Impossibility of perfectly-secure delegated quantum computing for classical client

Tomoyuki Morimae[1,*] and Takeshi Koshiba[2,†]

[1] *ASRLD Unit, Gunma University, 1-5-1 Tenjin-cho Kiryu-shi Gunma-ken, 376-0052, Japan*

[2] *Graduate School of Science and Engineering, Saitama University,*
255 Shimo-Okubo, Sakura, Saitama 338-8570, Japan

(Dated: January 8, 2018)

Abstract

Blind quantum computing protocols enable a client who can generate or measure single-qubit states to delegate her quantum computing to a remote quantum server without leaking any privacy. Generations or measurements of single-qubit states are not too much burden for today's experimentalists. In other words, secure delegated quantum computing is possible for "almost classical" clients. However, is it possible for a "completely classical" client? Here we consider a protocol of perfectly-secure delegated quantum computing for a completely classical client, and show that the protocol cannot satisfy both the correctness (i.e., the correct result is obtained when the server is honest) and the blindness (i.e., the client's privacy is protected) simultaneously unless BQP is contained in NP.

[*] Electronic address: morimae@gunma-u.ac.jp

[†] Electronic address: koshiba@mail.saitama-u.ac.jp

I. INTRODUCTION

Imagine that Alice who does not have any sophisticated quantum technology wants to factor a large integer. She has a rich friend, Bob, who owns a full-fledged scalable quantum computer. Alice asks Bob to perform her quantum computing on his quantum computer. However, the problem is that Bob is not a reliable person, and therefore she does not want to reveal her input (the large integer), output (a prime factor), and the program (the fact that she is running Shor's algorithm) to Bob. Can she delegate her quantum computing to Bob while protecting her privacy?

Broadbent, Fitzsimons, and Kashefi [1] theoretically showed that such a secure delegated quantum computing is indeed possible if some minimum quantum technology is assumed for the client. (Proof-of-principle experiments were also done with photonic qubits [2–4].) In the protocol of Ref. [1] (Fig. 1), Alice, a client, has a device that emits randomly rotated single qubit states. She sends these states to Bob, the server, who has the full quantum technology. Alice and Bob are also connected with a two-way classical channel. Bob performs quantum computing by using qubits sent from Alice and classical messages exchanging with Alice via the classical channel. After finishing his quantum computation, Bob sends the output of his computation, which is a classical message, to Alice. This message encrypts the result of Alice's quantum computing, which is not accessible to Bob. Alice decrypts the message, and obtains the desired result of her quantum computing. (Ref. [1] also proposed a quantum input and quantum output protocol.) It was shown in Ref. [1] that whatever Bob does, he cannot learn anything about the input, the program, and the output of Alice's computation (except for some unavoidable leakage, such as upperbounds of the sizes of the input, output, and program, etc.). The composable security of the protocol was also shown in Ref. [5].

In the protocol, the client has to possess a device that generates single qubit states. Generations of single qubit states are ubiquitous in today's laboratories, and therefore not too much burden for the client. In other words, "almost classical" client can enjoy secure delegated quantum computing.

However, isn't it possible to realize secure delegated quantum computing for a "completely classical" client (Fig. 2)? Motivated by this question (and by other important questions such as the verifiability [6]), many variant protocols for blind quantum computing have been proposed [6–20]. For example, it was shown that, in stead of single-qubit states,

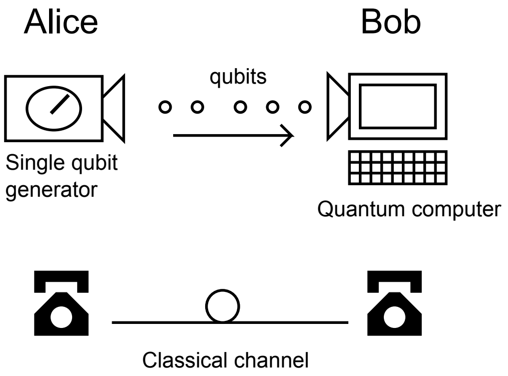

FIG. 1: The blind quantum computing protocol proposed in Ref. [1]. Alice possesses a device that emits randomly-rotated single-qubit states. Bob has a universal quantum computer. Alice and Bob share a two-way classical channel.

the client has only to generate weak coherent pulse states if we add more burden to the server [7]. Coherent states are considered as "more classical" than single-photon states, and therefore it enables secure delegated quantum computing for "more classical" client. It was also shown that secure delegated quantum computing is possible for a client who can only measure states [8, 9] (Fig. 3). A measurement of a bulk state with a threshold detector is sometimes much easier than the single-photon generation, and therefore the protocol also enables "more classical" client. However, these protocols still require the client to have some minimum quantum technologies, namely the generation of weak coherent pulses or measurements of quantum states. In fact, all protocols proposed so far require the client to have some minimum quantum abilities, such as generations, measurements, or routings of quantum states. (It is known that [1] if we have two quantum servers, a completely classical client can delegate her quantum computing. However, in this case, we have to assume that two servers cannot communicate with each other.)

In short, the problem of whether a perfectly-secure delegated quantum computing for a completely classical client is possible or not remains open. (Here, the perfect security means that an encrypted text gives no information about the plain text [21]. It is a typical security notion in the information theoretical security. Note that if we relax the requirement of the perfect security to a computational one, for example, there would be several ways of secure delegated quantum computing for a classical client. For example, the fully-homomorphic encryption scheme [22] might be able to achieve secure delegated quantum computing for a classical client.)

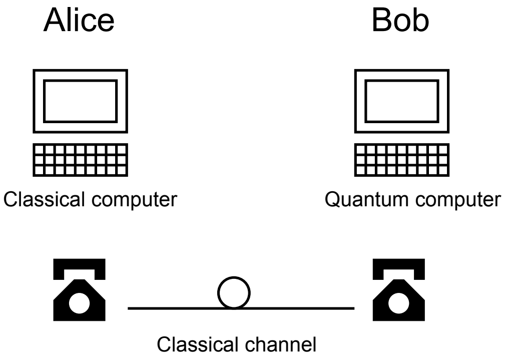

FIG. 2: The secure delegated quantum computing for a classical client. Alice has only a classical computer, whereas Bob has a universal quantum computer. Alice and Bob share a two-way classical channel.

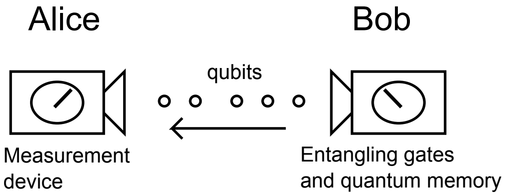

FIG. 3: The blind quantum computing protocol proposed in Refs. [8, 9]. Alice possesses a device that measure qubits. Bob has the ability of generating and storing entangled many-qubit states.

In this paper, we consider a protocol of perfectly-secure delegated quantum computing for a completely classical client, and show that the protocol cannot satisfy both the correctness (i.e., the correct result is obtained when the server is honest) and the blindness (i.e., the client's privacy is protected) simultaneously unless BQP $\subseteq$ NP.

II. PROTOCOL

In this section, we explain the protocol of perfectly-secure delegated quantum computing for a completely classical client. Let us assume that Alice has only a probabilistic polynomial-time Turing machine. Alice wants to solve a BQP problem. In other words, she wants to decide whether $x \in L$ or $x \notin L$ for an instance x of a language L in BQP. However, Alice cannot do it by herself (unless BQP = BPP), and therefore she delegates the computation to Bob as follows.

1. Alice generates a private key $k \in K$, where K is the set of valid keys. The key generation operation can be done in a polynomial time. She then encrypts L and x as $E_k(L, x)$, where E is the encryption operation, which is deterministic and in a polynomial time. She sends $E_k(L, x)$ to Bob.

2. Bob sends Alice 0 with probability $p_{Bob}(0|E_k(L, x))$ and 1 with probability $p_{Bob}(1|E_k(L, x)) = 1 - p_{Bob}(0|E_k(L, x))$.

3. Alice calculates the decrypting bit $d_k(L, x) \in \{0, 1\}$, which can be calculated deterministically and in a polynomial time. She accepts if and only if

$$d_k(L, x) \oplus (\text{the bit sent from Bob}) = 1.$$

When $d_k(L, x) = 0$, Bob has to send 1 to make Alice accept. On the other hand, if $d_k(L, x) = 1$, Bob has to send 0 to make Alice accept. In other words, Bob's bit has to be equal to $d_k(L, x) \oplus 1$ to make Alice accept. Therefore, for fixed L, x, and k, Alice's acceptance probability $p_{Alice}(acc|L, x, k)$ is

$$p_{Alice}(acc|L, x, k) = p_{Bob}(d_k(L, x) \oplus 1|E_k(L, x)).$$

We define the correctness and blindness as follows.

- Correctness: For any language $L \in$ BQP, instance x, private key $k \in K$, and polynomial $r \geq 2$, if $x \in L$ then

$$p_{Alice}(acc|L, x, k) \geq 1 - 2^{-r},$$

  while if $x \notin L$ then

$$p_{Alice}(acc|L, x, k) \leq 2^{-r}.$$

- Blindness: Bob cannot learn anything about Alice's (L, x) from $E_k(L, x)$.

III. PROOF

Now we show that if the protocol satisfies both the correctness and blindness simultaneously, then BQP $\subseteq$ NP. To show it, let L be a language in BQP. We show that the following NP protocol can verify L.

1. Merlin sends polynomial-length classical bit strings w and w_0 to Arthur. If Merlin is honest, w_0 is any private key from K, and w is a key from K that satisfies

   $$E_{w_0}(L_0, 0) = E_w(L, x), \tag{1}$$

   where

   $$L_0 \equiv \{x \in \{0,1\}^* | \text{the first bit of } x \text{ is } 0\}.$$

   Obviously, $0 \in L_0$ and $L_0 \in$ BQP. Note that such w always exists for any w_0, since otherwise Bob can learn that Alice's computation is not (L, x) when he receives $E_{w_0}(L_0, 0)$, which contradicts to the blindness.

2. Arthur checks whether w and w_0 are valid keys. We assume that the check can be done in a polynomial time. (Or, we assume that all bit strings are valid keys.) If at least one of them is non valid, Arthur rejects and aborts.

3. Arthur calculates $E_w(L, x)$ and $E_{w_0}(L_0, 0)$, which can be done deterministically and in a polynomial time. Arthur rejects and aborts if

   $$E_w(L, x) \neq E_{w_0}(L_0, 0).$$

4. Arthur calculates $d_w(L, x)$ and $d_{w_0}(L_0, 0)$, which can be done deterministically and in a polynomial time. Arthur accepts if and only if

   $$d_w(L, x) = d_{w_0}(L_0, 0).$$

We show that this NP protocol can verify L. Note that due to the correctness,

$$p_{Bob}(d_k(L_0, 0) \oplus 1 | E_k(L_0, 0)) \geq 1 - 2^{-r} \tag{2}$$

for any key $k \in K$ and any polynomial $r \geq 2$.

First let us consider the case of $x \in L$. In this case, due to the correctness,

$$p_{Bob}(d_k(L,x)\oplus 1|E_k(L,x)) \geq 1-2^{-r} \tag{3}$$

for any key $k \in K$ and any polynomial r. Furthermore, Arthur never rejects at steps 2 and 3. Finally, we can show $d_w(L,x)=d_{w_0}(L_0,0)$ and therefore Arthur accepts. In fact, if $d_w(L,x)\neq d_{w_0}(L_0,0)$, which means

$$d_{w_0}(L_0,0)=d_w(L,x)\oplus 1, \tag{4}$$

then

$$\begin{aligned}
1-2^{-r} &\leq p_{Bob}(d_{w_0}(L_0,0)\oplus 1|E_{w_0}(L_0,0)) \quad \text{(from Eq. (2))}\\
&= p_{Bob}(d_{w_0}(L_0,0)\oplus 1|E_w(L,x)) \quad \text{(from Eq. (1))}\\
&= p_{Bob}(d_w(L,x)|E_w(L,x)) \quad \text{(from Eq. (4))}\\
&= 1-p_{Bob}(d_w(L,x)\oplus 1|E_w(L,x))\\
&\leq 2^{-r} \quad \text{(from Eq. (3))},
\end{aligned}$$

which is a contradiction. Therefore, Arthur accepts when $x \in L$.

Next let us consider the case of $x \notin L$. In this case, due to the correctness,

$$p_{Bob}(d_k(L,x)\oplus 1|E_k(L,x)) \leq 2^{-r} \tag{5}$$

for any key $k \in K$ and any polynomial r. If Arthur arrives at step 4, w and w_0 are valid keys, and

$$E_w(L,x)=E_{w_0}(L_0,0) \tag{6}$$

is satisfied. Let us assume that

$$d_w(L,x)=d_{w_0}(L_0,0). \tag{7}$$

Then,

$$\begin{aligned}
1-2^{-r} &\leq p_{Bob}(d_{w_0}(L_0,0)\oplus 1|E_{w_0}(L_0,0)) \quad \text{(from Eq. (2))}\\
&= p_{Bob}(d_{w_0}(L_0,0)\oplus 1|E_w(L,x)) \quad \text{(from Eq. (6))}\\
&= p_{Bob}(d_w(L,x)\oplus 1|E_w(L,x)) \quad \text{(from Eq. (7))}\\
&\leq 2^{-r} \quad \text{(from Eq. (5))},
\end{aligned}$$

which is a contradiction. Therefore, $d_w(L,x)\neq d_{w_0}(L_0,0)$, which means that Arthur rejects. In summary, we have shown that L is in NP.

Acknowledgments

TM is supported by the Tenure Track System by MEXT Japan, Grant-in-Aid for Scientific Research on Innovative Areas No.15H00850 of MEXT Japan, and the Grant-in-Aid for Young Scientists (B) No.26730003 of JSPS. TK is supported by KAKENHI 26540002, 24106008, and 16H01705.

[1] A. Broadbent, J. F. Fitzsimons, and E. Kashefi, Proc. of the 50th Annual IEEE Sympo. on Found. of Comput. Sci. 517 (2009).

[2] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Science **335**, 303 (2012).

[3] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Nature Phys. **9**, 727 (2013).

[4] C. Greganti, M. C. Roehsner, S. Barz, T. Morimae, and P. Walther, New J. Phys. **18**, 013020 (2016).

[5] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, ASIACRYPT 2014, LNCS Volume 8874, pp.406-425 (2014).

[6] J. F. Fitzsimons and E. Kashefi, arXiv:1203.5217.

[7] V. Dunjko, E. Kashefi, and A. Leverrier, Phys. Rev. Lett. **108**, 200502 (2012).

[8] T. Morimae and K. Fujii, Phys. Rev. A **87**, 050301(R) (2013).

[9] M. Hayashi and T. Morimae, Phys. Rev. Lett. **115**, 220502 (2015).

[10] T. Morimae, V. Dunjko, and E. Kashefi, Quant. Inf. Comput. **15**, 0200 (2015).

[11] T. Morimae and K. Fujii, Nature Communications **3**, 1036 (2012).

[12] T. Morimae, Phys. Rev. Lett. **109**, 230502 (2012).

[13] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, Phys. Rev. Lett. **111**, 230501 (2013).

[14] A. Mantri, C. Pérez-Delgado, and J. F. Fitzsimons, Phys. Rev. Lett. **111**, 230502 (2013).

[15] Q. Li, W. H. Chan, C. Wu, and Z. Wen, Phys. Rev. A **89**, 040302(R) (2014).

[16] Y. B. Sheng and L. Zhou, Sci. Rep. **5**, 7815 (2015).

[17] Z. Sun, J. Yu, P. Wang, and L. Xu, Phys. Rev. A **91**, 052303 (2015).

[18] T. Sueki, T. Koshiba, and T. Morimae, Phys. Rev. A **87**, 060301(R) (2013).

[19] Y. Takeuchi, K. Fujii, T. Morimae, and N. Imoto, arXiv:1607.01568

[20] T. Morimae and K. Fujii, Phys. Rev. Lett. **111**, 020502 (2013).

[21] D. R. Stinson, *Cryptography: Theory and Practice*, (Chapman & Hall / CRC, 2006).

[22] C. Gentry, Fully homomorphic encryption using ideal lattices, Symposium on the Theory of Computing (STOC) pp.169 (2009).