Efficient Post Quantum Random Oblivious Transfer Based on Lattice

Lidong Xu and Mingqiang Wang[(✉)]

School of Mathematics, Shandong University, Jinan 250100, Shandong, China
xulidong@mail.sdu.edu.cn, wangmingqiang@sdu.edu.cn

Abstract. The large scale multiparty computation and private set intersection requires a number of oblivious transfer instances as subroutines, but the implementation of oblivious transfer protocols is relatively slow. An feasible way is to use the oblivious transfer variant called random oblivious transfer. In this paper, we propose a 1-out-of-2 random oblivious transfer protocol and extend it to a 1-out-of-k random oblivious transfer protocol based on the LWE assumption, quantum computation and measurement. Then, we analysis the stand-alone security of our 1-out-of-2 random oblivious transfer protocol under various malicious situations and prove its universally composable security in UC framework. As for the security of our 1-out-of-k random oblivious transfer protocol, the similar results can be obtained.

Keywords: Oblivious transfer · LWE problem · Quantum computation · UC-security

1 Introduction

Oblivious transfer (OT) is an important cryptographic primitive which can be used for designing secure multi-party computation (MPC) [1–3], bit commitment [4–6] and private set intersection (PSI) [7,8]. The OT protocol was firstly proposed, by Michael O. Rabin in 1981, to construct a secrets exchange scheme [9]. The original OT protocol has two participants, where one party (the sender) sends a message to another (the receiver) with the requirement that the receiver obtains this message with probability $\frac{1}{2}$ and the sender remains oblivious of whether the message has been received or not.

In order to build protocols for secure two-party computation, a more useful kind of OT protocol, called the 1-out-of-2 OT protocol, was developed [10–13]. In these protocols, the receiver is allowed to get one message from the sender's message pair without knowing anything about the other message, and the sender

Supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the National Key Research and Development Program of China (Grant No. 2018YFA0704702), and the National Natural Science Foundation of China (Grant No. 61832012).

L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 491–499, 2022.
https://doi.org/10.1007/978-3-031-19208-1_40

is required not to know about the receiver's choice. Another OT variant is the randomized oblivious transfer (ROT), the only difference from 1-out-of-2 OT lies in that the receiver is required to get one message randomly.

As is know, MPC protocols based on oblivious-circuit evaluation techniques require a large number of OT. Since the OT schemes are relatively slow, they become a major bottleneck for the large-scale MPC implementations. In order to deal with the problem of OT efficiency, Ishai et al. introduce the concept of OT extension [14] where one needs to use ROT instances as base OTs. In addition, the ROT scheme also is a main tool in designing efficient PSI protocols [8] which is one of the most popular MPC technique.

Motivated by the construction of trapdoor, claw free, 2-regular functions in [15–17], we propose a 1-out-of-2 ROT protocol based on quantum mechanics and LWE assumption. Then, we construct a family of trapdoor claw-free k-regular functions and extend the 1-out-of-2 ROT protocol to a 1-out-of-k ROT protocol. Furthermore, we analysis the stand-alone security of our ROT protocols under various malicious situations and prove their universally composable security in UC framework. The key technique of our protocol is to construct a family of trapdoor, claw free, k-regular function based on the LWE assumption. Another technique used in our protocol is quantum computation and quantum entanglement by which Bob can obtain only one of k preimages after measuring the produced quantum state.

2 The Construction of TCF k-Regular Functions

In this section, we will describe the construction of trapdoor claw-free (TCF) 2-regular functions defined in [17] and the construction of trapdoor claw-free k-regular functions, which are necessary for our ROT protocols. We start with the definition of trapdoor claw-free k-regular functions as follows:

Definition 1 *(Trapdoor claw-free k-regular). A deterministic function $f : D \to R$ is a trapdoor claw-free k-regular function if the following conditions hold:*

- *k-regular: $\forall y \in Im(f)$, we have $|f^{-1}(y)| = k$.*
- *collision resistance: It is impossible to find out any pair (x_0, x_1) such that $x_0 \neq x_1 \wedge f(x_0) = f(x_1)$ for any QPT algorithm without the trapdoor.*
- *Trapdoor one-way: Given $y \in Im(f)$ and the trapdoor t_f of the function f, there exists a QPT algorithm that can return the set $f^{-1}(y)$. Moreover, it is impossible to get any $x \in f^{-1}(y)$ for any QPT algorithm without the trapdoor.*

2.1 Requirements on Parameters

Let $\lambda \in \mathbb{Z}$ be the security parameter in the LWE problem, all other parameters be the functions of λ.

- $n = \lambda$, the length of vector $\mathbf{s}$;
- $q = poly(n)$, the prime modulus;

- $m \approx 2n \lg q$, the length of the error vector $\mathbf{e}$;
- $\alpha \in (0, 1)$, the discrete Gaussian distribution $\overline{\Phi}_\alpha$ is centered around 0 with standard deviation $\alpha q \geq 2\sqrt{n}$.

Under the setting of the parameters above, the LWE problem is at least as hard as solving SIVP [18,19]. And thus, the functions constructed in Sect. 3.2 and Sect. 3.3 are all trapdoor claw-free.

2.2 On the TCF 2-Regular Functions

In [17], the authors constructed a family $\mathcal{F}_2$ of TCF 2-regular functions based on the existence of a family $\mathcal{G}$ of injective, homomorphic, trapdoor one-way functions. For the completeness, we will recall the construction of $\mathcal{F}_2$ and related knowledge in this subsection.

The specific family $\mathcal{G}$ of injective, homomorphic, trapdoor one-way functions was constructed by Micciancio and Peikert [20]. Here, we list the outline of their construction and leave the detail to readers. First, generate a $n \times \bar{m}$ matrix A by randomly choosing its elements from $\mathbb{Z}_q$ and a $\bar{m} \times kn$ trapdoor matrix R by sampling its elements from a discrete Gaussian distribution $\mathcal{D}_{\alpha q}^{\bar{m} \times \omega}$ with mean 0 and standard deviation αq. Then, select a fixed matrix G as in [20] for which the function $g_G(s, e) = s^t G + e^t$ can be efficiently inverted, and construct the index matrix K by concatenating A and $G - AR$, i.e. $K = (A, G - AR)$. Finally, define the function g_K with trapdoor $t_K = R$, which forms the family $\mathcal{G}$, as follow:

$$g_K(s, e) = s^t K - e^t, \tag{1}$$

where $s \in \mathbb{Z}_q^n$ and $e \in L^m$, L is the domain of the errors in the LWE problem (the set of integers bounded in absolute value by μ).

Theorem 1 **([20]).** *The functions in $\mathcal{G}$ are injective, homomorphic, trapdoor one-way.*

2.3 The Construction of TCF *k*-Regular Functions

In order to design the $1 - k$ ROT protocol, we need to construct a family $\mathcal{F}_k$ of TCF k-regular functions. Motivated by the idea of constructing TCF 2-regular functions in Sect. 3.2, we construct the family $\mathcal{F}_k$ also by using the family $\mathcal{G}$ of homomorphic injective trapdoor one-way functions.

Let $g_K \in \mathcal{G}$ with trapdoor t_K, $x^i \in \mathcal{D} \setminus \{0\} (0 \leq i \leq k - 2)$ satisfying $x^i \neq x^j$ whenever $i \neq j$, we define the function $f : \mathcal{D} \times \mathbb{Z}_k \to \mathcal{R}$ with trapdoor $t_f = (t_K, x^0, ..., x^{k-1})$, which forms the family $\mathcal{F}_k$, as follows:

$$f(x, c) = \begin{cases} g_K(x), & \text{if c = 0;} \\ g_K(x) + g_K(x^0), & \text{if c = 1;} \\ g_K(x) + g_K(x^1), & \text{if c = 2;} \\ ... & \\ g_K(x) + g_K(x^{k-2}), & \text{if c = k-1.} \end{cases} \tag{2}$$

In a similar way as proving the functions in $\mathcal{F}_2$ are TCF 2-regular in [17], we can prove that the functions in $\mathcal{F}_k$ constructed above is TCF k-regular.

Theorem 2. *The functions in the family $\mathcal{F}_k$ are trapdoor claw-free k-regular.*

3 Our $1 - k$ ROT Protocols

In this section, we will present a $1 - 2$ ROT protocol by using the family $\mathcal{F}_2$ of TCF 2-regular functions in [17], and extend this protocol into a $1 - k$ ROT protocol by using the family $\mathcal{F}_k$ of TCF k-regular functions constructed in Sect. 3. As in [17], for $k \geq 2$, we employ the family $\mathcal{F}_k$ of TCF k-regular functions in a convenient form as $\mathcal{F}_k = \{f : \{0,1\}^n \to \{0,1\}^m\}$, where the domain of each f is also denoted by D.

3.1 The $1 - 2$ ROT Protocol

In the prepare stage, first choosing a fixed function f and its trapdoor t_f from the family $\mathcal{F}_2 = \{f : \{0,1\}^n \to \{0,1\}^m\}$ of TCF 2-regular functions. Then, giving (f, t_f) to the sender Alice and f to the receiver Bob. To transfer the two messages $b_1, b_2 \in \{0,1\}^m$ from Alice to Bob obliviously, our $1-2$ ROT protocol performs as follows:

1. Bob prepares his registers at $\frac{1}{\sqrt{|D|}} \sum_{x \in D} (|x\rangle \otimes |0\rangle)$.
2. Bob applies the operator U_f by using the first register as control and the second one as target, and the state in the two registers is in the form of $\frac{1}{\sqrt{|D|}} \sum_{x \in D} |x\rangle|f(x)\rangle$. After that, Bob sends the second register to Alice.
3. Alice measures her register in the computational basis and obtains the outcome y. Then, Bob's register becomes $\frac{1}{\sqrt{2}}(|x_1\rangle + |x_2\rangle)$, where $f(x_1) = f(x_2) = y$. Bob measures his register in the computational basis and obtains the outcome $\widetilde{x}$ $(= x_1$ or $x_2)$.
4. Alice computes the preimages x_1, x_2 of y by using the trapdoor t_f. Then, she sends the pairs $(a_1 = b_1 \oplus x_1, h(x_1))$ and $(a_2 = b_2 \oplus x_2, h(x_2))$ to Bob, where $h(x)$ represents the last bit of x.
5. Bob computes the value of $f(a_1 \oplus a_2 \oplus \widetilde{x})$. If the result is y (which means $b_1 = b_2$), then he terminates this protocol.
6. Bob gets the message b_σ by computing $a_\sigma \oplus \widetilde{x}$ if $h(x_\sigma) = h(\widetilde{x})$ $(\sigma = 1$ or $2)$.

3.2 The $1 - k$ ROT Protocol

To extend the protocol above into the general $1 - k$ ROT protocol, we only need to substitute the TCF 2-regular function for a TCF k-regular function constructed in Sect. 3.3.

In the prepare stage, first choosing a fixed function f and its trapdoor t_f from the family $\mathcal{F}_k = \{f : \{0,1\}^n \to \{0,1\}^m\}$ of TCF k-regular functions. Then, giving (f, t_f) to the sender Alice and f to the receiver Bob. To transfer the k messages $b_1, b_2, ..., b_k \in \{0,1\}^m$ from Alice to Bob obliviously, our $1 - k$ O.T. protocol performs as follows:

1. Bob prepares his registers at $\frac{1}{\sqrt{|D|}}\sum_{x\in D}(|x\rangle\otimes|0\rangle)$.
2. Bob applies the operator U_f by using the first register as control and the second one as target, and the state in the two registers is in the form of $\frac{1}{\sqrt{|D|}}\sum_{x\in D}|x\rangle|f(x)\rangle$. After that, Bob sends the second register to Alice.
3. Alice measures her register in the computational basis and obtains the outcome y. Then, Bob's register becomes $\frac{1}{\sqrt{k}}(|x_1\rangle+...+|x_k\rangle)$ where $f(x_1)=...=f(x_k)=y$. Bob measures his register in the computational basis and obtains the outcome $\widetilde{x}(\in\{x_1,x_2,...,x_k\})$.
4. Alice computes the preimages $x_1,...,x_k$ of y by using the trapdoor t_f. Then, she sends the pairs $(a_i=b_i\oplus x_i, h(x_i))(1\leq i\leq k)$ to Bob, where $h(x)$ presents the last $\lfloor\log k\rfloor$ bits of x.
5. Bob computes the value of $f(a_i\oplus a_j\oplus\widetilde{x})(1\leq i<j\leq k)$. If some $f(a_i\oplus a_j\oplus\widetilde{x})=y$ (which means $b_i=b_j$), then he terminates this protocol.
6. Bob gets the message b_σ by computing $a_\sigma\oplus\widetilde{x}$ if $h(x_\sigma)=h(\widetilde{x})$ where $\sigma\in\{1,2,...,k\}$.

3.3 The Security Analysis of Our $1-2$ ROT Protocol

In this section, we will consider the stand-alone security of our $1-2$ ROT protocol in two aspects, Bob's malicious operation and Alice's malicious operation. The extended version, $1-k$ ROT protocol, can be analysed in the same way. Let us first recall the following property of the family $\mathcal{F}_2$ described in Sect. 3.2, on which the security of our $1-2$ ROT protocol is based.

Theorem 3 *[17]. The functions in the family $\mathcal{F}_2$ described in Sect. 3.2 are TCF 2-regular.*

Bob's Malicious Strategy. A malicious receiver Bob aims to get both two messages b_1 and b_2 from Alice. To achieve his aim, Bob has to find a method to get the collision x' for his measurement outcome $\widetilde{x}$ in Step 3. Except for guessing x', what he could do is computing $y=f(\widetilde{x})$, and managing to find the preimages of y with respect to f. But, the function f is one-way according to Theorem 3, and thus Bob cannot obtain the preimages of y by inverting f. So, it is impossible that Bob have an efficient method to get both b_1 and b_2.

Alice's Malicious Strategy. A malicious sender Alice wants to know what message Bob gets from the transfer procedure. There are two ways for Alice to achieve her aim, one is to get Bob's measurement outcome $\widetilde{x}$ and another is to cheat by sending illegal information to Bob in Step 4.

Note that, Alice gets y by measuring her register and Bob obtains $\widetilde{x}$ by measuring his register with the superposition state $\frac{1}{\sqrt{2}}(|x_1\rangle+|x_2\rangle)$ in Step 3. Although Alice can computes the preimages x_1 and x_2 of y by the trapdoor t_f in Step 5, and $\widetilde{x}$ must be one of x_1 and x_2, Alice has no way to determine which one $\widetilde{x}$ is. So, the first way is not possible.

As for the second way, Alice may send two pairs $(a_1 = b_1 \oplus w_1, h(w_1))$ and $(a_2 = b_2 \oplus x_2, h(x_2))$ with $b_1 = b_2$ to Bob in Step 4. If Bob does not verify whether the two pairs are legal, he will always get b_1 in Step 6, no matter what his measurement outcome $\widetilde{x}$ is. And thus, Alice can know what the message Bob obtains. But in Step 5, Bob verifies the reality of the two pairs from Alice by computing the value of $f(a_1 \oplus a_2 \oplus \widetilde{x})$. If the result is y, then Bob infers that b_1 and b_2 are the same, and terminates the protocol. Therefore, this strategy also does not work.

4 The UC-security of Our $1-2$ ROT Protocol

In this section, we will prove the universally composable security of our $1-2$ ROT protocol in the UC framework. As for our $1-k$ ROT protocol, its UC-security can be proven in the same way.

We work in the standard universal composability framework of Canetti [21] with static corruption of some parties. The ideal world execution involves dummy parties (some of whom may be corrupted by an ideal adversary) interacting with the functionality $\mathcal{F}$. The dummy parties only relay the inputs to $\mathcal{F}$, and relay the outputs of $\mathcal{F}$ to the calling machine. The real world execution involves parties (some of whom may be corrupted by a real world adversary) interacting only with each other.

For our $1-2$ ROT protocol interacting with an adversary, the functionality $\mathcal{F}_{ROT}$ interacting with the simulator in the ideal world is defined as follows:

Functionality $\mathcal{F}_{ROT}$

Parameters: String length n.
Parties: The sender Alice and the receiver Bob.

1. Upon receiving the message b_0, b_1 from Alice and activated by Bob, $\mathcal{F}_{ROT}$ outputs b_σ to Bob randomly

Fig. 1. The functionality $\mathcal{F}_{ROT}$

Let $\mathcal{A}$ be a static adversary that interacts with the parties Alice and Bob running the $1-2$ ROT protocol, we now construct a simulator $\mathcal{S}$ in ideal world interacting with the ideal functionality $\mathcal{F}_{ROT}$, such that no environment $\mathcal{Z}$ can distinguish the interaction with $\mathcal{A}$ in the real world from the interaction with $\mathcal{S}$ in the ideal world. The simulator $\mathcal{S}$ starts by invoking a copy of $\mathcal{A}$ and runs a simulated interaction of $\mathcal{A}$ with $\mathcal{Z}$ and the parties Alice and Bob. More specifically, the simulator $\mathcal{S}$ works as follows:

Simulating the communication with $\mathcal{Z}$: Every input value that $\mathcal{S}$ receives from $\mathcal{Z}$ is written on the adversary $\mathcal{A}$'s input tape (as if coming from $\mathcal{A}$'s environment). Every output value written by $\mathcal{A}$ on its output tape is copied to $\mathcal{S}$'s output tape (to be read by the environment $\mathcal{Z}$).

Simulating the case when only Alice is corrupted: The simulator $\mathcal{S}$ randomly selects a function f with its trapdoor t_f from the family $\mathcal{F}_2$ of TCF 2-regular functions, and sends (f, t_f) to Alice and f to Bob respectively.
When $\mathcal{A}$ produces (a_1, w_1) and (a_2, w_2) with $w_1 \neq w_2$ for honest Bob, $\mathcal{S}$ randomly chooses some $\widetilde{x} \in D$. Then, $\mathcal{S}$ computes $y = f(\widetilde{x})$ and another preimage $\widetilde{x'}$ of y by the trapdoor t_f. After that, $\mathcal{S}$ computes $b_1 = a_1 \oplus \widetilde{x}$ and $b_2 = a_2 \oplus \widetilde{x'}$ where $h(w_1) = h(\widetilde{x}), h(w_2) = h(\widetilde{x'})$ and stores them. When dummy Bob is activated, $\mathcal{S}$ sends b_1 and b_2 to $\mathcal{F}_{ROT}$. When $\mathcal{F}_{ROT}$ returns b_σ, $\mathcal{S}$ outputs it as if from Bob.
Simulating the case when only Bob is corrupted: The simulator $\mathcal{S}$ randomly selects a function f and its trapdoor t_k from the family $\mathcal{F}_2$ of TCF 2-regular functions, and sends (f, t_f) to Alice and f to Bob respectively.
When the dummy Alice is activated, $\mathcal{S}$ gets b_σ from the functionality $\mathcal{F}_{ROT}$ and stores it. When $\mathcal{A}$ is activated, $\mathcal{S}$ outputs b_σ as if from Bob.
Simulating the remaining cases: When both parties are corrupted, the simulator just runs $\mathcal{A}$ internally (who itself generates the messages from both Alice and Bob). When neither party is corrupted, there is no necessity to construct $\mathcal{S}$.
According to the above models of different corrupted cases, we obtain the following two propositions. And thus, our $1-2$ ROT protocol possesses the UC-security.

Proposition 1. *If the adversary $\mathcal{A}$ corrupts Alice in an execution of our $1-2$ ROT protocol π, then we have*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

Proposition 2. *If the adversary $\mathcal{A}$ corrupts Bob in an execution of our $1-2$ ROT protocol π, then we have*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

Theorem 4. *Denote our $1-2$ ROT protocol as π, then*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

Thus, π UC-emulates the ideal function $\mathcal{F}_{ROT}$, in other word, π is UC-secure.

5 Conclusion

Motivated by the construction of trapdoor claw-free 2-regular functions in [17], we propose a $1-2$ ROT protocol and construct a family of trapdoor claw-free k-regular functions based on which we extend the $1-2$ ROT protocol to the $1-k$ ROT protocol. In our protocols, the key techniques are quantum computation and the family of trapdoor, claw free, k-regular functions. Furthermore, We analysis the stand-alone security of our $1-2$ ROT protocol in various malicious situations and prove its composable security in the UC framework. Certainly, the security of our $1-k$ ROT protocol can be obtained by a similar discussion.

Comparing with other OT protocols, our $1-2$ ROT protocol possesses stronger security and needs fewer rounds between the sender and the receiver. We give an intuitional comparison between our $1-2$ ROT protocol and the others presented before in the following table:

Table 1. Comparison with other OT (ROT) protocols

Protocol	Round (moves)	Security
OT in [22]	5 (including 2 with functionality)	UC-secure
OT in [23]	6	Non proof
OT in [24]	$O(\log n)$	FullSim
ROT in [3]	5 (including 2 with functionality)	UC-secure
Our ROT	3	UC-secure

References

1. Yao, A.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science, pp. 162–167 (1986)
2. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing, pp. 218–229 (1987)
3. Costa, B., Branco, P., Goulao, M., Lemus, M., Mateus, P.: Randomized oblivious transfer for secure multiparty computation. Entropy **23**, 1001 (2021)
4. Yang, W., Huang, L.S., Wang, Q.Y., Luo, Y.L.: Quantum bit commitment based on qubit oblivious transfer. Chin. J. Electron. **18**(3), 422–426 (2009)
5. Yang, L.: Bit commitment protocol based on random oblivious transfer via quantum channel. arXiv: 1306.5863 (2013)
6. Song, Y.Q., Yang, L.: Practical quantum bit commitment protocol based on quantum oblivious transfer. Appl. Sci. **8**, 1990 (2018)
7. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: SpOT-light: lightweight private set intersection from sparse OT extension. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 401–431. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_13
8. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: SpOT-Light: lightweight private set intersection from sparse OT extension. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 401–431. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_13
9. Rabin, M.O.: How to Exchange Secrets by Oblivious Transfer. Technical Memo TR-81 (1981)
10. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8
11. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_33

12. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_16
13. Jannati, H., Bahrak, B.: An oblivious transfer protocol based on elgamal encryption for preserving location privacy. Wireless Pers. Commun. **97**(2), 3113–3123 (2017). https://doi.org/10.1007/s11277-017-4664-7
14. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_9
15. Mahadev, U.: Classical Homomorphic Encryption for Quantum Circuits. SIAM J. Comput. 189 (2020)
16. Mahadev, U.: Classical Verification of Quantum Computations. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pp. 259–267 (2018)
17. Alexandru, C., Léo, C., Elham, K., Petros, W.: On the possibility of classical client blind quantum computing. Cryptography **5**(1), 3 (2021)
18. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: ACM Symposium on Theory of Computing, 84–93 (2005)
19. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_2
20. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
21. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: IEEE Symposium on Foundations of Computer Science, p. 136 (2001)
22. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
23. Wang, F.H., Hu, P.Y., Liu, Z.H.: Lattice-based oblivious transfer protocol. J. Commun. **32**(3), 125–130 (2011)
24. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Adaptive oblivious transfer with access control from lattice assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 533–563. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_19