bits.

2.1 Blind quantum computation protocols

In this subsection, we first describe blind quantum computation protocols. A blind quantum computation 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 using a blind quantum computation protocol, the server does not even know what calculation the user has performed. Naturally, quantum computation trivially includes classical computation; thus, the classical computation can also use blind quantum computation protocols as part of quantum computation. Broadbent, Fitzsimons, and Kashefi gave the following definition for a blind quantum computation 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 1 be called *blindness*, and let the protocol that satisfies the blindness be called a blind quantum computation protocol. This definition refers to the fact that the server only gets information obtained through calculations, such as the size of the circuit, and not information that is dependent on the calculation.

2.2 Single-server Protocol and Multi-server Protocol

In this subsection, we define a single-server blind quantum computation protocol in which users can only use classical capabilities, and a multi-server blind quantum computation protocol in which servers can freely communicate with each other. We will assume in the following section that honest servers have quantum computing power and malicious servers have unbounded computing power.

We first define a single-server blind quantum computation protocol in which the user has only classical capabilities.

**Definition 2** (Single-server blind quantum computation protocol with a classical user) A user has classical computing and classical communication capabilities. If the following user-server interaction’s delegating computation protocol satisfies blindness, we define it as a single-server blind quantum computation protocol with a classical user. The number of protocol steps \( p(n) \) is the polynomial size of \( n \).
Step 1. Send the first message to the server
The user sends a classical polynomial-sized 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 transmits to the user a classical polynomial-sized 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-sized message \( m_2 \) to the server, the size which relies on the content 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-sized message \( s_i \), which depends on the content of 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-sized message \( m_{i+1} \), which depends on the content of 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 executing a classical calculation.

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

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

Definition 3 (Multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other) A user is capable of both classical computing and classical communication. The number of servers is polynomial-size \( q(n) \). Servers do not share quantum entanglements, and only classical communication is allowed between servers. If the following user–server interaction’s delegating computation protocol satisfies blindness, we define it as a multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other. The number of protocol steps \( p(n) \) is the polynomial size of \( n \).
Step 1. **Send first messages to the servers**

The user sends classical polynomial-sized 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-sized message $s_{1,j}$, which depends on the content of 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-sized messages to all servers, the size of which depends on the content of 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-sized message $s_{i,j}$, which depends on the content of 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-sized messages to all servers, the size of which depends on the content of 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 performing a classical calculation.

**Definition 4** *(Multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other)* A user has classical computing and classical communication capabilities. The number of servers is polynomial-size $q(n)$. Servers share quantum entanglements, and servers can communicate in both classical and quantum. When quantum communication as well as classical communication is available in the server’s computation step of the multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other*(Definition 3)*, it is defined as a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other. The number of protocol steps $p(n)$ is the polynomial size of $n$. 
These definitions just state that the blind protocols performed by the afore-
mentioned 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.

Whether the server shares entanglement or not, the user can delegate quantum
computation to the server if the server is honest because the server has quantum
computation capability. There is no requirement for actual quantum communication
in the protocol with shared entanglement because quantum teleportation is conceivable
by utilizing classical communication plus entanglement.

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

In this section, we show that if the single-server blind quantum computation protocol
defined in the previous section exists, then there is a multi-server blind quantum
computation protocol that allows servers to communicate with each other, and vice
versa, if the multi-server blind quantum computation protocol exists, then there is the
single-server blind quantum computation protocol.

Theorem 1 If a single-server blind quantum computation protocol with a classical
user exists, then a multiple-server without entanglement blind quantum computation
protocol that allows servers to communicate freely with each other can be con-
structed from a single-server blind quantum computation protocol with a classical
user. Furthermore, if a multiple-server without entanglement blind quantum com-
putation protocol that allows servers to communicate freely with each other exists,
then a single-server blind quantum computation protocol with a classical user can be
constructed from a multiple-server without entanglement blind quantum computation
protocol that allows servers to communicate freely with each other.

Proof We first show that if there exists a single-server blind quantum computation
protocol with a classical user, then there exists a multiple-server without entanglement
blind quantum computation protocol that allows servers to communicate freely with
each other. 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 protocol, we explore the scenario when a user delegates computation to
multiple servers. It is important to note that the terms \( m_i \) and \( s_i \) relate to messages in
the single-server blind quantum computation protocol with a classical user.

Step 1. Send first messages to the servers
The user sends classical polynomial-sized messages to all servers. Let
\( m_{1,j} \) be the message that the user sends to the \( j \)-th server, and \( m_{1,1} = m_1 \) 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-sized
message $s_{1,j}$, which depends on the content of 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-sized messages to all servers, the size of which depends on the content of the user’s calculation. Let $m_{2,j}$ be the message that the user sends to the $j$-th server, and $m_{2,1} = m_2$ and $j \neq 1$ message $m_{2,j}$ is a meaningless string.

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-sized message $s_{i,j}$, which depends on the content of 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-sized messages to all servers, the size of which depends on the content of the user’s calculation. Let $m_{i+1,j}$ be the message that the user sends to the $j$-th server, and $m_{i+1,1} = m_{i+1}$ and $j \neq 1$ message $m_{i+1,j}$ is a meaningless string.

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 performing a classical calculation.

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-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other.

We then show that if there exists a multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other, then there exists a single-server blind quantum computation protocol with a classical user. Assume there is a multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other. We consider the scene where a user delegates computation to a single server using the protocol described below. Note that $m_{i,j}$ and $s_{i,j}$ refer to messages in the multiple-
server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other.

**Step 1. Send the first message to the server**
The user sends a classical polynomial-sized 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-sized message \( s_1 = \{s_{1,1}, \cdots, s_{1,q(n)}\} \), which depends on the content of 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-sized message \( m_2 = \{m_{2,1}, \cdots, m_{2,q(n)}\} \), which depends on the content of 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-sized message \( s_i = \{s_{i,1}, \cdots, s_{i,q(n)}\} \), which depends on the content of 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-sized message \( m_{i+1} = \{m_{i+1,1}, \cdots, m_{i+1,q(n)}\} \), which depends on the content of 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.

This protocol may be thought of as a single server simulation of the multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other. 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-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other. In other words, malicious servers might transmit all user messages to a single server and calculate them alone on that server. 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. The multiple-server blind quantum computation protocol satisfies blindness to such attacks by malicious servers by assumption. 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-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other exists, so does a single-server blind quantum computation protocol with a classical user.

\[ \square \]

**Theorem 2** If a single-server blind quantum computation protocol with a classical user exists, then a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other can be constructed from a single-server blind quantum computation protocol with a classical user. Furthermore, if a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other exists, then a single-server blind quantum computation protocol with a classical user can be constructed from a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other.

**Proof** The proof is the same as in Theorem 1: If a single-server blind quantum computation protocol with a classical user exists, then a multiple-server blind quantum computation protocol with entanglement also exists.

We show that if there exists a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other, then there exists a single-server blind quantum computation protocol with a classical user. Assume a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other exists. 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 \( q(n) \). Note that \( m_{i,j} \) and \( s_{i,j} \) refer to messages in the multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other.

**Step 1. Send the first message to the server**
The user sends a classical polynomial-sized 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-sized message \( s_1 = \{s_{1,1}, \cdots, s_{1,q(n)}\} \), which depends on the content of 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-sized message \( m_2 = \{m_{2,1}, \cdots, m_{2,q(n)}\} \), which depends on the content of 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-sized message \( s_i = \{s_{i,1}, \cdots, s_{i,q(n)}\} \), which depends on the content of 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-sized message $m_{i+1} = \{m_{i+1,1}, \ldots, m_{i+1,q(n)}\}$, which depends on the content of 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.

A single server can also easily prepare entanglement, making such protocols feasible. This protocol can be interpreted as a simulation by a single server of the multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other. If the server is honest, this simulation can be performed by a single server since it is just a quantum computation. In the multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other, malicious servers can do classical/quantum communication during computation. However, we are not required to consider the quantum communication that the malicious servers do, because the quantum states that what each malicious server can prepare can also be prepared by other malicious servers. In other words, malicious servers might transmit all user messages to a single server and calculate them alone on the server. 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. The multiple-server blind quantum computation protocol satisfies blindness to such attacks by malicious servers by assumption. 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-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other exists, so does a single-server blind quantum computation protocol with a classical user.

\[\square\]

4 Discussion

In this research, we have defined a single-server blind quantum computation protocol with a classical user, a multiple-server without entanglement blind quantum computation protocol that allows servers to communicate freely with each other, and a multiple-server with entanglement blind quantum computation protocol that allows servers to communicate freely with each other, and have proved the equivalence of the existence of these protocols. It is not known if a single-server blind quantum computation protocol with a classical user exists [14–16]. 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.
Acknowledgements We would like to thank Takayuki Miyadera for the many helpful comments, and we are grateful to Kazuki Yamaga for his important advice. This work was supported by JST SPRING, Grant Number JPMJSP2110.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
2. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996). https://doi.org/10.1145/237814.237866
3. Abadi, M., Feigenbaum, J., Kilian, J.: On hiding information from an oracle. J. Comput. Syst. Sci. 39(1), 21–50 (1989). https://doi.org/10.1016/0022-0000(89)90018-4
4. Childs, A.M.: Secure assisted quantum computation. Quantum Inf. Comput. 5(6), 456–466 (2005). https://doi.org/10.26421/QIC5.6-4
5. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: Innovations in Computer Science—ICS 2010, pp. 453–469 (2010)
6. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: 2009 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 517–526 (2009). https://doi.org/10.1109/FOCS.2009.36
7. Fitzsimons, J., Kashefi, E.: Unconditionally verifiable blind computation. Phys. Rev. A 96, 012303 (2017). https://doi.org/10.1103/PhysRevA.96.012303
8. Morimae, T., Fujii, K.: Blind quantum computation protocol in which alice only makes measurements. Phys. Rev. A 87, 050301 (2013). https://doi.org/10.1103/PhysRevA.87.050301
9. Hayashi, M., Morimae, T.: Verifiable measurement-only blind quantum computing with stabilizer testing. Phys. Rev. Lett. 115, 220502 (2015). https://doi.org/10.1103/PhysRevLett.115.220502
10. Sano, Y.: Blind quantum computation using a circuit-based quantum computer. J. Phys. Soc. Jpn. 90(12), 124001 (2021). https://doi.org/10.7566/JPSJ.90.124001
11. Reichardt, B.W., Unger, F., Vazirani, U.: 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, pp. 321–322 (2013). https://doi.org/10.1145/2422436.2422473
12. McKague, M.: Interactive proofs for BQP via self-tested graph states. Theory Comput. 12(3), 1–42 (2016). https://doi.org/10.4086/toc.2016.v012a003
13. Sano, Y.: Multi-server blind quantum computation protocol with limited classical communication among servers. Quantum Inf. Process. 21, 88 (2022). https://doi.org/10.1007/s11128-022-03430-y
14. Fitzsimons, J.F.: Private quantum computation: an introduction to blind quantum computing and related protocols. NPJ Quantum Inf. 3(1), 1–11 (2017). https://doi.org/10.1038/s41534-017-0025-3
15. Morimae, T., Koshiba, T.: Impossibility of perfectly-secure one-round delegated quantum computing for classical client. Quantum Inf. Comput. 19(3–4), 214–221 (2019). https://doi.org/10.26421/qic19.3-4-2
16. Aaronson, S., Cojocaru, A., Gheorghiu, A., Kashefi, E.: Complexity-theoretic limitations on blind delegated quantum computation. In: 46th International Colloquium on Automata, Languages, and Programming vol. 132, pp. 6–1613 (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.