entropy

MDPI

Article

Randomized Oblivious Transfer for Secure Multiparty Computation in the Quantum Setting

Bruno Costa [1,2], Pedro Branco [1,3], Manuel Goulão [1,3], Mariano Lemus [1] and Paulo Mateus [1,3,*]

[1] Departamento de Matemática, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisbon, Portugal; brunofilipe.antunescosta@capgemini.com (B.C.); pedrodemelobranco@gmail.com (P.B.); manuel.goulao@tecnico.ulisboa.pt (M.G.); marianojlemush@gmail.com (M.L.)

[2] Capgemini Engineering, Av. D. João II, Lote 1.07.2.1, Piso 2, 1990-096 Lisbon, Portugal

[3] Instituto de Telecomunicações, IST Av. Rovisco Pais, 1049-001 Lisbon, Portugal

[*] Correspondence: pmat@math.tecnico.ulisboa.pt

Abstract: Secure computation is a powerful cryptographic tool that encompasses the evaluation of any multivariate function with arbitrary inputs from mutually distrusting parties. The oblivious transfer primitive serves is a basic building block for the general task of secure multi-party computation. Therefore, analyzing the security in the universal composability framework becomes mandatory when dealing with multi-party computation protocols composed of oblivious transfer subroutines. Furthermore, since the required number of oblivious transfer instances scales with the size of the circuits, oblivious transfer remains as a bottleneck for large-scale multi-party computation implementations. Techniques that allow one to extend a small number of oblivious transfers into a larger one in an efficient way make use of the oblivious transfer variant called randomized oblivious transfer. In this work, we present randomized versions of two known oblivious transfer protocols, one quantum and another post-quantum with ring learning with an error assumption. We then prove their security in the quantum universal composability framework, in a common reference string model.

Keywords: oblivious transfer; quantum cryptography; post-quantum cryptography; universal composability

check for updates

Citation: Costa, B.; Branco, P.; Goulão, M.; Lemus, M.; Mateus, P. Randomized Oblivious Transfer for Secure Multiparty Computation in the Quantum Setting. *Entropy* **2021**, *23*, 1001. https://doi.org/10.3390/e23081001

Academic Editor: Ivan B. Djordjevic

Received: 14 June 2021
Accepted: 22 July 2021
Published: 31 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Oblivious transfer (OT), first introduced by Rabin in 1981 [1], is an important primitive in modern cryptography. The OT primitive is known to be a basic building block for other cryptographic tasks, including secure Multi-Party Computation (MPC), Bit Commitment (BC), Coin-Tossing, and Zero-Knowledge Proofs [2–7].

A 1-out-of-2 OT protocol [8] consists of two parties, a sender with two input messages (m_0, m_1) and a receiver with a choice bit $b \in \{0, 1\}$. The goal of the protocol is to output only the message m_b to the receiver, with no information about m_{1-b}, and the sender remains oblivious to the receiver's input bit b. Note that, in the original work by Rabin, called all-or-nothing OT [1], the sender has a single input message, while the receiver has none. The protocol outputs the message to the receiver with probability $\frac{1}{2}$, such that the receiver has no information whether or not the receiver obtained the message. It was shown that one can construct 1-out-of-2 OT from all-or-nothing OT [9]. Another OT variant is that of Randomized Oblivious Transfer (ROT), where neither of the parties have any inputs. The ROT protocol, instead, outputs the messages (m_0, m_1) to the sender and (b, m_b) to the receiver, with (m_0, m_1, b) chosen uniformly at random from their domains.

MPC [10,11], which is an extremely useful cryptographic tool to compute arbitrary functionalities, can be reduced to the OT primitive; i.e., having access to a secure OT is sufficient [2]. MPC implementations based on oblivious-circuit evaluation techniques require a large number of OT (one per input wire for Yao [10], and one per AND gate for

GMW [11]). Since classical OT schemes (being based on asymmetric-key cryptography) are relatively slow, the development of large-scale MPC implementations has been severely hindered by the required OT rates. In order to deal with this issue of OT efficiency, the concept of OT extension was introduced by Ishai et al. in 2003 [12]. This technique refers to extending a small number of computationally expensive *base* OTs into a larger number of OTs, using only cheap symmetric cryptography primitives. For proving the security of these OT extension techniques in the malicious-adversary setting [13], it turns out that one is required to use ROT instances as the base OTs. Additionally, ROT finds direct application in designing efficient Private Set Intersection (PSI) protocols [14], one of the most popular MPC techniques.

Moreover, even though the efficiency issue can be solved by the use of OT extensions for MPC applications, there is the underlying threat that asymmetric-key based schemes (e.g., integer-factorization or discrete-logarithm problems) will be faced with the arrival of quantum computers [15]. The research initiatives for developing quantum-resistant solutions have been following two paths. The first being on the development of more hard-to-break classical cryptography algorithms that will remain secure even against a quantum adversary. These solutions include the approximate Shortest Vector Problem (SVP) on ideal lattices [16], the Learning with Errors (LWE) problem [17] and its ring version, Ring Learning with Errors (RLWE) [16], constituting a new area of research, called post-quantum cryptography. The second approach is that of quantum cryptography, where solutions for Quantum Key Distribution (QKD), BC, and OT already exist [18]. While unconditional security for QKD has been proven [19], there are impossibility results to achieve for the case of BC and OT [20–22]. Nevertheless, practical solutions for BC and OT were proposed under the assumption of physical limitations on the devices, such as noisy storage and bounded quantum memories [23–27].

Our Contribution

In this work, we explore the construction of two ROT protocols in the quantum Universal Composability (UC) framework, in the Common Reference String (CRS) model:

- A quantum protocol based on the UC construction by Unruh [28] and augmented with an additional subroutine to enforce randomized outputs.
- A classical protocol based on a variant of the RLWE assumption that adapts the one presented in [29,30] but does not require a random oracle model and, instead, uses a composable commitment scheme and a composable non-interactive zero knowledge (NIZK) protocol.

In both cases, the basic idea is to build upon existing non-randomized OT protocols in such a way as to force the values of all of the protocol's outputs to be influenced by both parties. This allows us to randomize both the messages m_0, m_1 and the choice bit b as long as at least one party is honest, leading to a ROT protocol. Furthermore, we prove that the resulting protocols are secure in the quantum UC framework.

This paper is organized in five sections. In Section 2, we briefly review some definitions and functionalities relevant for the description and analysis of the protocols. In Section 3, we present the generic construction of ROT from OT and afterwards present the commitment scheme and OT protocols that we will be using to achieve the quantum security we need. The security of the protocols are then shown in Section 4. Finally, in Section 5, we present the main results of this work.

2. Background

The problems regarding Ring Learning with Errors are conjectured to be hard on both classical and quantum computers. Before defining the RLWE distribution and its decision problem, we first present the notation used. Let $R_q = \mathbb{Z}_q[X]/f(X)$ be a ring, where $q > 2$ is a prime, and $f(X)$ is a cyclotomic polynomial of degree n. Let $\beta \in \mathbb{N}$ and χ be the error distribution that outputs elements of R_q with a norm greater than β with negligible probability.

Definition 1 (RLWE distribution)**.** *Let q, R_q and χ be as above. The RLWE distribution $A_{s,\chi}$ is obtained by sampling $a \in R_q$ uniformly, choosing $e \leftarrow_{\$} \chi$ and outputting $(a, b = as + e \mod q)$ for a secret $s \in R_q$.*

Definition 2 (decision-RLWE)**.** *Let q, R_q, χ and $A_{s,\chi}$ be as above. For $s \leftarrow_{\$} R_q$, given many polynomial samples, the goal is to distinguish between $A_{s,\chi}$ and a uniform distribution over $R_q \times R_q$.*

By using the the RLWE variant of the LWE problem we are able to not only work with smaller keys but also increase the speed of the operations by using the Number Theoretic Transform (NTT). The protocol we will be analyzing uses a variant of the RLWE problem, the Hermite Normal Form of the RLWE problem (HNF-RLWE), in which the secret s is sampled from the error distribution χ instead of being chosen uniformly at random from the ring R_q. This version of the problem is assumed to be hard as well, since RLWE reduces to it [31].

Often times studying the standalone security of protocols is not enough, since they will be frequently used as subroutines in more complex tasks, as is the case of OT, as well as Coin Tossing, Commitment schemes, Zero-Knowledge proofs, etc. In order to ensure that protocols are secure in any computational environment, Canetti [32] introduced the Universal Composability (UC) framework, which we define next.

Let π be an n-party protocol and $\mathcal{F}$ be an ideal functionality. We denote as $\text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ the output of the environment $\mathcal{Z}$ at the end of the ideal-world execution of functionality $\mathcal{F}$ with adversary $\mathcal{S}$, and as $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$ the output of the environment $\mathcal{Z}$ at the end of the real-world execution of π with adversary $\mathcal{A}$. The notion of a protocol securely emulating some ideal functionality is as follows:

Definition 3 (UC-secure)**.** *We say that π UC-emulates $\mathcal{F}$ if for any adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$, such that, for all environment $\mathcal{Z}$,*

$$IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}} \approx EXEC_{\pi,\mathcal{A},\mathcal{Z}}.$$

When discussing UC security, we can consider either a bounded (computational) or unbounded (statistical) approach. In computational UC security, we restrict the adversary, simulator, and environment to polynomial-time machines, and this approach is used when showing security based on computational assumptions. On the other hand, in statistical UC security, we quantify over all adversaries, simulators, and environments; as such, we can model statistical security.

In this work, we consider *malicious* adversaries, that is, adversaries that can deviate in any way from the protocol. However, we assume that the corruption of a party happens before the start of the protocol, and both the sender or the receiver may be corrupted.

In Figures 1–5 we present the functionalities that will be relevant in this work.

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

Parameters: String size ℓ.

Parties: The sender S and the receiver R.

1. Upon receiving inputs $(m_0, m_1) \in \{0,1\}^\ell \times \{0,1\}^\ell$ from S and $b \in \{0,1\}$ from R, $\mathcal{F}_{OT}$ sends m_b to R.

Figure 1. OT functionality .

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

Parameters: String size ℓ.

Parties: The sender S and the receiver R.

1. Upon receiving message START from both S and R, $\mathcal{F}_{ROT}$ samples $m_0, m_1 \overset{\$}{\leftarrow} \{0,1\}^\ell$ and $b \overset{\$}{\leftarrow} \{0,1\}$. It then sends (m_0, m_1) to S and (b, m_b) to R.

Figure 2. ROT functionality.

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

Parameters: Commitment size ℓ (for bit commitment, $\ell = 1$).

Parties: The sender S and the recipient R.

1. Upon input (COMMIT, x) with $x \in \{0,1\}^\ell$ from S, $\mathcal{F}_{COM}$ records x and sends a receipt to R.
2. Upon input OPEN from S, send (OPEN, x) to R.

Figure 3. Commitment functionality.

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

Parameters: Distribution $\mathcal{D}$.

1. When activated for the first time on input VALUE, $\mathcal{F}^{\mathcal{D}}_{CRS}$ chooses a value $d \overset{\$}{\leftarrow} \mathcal{D}$ and sends d back to the activating party. Every other activation will return the same d to the activating party.

Figure 4. Common Reference String functionality.

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

Parameters: Common statement x.

Parties: The verifier V and the prover P.

- **Proof:** On input (x, w) from P, if $\mathcal{R}(x, w) = 1$, then send $p(w)$ to P.
- **Verification:** On input $(x, p(w))$ from V, send $\mathcal{R}(x, w)$ to V.

Figure 5. Non-Interactive Zero-Knowledge functionality.

We stress that the definition of $\mathcal{F}_{ROT}$ presented here is stronger than the one presented in Unruh's original paper [28], in which the outputs are only random if the parties are both honest. In the same paper, the UC framework is extended to the quantum setting by allowing the protocol π, the adversary $\mathcal{A}$, the simulator $\mathcal{S}$, and the environment $\mathcal{Z}$ to be quantum.

Unruh [28] also showed that, when π is a classical protocol and π statistically UC-emulates $\mathcal{F}$, then π statistically quantum-UC-emulates $\mathcal{F}$, providing a lift from statistical classical-UC to statistical quantum-UC. A similar result exists for the computational case [28], but it is required that the adversary in the classical case is given the same computational power as in the quantum setting; in other words, we need to guarantee that the classical machines present in the proof of UC security are as powerful as quantum-polynomial-time machines.

Consider protocols π and σ, we denote the protocol where σ invokes instances of π by σ^π. A usual situation would be $\sigma^{\mathcal{F}}$, being a protocol that uses some ideal functionality $\mathcal{F}$, and σ^π would then be the protocol that results from implementing that functionality with some protocol π. Composition has been shown to be secure, both in the classical [32] and quantum settings [28].

Theorem 1 (Universal Composition Theorem [28])**.** *Let $\mathcal{F}, \mathcal{G}$ be ideal functionalities. Let π be an n-party protocol that UC-emulates $\mathcal{G}$ in the $\mathcal{F}$-hybrid model, and let η be an n-party protocol that UC-emulates $\mathcal{F}$. Protocol π^{η} then UC-emulates $\mathcal{G}$.*

3. Protocols

In this section, we start by presenting the generic construction of ROT from OT, using a commitment scheme, and afterwards describe the commitment scheme and the quantum OT protocol that will allow our ROT protocol to computationally quantum-UC-emulate $\mathcal{F}_{ROT}$. Finally, we describe a post-quantum approach, a ROT protocol based on the RLWE assumption, inspired by the recent work of [30], with a small tweak to avoid using random oracles, which misbehave against quantum adversaries.

3.1. Generating an UC-Secure Random OT

The protocol $\pi_{OT \to ROT}$ is presented in Figure 6. We consider the two parties: the sender S and the receiver R. It begins with R sampling two strings $r_0, r_1 \in \{0,1\}^{\ell}$ and committing them to S. R then chooses a random bit c, and S chooses two random strings, $w_0, w_1 \in \{0,1\}^{\ell}$. With these, the parties invoke the $\mathcal{F}_{OT}$ functionality. Following that, S chooses a random bit d and sends it over to R. Finally, R opens his commitment, and S checks if it matches the initial commit. If it does not, it aborts; otherwise, it outputs $(M_0 = w_d \oplus r_d,\ M_1 = w_{d\oplus 1} \oplus r_{d\oplus 1})$. R outputs $(b = c \oplus d,\ M_b = w_c \oplus r_c)$.

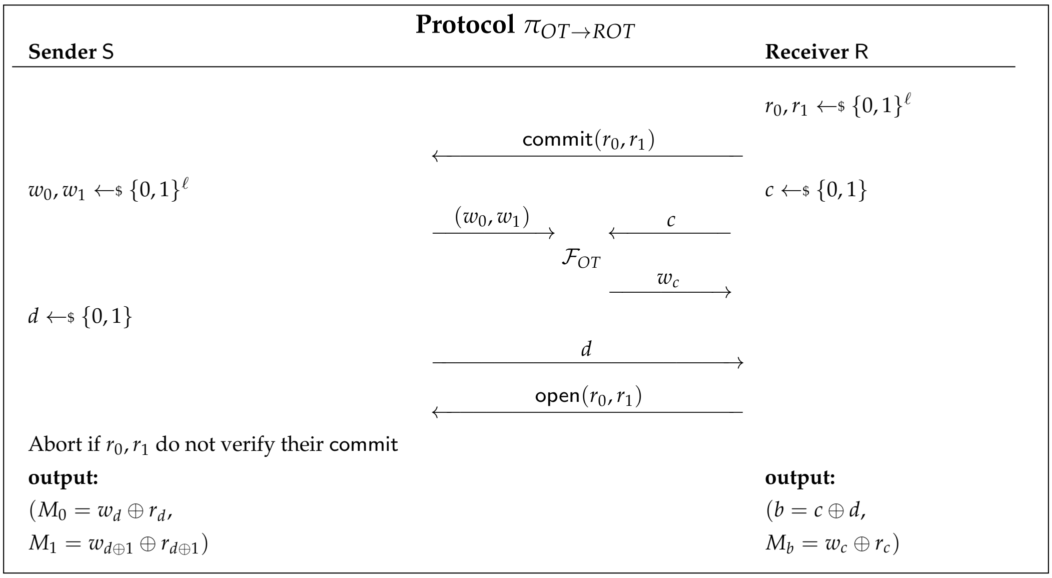
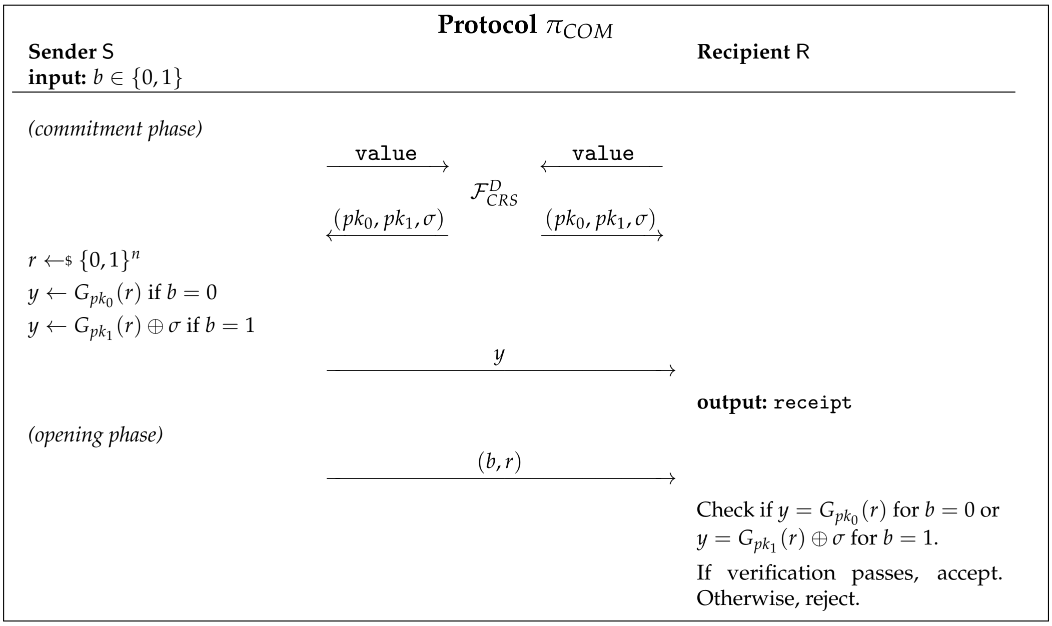


Figure 6. ROT protocol based on secure commitments.

3.2. UC-Secure Commitment Scheme

Canetti [33] showed that UC-secure commitment schemes are impossible in the plain model, and the same result was later proven for the quantum setting as well [22]. With that in mind, we will be working on the Common Reference String (CRS) model defined in Figure 4.

The protocol π_{COM} in Figure 7 has been shown to be computationally UC-secure in the CRS model [33]. The key to this protocol's composability is the use of a trapdoor pseudo-random generator (PRNG) G_{pk}, which is described by its public key pk. This generator G_{pk} stretches n-bit inputs to $4n$-bit outputs, and has a trapdoor td. Having access to both pk and td, we can easily check if a given string $y \in \{0,1\}^{4n}$ is in the range of G_{pk}.



Protocol π_{COM}

Sender S
input: $b \in \{0,1\}$

Recipient R

(commitment phase)

$$\xrightarrow{\texttt{value}} \mathcal{F}^{D}_{CRS} \xleftarrow{\texttt{value}}$$

$$\xleftarrow{(pk_0, pk_1, \sigma)} \quad \xrightarrow{(pk_0, pk_1, \sigma)}$$

$r \leftarrow_{\$} \{0,1\}^n$
$y \leftarrow G_{pk_0}(r)$ if $b = 0$
$y \leftarrow G_{pk_1}(r) \oplus \sigma$ if $b = 1$

$$\xrightarrow{\quad y \quad}$$

output: `receipt`

(opening phase)

$$\xrightarrow{\quad (b, r) \quad}$$

Check if $y = G_{pk_0}(r)$ for $b = 0$ or $y = G_{pk_1}(r) \oplus \sigma$ for $b = 1$.

If verification passes, accept. Otherwise, reject.

Figure 7. UC-secure BC scheme in the One-Time CRS Model [32].

Note that the protocol π_{COM} is a bit commitment protocol, and for string commitment, an instance of π_{COM} is needed to run for each bit of the string.

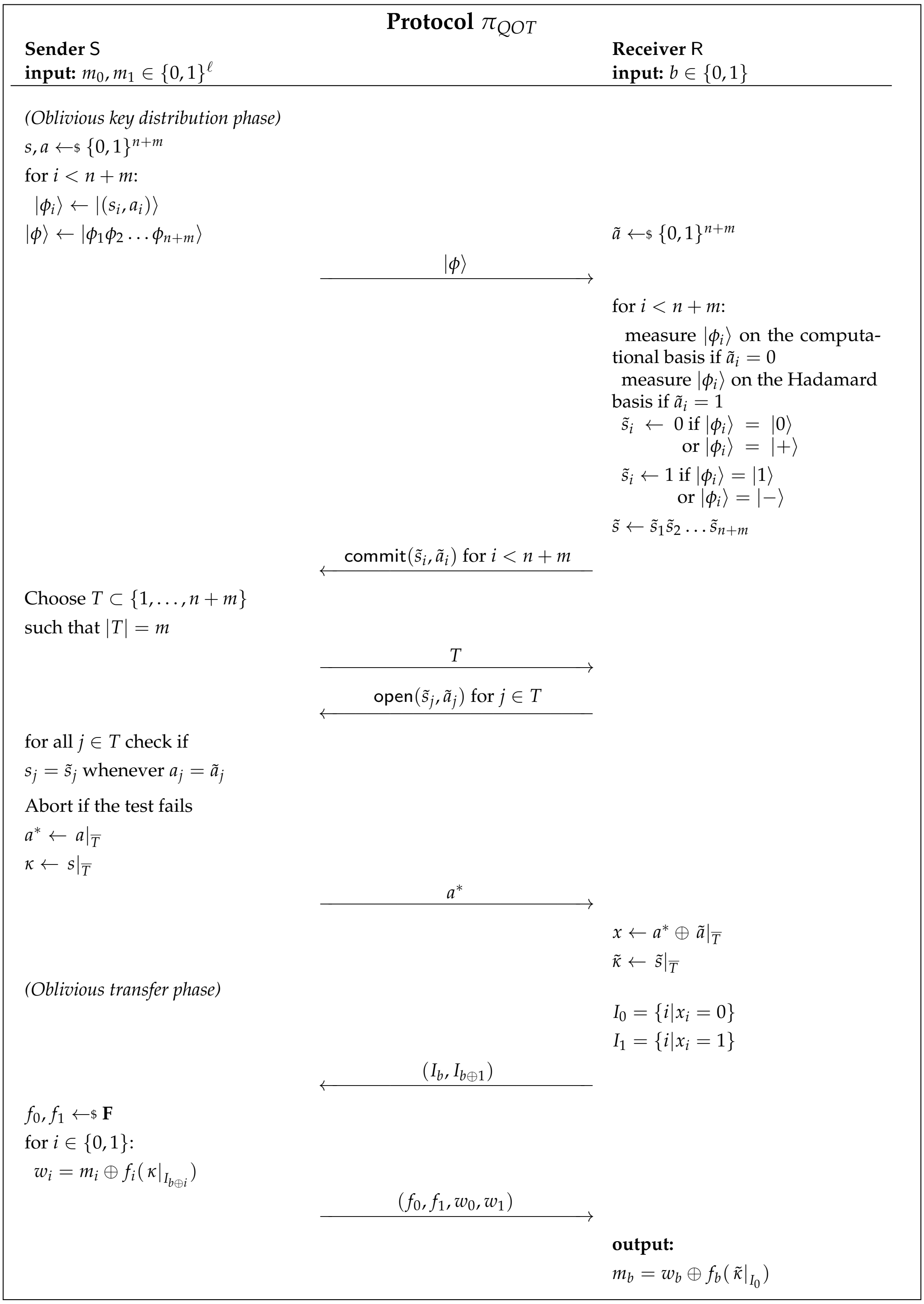
3.3. UC-Secure Quantum OT Protocol

The protocol in Figure 8 was proposed by Yao and has been shown to be statistically quantum-UC-secure with ideal commitments [28].

We describe the logical qubit states $|0\rangle$ and $|1\rangle$ (representing the computational basis), and the states $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ (representing the Hadamard basis). We use the following notation to define the states $|(s_i, a_i)\rangle$ for $s_i, a_i \in \{0,1\}$:

$$|(0,0)\rangle = |0\rangle \quad |(0,1)\rangle = |+\rangle,$$
$$|(1,0)\rangle = |1\rangle \quad |(1,1)\rangle = |-\rangle.$$

The protocol begins with the sender S preparing qubit states and sending them to the receiver R, which then samples a random string $\tilde{a}$. For every qubit received, R measures the i-th state on a computational basis if $\tilde{a}_i = 0$ or, on the Hadamard basis, if $\tilde{a}_i = 1$. Therefore, approximately half of R's measurement results will be correlated with the prepared states by S, while the rest will be uncorrelated. To ensure security against a dishonest R, it is required to commit information on all of his measurement bases and outcomes to S, which then picks a random subset of them and tests for correlations. The passing of this test (statistically) ensures that R measured its qubits honestly. Next, S shares with R the bases it used for her state-preparation and, with this information, R knows which of its results are correlated with the sender's. The receiver, then, creates two sets: I_0, with indices where it is measured on the same basis as S, and I_1, where their measuring bases differ. Following that, R uses its choice bit b to select the order in which it sends the two sets to S. The sender samples two hash functions f_0, f_1 at random, from a *2-universal* family of hash functions **F**, in order to generate uniform keys of appropriate size, as that of the messages m_0, m_1. S sends the encrypted messages w_0, w_1 to R, which can only decrypt the message corresponding to the set I_0.



Protocol π_{QOT}

Sender S **input:** $m_0, m_1 \in \{0,1\}^\ell$		**Receiver** R **input:** $b \in \{0,1\}$
(Oblivious key distribution phase) $s, a \leftarrow_\$ \{0,1\}^{n+m}$ for $i < n+m$: $\lvert\phi_i\rangle \leftarrow \lvert(s_i, a_i)\rangle$ $\lvert\phi\rangle \leftarrow \lvert\phi_1\phi_2\dots\phi_{n+m}\rangle$		$\tilde{a} \leftarrow_\$ \{0,1\}^{n+m}$
	$\xrightarrow{\lvert\phi\rangle}$	
		for $i < n+m$: measure $\lvert\phi_i\rangle$ on the computational basis if $\tilde{a}_i = 0$ measure $\lvert\phi_i\rangle$ on the Hadamard basis if $\tilde{a}_i = 1$ $\tilde{s}_i \leftarrow 0$ if $\lvert\phi_i\rangle = \lvert 0\rangle$ or $\lvert\phi_i\rangle = \lvert +\rangle$ $\tilde{s}_i \leftarrow 1$ if $\lvert\phi_i\rangle = \lvert 1\rangle$ or $\lvert\phi_i\rangle = \lvert -\rangle$ $\tilde{s} \leftarrow \tilde{s}_1\tilde{s}_2\dots\tilde{s}_{n+m}$
	$\xleftarrow{\mathsf{commit}(\tilde{s}_i, \tilde{a}_i) \text{ for } i < n+m}$	
Choose $T \subset \{1, \dots, n+m\}$ such that $\lvert T\rvert = m$		
	$\xrightarrow{T}$	
	$\xleftarrow{\mathsf{open}(\tilde{s}_j, \tilde{a}_j) \text{ for } j \in T}$	
for all $j \in T$ check if $s_j = \tilde{s}_j$ whenever $a_j = \tilde{a}_j$ Abort if the test fails $a^* \leftarrow a\vert_{\overline{T}}$ $\kappa \leftarrow s\vert_{\overline{T}}$		
	$\xrightarrow{a^*}$	
		$x \leftarrow a^* \oplus \tilde{a}\vert_{\overline{T}}$ $\tilde{\kappa} \leftarrow \tilde{s}\vert_{\overline{T}}$
(Oblivious transfer phase)		$I_0 = \{i \mid x_i = 0\}$ $I_1 = \{i \mid x_i = 1\}$
	$\xleftarrow{(I_b, I_{b\oplus 1})}$	
$f_0, f_1 \leftarrow_\$ \mathbf{F}$ for $i \in \{0,1\}$: $w_i = m_i \oplus f_i(\kappa\vert_{I_{b\oplus i}})$		
	$\xrightarrow{(f_0, f_1, w_0, w_1)}$	
		output: $m_b = w_b \oplus f_b(\tilde{\kappa}\vert_{I_0})$

Figure 8. Quantum UC-secure Quantum OT Protocol based on secure commitments [28].

3.4. Post-Quantum UC-Secure ROT Protocol

The protocol in Figure 9 is based on the recently proposed protocol by [30] (which was based on [29]), which has been shown to be UC-secure under the RLWE assumption in the Random Oracle Model (ROM). However, UC security using ROM does not directly lift to UC security against quantum adversaries. Taking that into consideration, our idea is to replace the random oracle calls, which are used to either commit to a string or to generate a random string.

In order to understand the protocol π_{ROT}, we need to provide some preliminary definitions. A signal function Sig and an extraction function Ext are described as in the key exchange protocol using RLWE of [34], to be used by the involved parties to reconcile a shared key.

Let $\sigma_0, \sigma_1 : \mathbb{Z}_q \to \{0,1\}$. We define σ_0, σ_1 as follows:

$$\sigma_0(a) = \begin{cases} 0, & a \in [-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor] \\ 1, & \text{otherwise} \end{cases} \quad \text{and} \quad \sigma_1(a) = \begin{cases} 0, & a \in [-\lfloor \frac{q}{4} + 1 \rfloor, \lfloor \frac{q}{4} + 1 \rfloor] \\ 1, & \text{otherwise} \end{cases}$$

Next, we need to extend σ_0, σ_1 to the ring case. For any $a = \sum_{i=0}^{n-1} a_i X^i \in R_q$, we define $\sigma_0, \sigma_1 : R_q \to R_2$ as follows:

$$\sigma_0(a) = \sum_{i=0}^{n-1} \sigma_0(a_i) X^i \quad \text{and} \quad \sigma_1(a) = \sum_{i=0}^{n-1} \sigma_1(a_i) X^i$$

The signal function $\mathsf{Sig} : R_q \to R_2$ can now be defined as $\mathsf{Sig}(a) = \sigma_b(a)$, where $b \leftarrow_\$ \{0,1\}$, while the extraction function $\mathsf{Ext} : R_q \times R_2 \to R_2$ is

$$\mathsf{Ext}(a, \sigma) = \left(a + \sigma \frac{q-1}{2} \mod q \right) \mod 2.$$

We can now describe the ROT protocol based on the RLWE assumption, Figure 9, which can be seen as a tweaked version of the protocol of [30], where we replace the random oracles by a commitment scheme and a NIZK protocol, modeled as functionalities.

Let χ and q be as in Definition 2 and ℓ be the security parameter. Let (m, h) be the common string, where $m, h \in R_q$, and let Ext and Sig be the algorithms defined above.

The protocol starts with both parties generating an RLWE sample. The sender S generates $p_\mathsf{S} = m s_\mathsf{S} + 2 e_\mathsf{S} \mod q$, and the receiver R generates $p_\mathsf{R}^c = m s_\mathsf{R} + 2 e_\mathsf{R} \mod q$, where c is a bit randomly chosen by R. If the sampled bit $c = 1$, then R computes $p_\mathsf{R}^0 = p_\mathsf{R}^1 - h \mod q$. The receiver then samples two strings $t_0, t_1 \leftarrow_\$ \{0,1\}^\ell$, commits both strings, and sends p_R^0 to S. The sender uses the common string h and p_R^0 to compute $p_\mathsf{R}^1 = p_\mathsf{R}^0 + h \mod q$ and uses both values $p_\mathsf{R}^0, p_\mathsf{R}^1$ to generate two RLWE samples. $k_\mathsf{S}^i = s_\mathsf{S} p_\mathsf{R}^i + 2 e'_\mathsf{S} \mod q$ for $i \in \{0,1\}$. S now computes $\sigma_i = \mathsf{Sig}(k_\mathsf{S}^i)$ and $\mathtt{sk}_\mathsf{S}^i = \mathsf{Ext}(k_\mathsf{S}^i, \sigma_i)$, for $i \in \{0,1\}$, and sends $p_\mathsf{S}, \sigma_0, \sigma_1$ to R. The receiver then generates an RLWE sample $k_\mathsf{R} = s_\mathsf{R} p_\mathsf{S} + 2 e'_\mathsf{R} \mod q$ from p_S and computes $\mathtt{sk}_\mathsf{R} = \mathsf{Ext}(k_\mathsf{R}, \sigma_c)$. The key exchange protocol guarantees that $\mathtt{sk}_\mathsf{S}^c = \mathtt{sk}_\mathsf{R}$ with overwhelming probability, so as to guarantee that R did not cheat (and indeed the computed $\mathtt{sk}_\mathsf{R}$). Both parties engage in a NIZK protocol. If the proof fails, S aborts; otherwise, he samples a bit a and two strings $r_0, r_1 \leftarrow_\$ \{0,1\}^\ell$ and sends a, r_0, r_1 to R. The receiver opens his initial commitment to S, and if the test passes, both parties output their messages: S outputs $(M_0 = \mathtt{sk}_\mathsf{S}^a \oplus r_a \oplus t_a, M_1 = \mathtt{sk}_\mathsf{S}^{a \oplus 1} \oplus r_{a \oplus 1} \oplus t_{a \oplus 1})$, and R outputs $(b = a \oplus c, M_b = \mathtt{sk}_\mathsf{R} \oplus r_c \oplus t_c)$.

To simplify the description of π_{ROT} in Figure 9, we represent $\mathcal{F}_{NIZK}$ with a single input from the prover R (the witness w) and a single output to the verifier S, where this output is 1 if w satisfies $\mathcal{R}$ or 0 otherwise. Let the binary relation $\mathcal{R}$ be such that

$$\mathcal{R}(x, w) = 1 \iff w = \mathtt{sk}_\mathsf{S}^0 \vee w = \mathtt{sk}_\mathsf{S}^1,$$

where $x = \mathsf{Enc}(\mathtt{sk}_\mathsf{S}^0, \mathtt{sk}_\mathsf{S}^1)$ for a given public key encryption scheme.

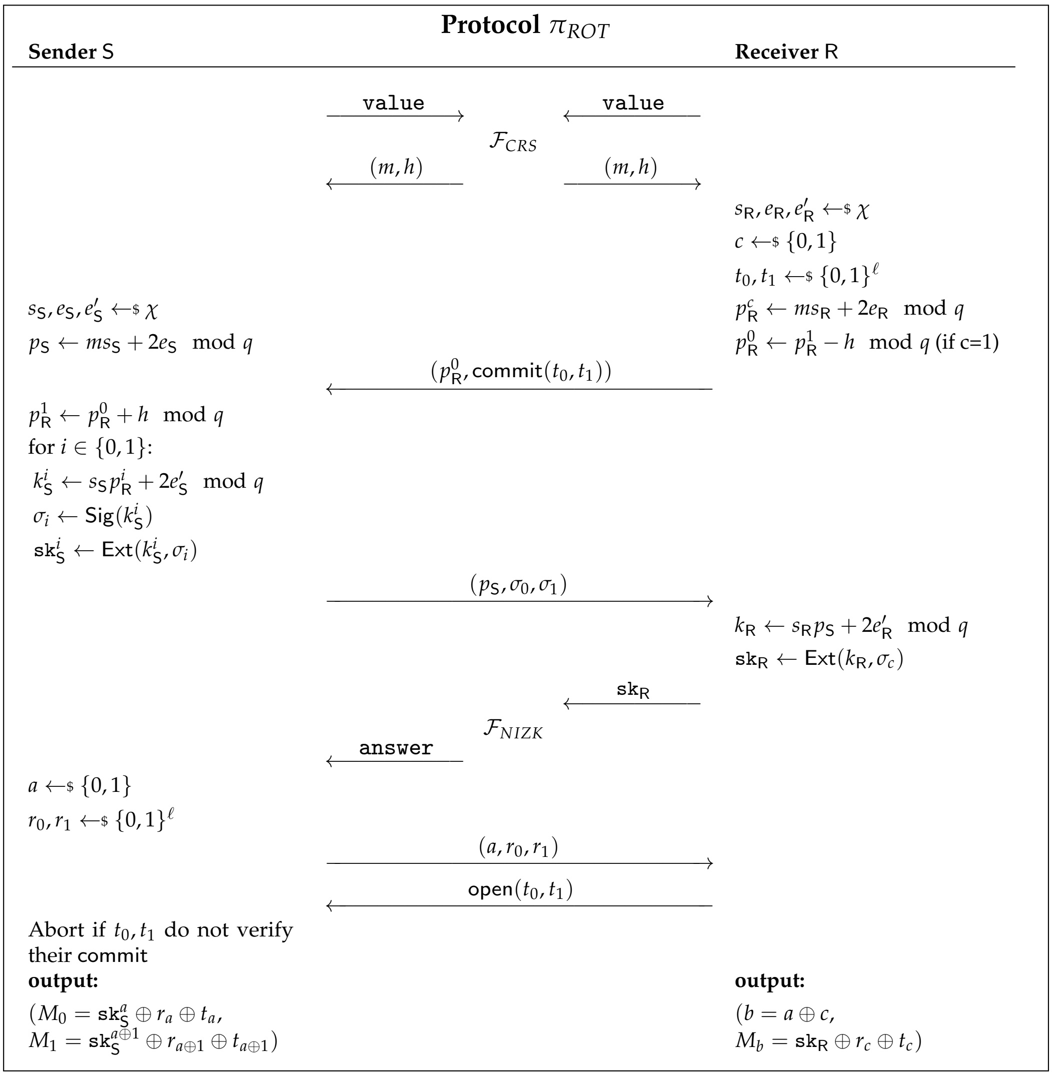

Protocol π_{ROT}
Sender S
Receiver R
value
value
$\mathcal{F}_{CRS}$
(m, h)
(m, h)
$s_R, e_R, e'_R \leftarrow_\$ \chi$
$c \leftarrow_\$ \{0,1\}$
$t_0, t_1 \leftarrow_\$ \{0,1\}^\ell$
$s_S, e_S, e'_S \leftarrow_\$ \chi$
$p_S \leftarrow ms_S + 2e_S \mod q$
$p_R^c \leftarrow ms_R + 2e_R \mod q$
$p_R^0 \leftarrow p_R^1 - h \mod q$ (if c=1)
$(p_R^0, \mathsf{commit}(t_0, t_1))$
$p_R^1 \leftarrow p_R^0 + h \mod q$
for $i \in \{0,1\}$:
$k_S^i \leftarrow s_S p_R^i + 2e'_S \mod q$
$\sigma_i \leftarrow \mathsf{Sig}(k_S^i)$
$\mathtt{sk}_S^i \leftarrow \mathsf{Ext}(k_S^i, \sigma_i)$
$(p_S, \sigma_0, \sigma_1)$
$k_R \leftarrow s_R p_S + 2e'_R \mod q$
$\mathtt{sk}_R \leftarrow \mathsf{Ext}(k_R, \sigma_c)$
$\mathtt{sk}_R$
$\mathcal{F}_{NIZK}$
answer
$a \leftarrow_\$ \{0,1\}$
$r_0, r_1 \leftarrow_\$ \{0,1\}^\ell$
(a, r_0, r_1)
$\mathsf{open}(t_0, t_1)$
Abort if t_0, t_1 do not verify their commit
output:
$(M_0 = \mathtt{sk}_S^a \oplus r_a \oplus t_a,$
$M_1 = \mathtt{sk}_S^{a\oplus 1} \oplus r_{a\oplus 1} \oplus t_{a\oplus 1})$
output:
$(b = a \oplus c,$
$M_b = \mathtt{sk}_R \oplus r_c \oplus t_c)$

Figure 9. UC ROT protocol in the CRS model based on the RLWE assumption.

The $\mathcal{F}_{NIZK}$ functionality can, for instance, be instantiated using the protocol described in [35]. This protocol is shown to be quantum-composable in the CRS model, based on the LWE assumption.

4. Security

In this section, we establish the quantum-UC security of the proposed protocols in the CRS model. We begin by analyzing the quantum protocol first and proving that $\pi_{OT\rightarrow ROT}$ is quantum-UC-secure when instantiated with π_{COM} and $\pi_{QOT}^{\pi_{COM}}$. We then prove the quantum-UC security of the π_{ROT}.

4.1. Quantum-UC Security of the Quantum ROT Protocol

Theorem 2. *Protocol $\pi_{OT\rightarrow ROT}$ quantum-UC-emulates $\mathcal{F}_{ROT}$ in the $\langle\mathcal{F}_{OT}, \mathcal{F}_{COM}\rangle$-hybrid model.*

Proof. We start by describing how the simulator $\mathcal{S}$ behaves in each of the possible cases for the execution of the protocol when an adversary $\mathcal{A}$ is present.

Corrupted Sender. In this case, $\mathcal{S}$ simulates the view of the sender, effectively controlling the inputs to $\mathcal{F}_{COM}$ and the input bit to $\mathcal{F}_{OT}$. In order to do so, we start by replacing $\mathcal{F}_{COM}$ by a commitment functionality $\mathcal{F}_{FakeCOM}$, which allows the receiver to cheat. In the commit phase, $\mathcal{F}_{FakeCOM}$ expects a message COMMIT instead of (COMMIT, x); in the open phase, $\mathcal{F}_{FakeCOM}$ expects a message (OPEN, x) instead of OPEN, which is then sent to the sender. We now change the receiver's implementation to match with the new functionality; that is, when committing to message m, the receiver stores that message and later gives it to $\mathcal{F}_{FakeCOM}$ when opening the commitment.

We can now describe how the simulator works. $\mathcal{S}$ starts by receiving (M_0, M_1) from $\mathcal{F}_{ROT}$; afterwards, it sends COMMIT to $\mathcal{F}_{FakeCOM}$, samples $c \leftarrow_{\$} \{0,1\}$, and sends c to $\mathcal{F}_{OT}$. Upon receiving d, the simulator extracts w_0, w_1 from observing the sender's call to $\mathcal{F}_{OT}$ and computes $r_d = M_0 \oplus w_d$ and $r_{d\oplus 1} = M_1 \oplus w_{d\oplus 1}$. Finally, it sends (OPEN,(r_0, r_1)) to $\mathcal{F}_{FakeCOM}$.

Corrupted Receiver. Now, $\mathcal{S}$ simulates the view of the receiver, controlling the input messages to $\mathcal{F}_{OT}$. The simulator starts by receiving (b, M) from $\mathcal{F}_{ROT}$. After receiving the commitment message, $\mathcal{S}$ extracts the strings r_0, r_1 and the bit c from observing the receiver's call to $\mathcal{F}_{COM}$ and $\mathcal{F}_{OT}$, respectively. It then computes $w_c = r_c \oplus M$ and $d = b \oplus c$ and samples $w_{c\oplus 1} \leftarrow_{\$} \{0,1\}^{\ell}$; afterwards, send (w_0, w_1) to $\mathcal{F}_{OT}$ and d to $\mathcal{A}$. When $\mathcal{F}_{COM}$ replies with $\mathsf{open}(r_0, r_1)$, it checks if the values received match the original commitments and aborts if they do not.

Both/None parties corrupted. When both parties are corrupted, $\mathcal{S}$ internally runs $\mathcal{A}$, which generates the messages for both parties.

When the adversary does not corrupt any party, the simulator does not have an input from the ideal functionality $\mathcal{F}_{ROT}$. As such, $\mathcal{S}$ runs the honest receiver and the honest sender, executing the needed algorithms when a dummy party is called in the ideal execution. The simulator forwards the messages of the honestly simulated protocol to $\mathcal{A}$.

To finish the proof, it remains to show that the simulated executions of the protocol are indistinguishable from the real one.

Claim 1. *If the adversary $\mathcal{A}$ corrupts the sender, then the real execution of the protocol $\pi_{OT\rightarrow ROT}$ is indistinguishable from the simulated one.*

Proof. The real world execution can be viewed as a game that proceeds as follows:

1. Sample values $r_0, r_1 \leftarrow_{\$} \{0,1\}^{\ell}$ and commit to values r_0, r_1.
2. Sample bit $c \leftarrow_{\$} \{0,1\}$ and run the OT protocol with the choice bit c.
3. Open the commitment to values r_0, r_1.

The ideal world execution can be viewed as a game that proceeds as follows:

1. Send COMMIT to $\mathcal{F}_{FakeCOM}$.
2. Sample bit $c \leftarrow_{\$} \{0,1\}$ and send c to $\mathcal{F}_{OT}$.
3. Send (OPEN, (r_0, r_1)) to $\mathcal{F}_{FakeCOM}$, where $r_d = M_0 \oplus w_d$ and $r_{d\oplus 1} = M_1 \oplus w_{d\oplus 1}$.

The differences between the two traces are the commitment functionality and how the values r_0, r_1 are generated. However, since the commitments are opened in the same way, replacing $\mathcal{F}_{COM}$ by $\mathcal{F}_{FakeCOM}$ leads to a perfectly indistinguishable network. Regarding r_0, r_1, since M_0, M_1 are uniform random values, which come from $\mathcal{F}_{ROT}$, the values r_0, r_1 are also statistically indistinguishable from uniform random values. Therefore, the two executions are statistically indistinguishable. □

Claim 2. *If the adversary $\mathcal{A}$ corrupts the receiver, then the real execution of the protocol $\pi_{OT\rightarrow ROT}$ is indistinguishable from the simulated one.*

Proof. The real world execution can be viewed as a game that proceeds as follows:

1. Sample strings $w_0, w_1 \leftarrow_{\$} \{0,1\}^{\ell}$ and run the OT protocol with w_0, w_1.
2. Sample bit d and send it to R.
3. Check if the received values verify their commitment.

The ideal world execution can be viewed as a game that proceeds as follows:

1. Sample string $w_{c\oplus 1} \leftarrow_{\$} \{0,1\}^{\ell}$ and compute $w_c = r_c \oplus M$; afterwards, send (w_0, w_1) to $\mathcal{F}_{OT}$.
2. Compute $d = b \oplus c$ and send it to R.
3. Check if the received values verify their commit.

In this case, the difference between both traces is in how w_c and d are generated. Since M and b are uniform random values, which come from $\mathcal{F}_{ROT}$, both the string w_c and the bit d are statistically indistinguishable from a uniform random string and a uniform random bit, respectively. Thus, the above two executions are statistically indistinguishable. □

Finally, it is trivial to conclude that, when both parties are corrupted and when neither parties are corrupted, the simulated executions of the protocol are indistinguishable from the real execution. This concludes the proof. □

We have shown that, with $\pi_{OT\rightarrow ROT}$, we can transform π_{QOT} into a ROT. We now need to prove that π_{COM} remains UC-secure when working in a quantum setting.

Theorem 3. *Let G_{pk} be a quantum robust PRNG. π_{COM} then (computationally) quantum UC-emulates $\mathcal{F}_{COM}$ in the CRS model.*

Proof. We start by briefly describing the UC security proof of π_{COM} by Canneti in [33].

The simulation starts with the simulator $\mathcal{S}$ by generating pk_0, pk_1, sampling random $r_0, r_1 \in \{0,1\}^n$, and setting $\sigma = G_{pk_0}(r_0) \oplus G_{pk_1}(r_1)$. With this fake string, $\mathcal{S}$ tells the adversary $\mathcal{A}$ that the sender is committed to $y = G_{pk_0}(r_0)$. By later sending r_0 or r_1, the simulator is able to open the commitment to either $b = 0$ or to $b = 1$, respectively. If it were possible to distinguish the fake string from the real one, it would contradict the pseudo-randomness of the generator.

When working in a quantum setting, the indistinguishability of the fake string reduces to the pseudo-randomness of the generator; that is, the environment can only distinguish between the real world and ideal world executions if it is possible to distinguish the fake string σ from the real one. As such, if the generators are quantum robust, the environment will not be able to distinguish between both strings. Therefore, the arguments used in the classical UC security proof follow for quantum UC security as well. □

Finally, we analyze the security of the proposed composition of protocols. Let π_{QROT} denote $\pi_{OT\rightarrow ROT}$ instantiated with π_{COM} and $\pi_{QOT}^{\pi_{COM}}$.

Theorem 4. *Protocol π_{QROT} quantum-UC-emulates $\mathcal{F}_{ROT}$.*

Proof. First, we analyze the UC security of $\pi_{QOT}^{\pi_{COM}}$. Protocol π_{QOT} with ideal commitments is known to be universally composable [28]; as such, since π_{COM} is a composable commitment scheme, we have that $\pi_{QOT}^{\pi_{COM}}$ quantum-UC-emulates $\mathcal{F}_{OT}$.

Finally, as was shown in Theorem 2, $\pi_{OT\rightarrow ROT}$ with ideal commitments and an ideal OT is universally composable. Since both π_{COM} and $\pi_{QOT}^{\pi_{COM}}$ are universally composable, the result follows directly. □

A downside of using π_{COM} as the commitment scheme is that we require a call to π_{COM} for each bit of the string we intend to commit, which will affect the protocol's efficiency. However, since a composable commitment is required, this is our best suggestion in the CRS model.

4.2. Quantum-UC Security of the Post-Quantum ROT Protocol

We now analyze the security of π_{ROT}. The simulator will use its ability to program the CRS and extract the NIZK witness in order to obtain the desired UC security.

Theorem 5. *Protocol π_{ROT} (computationally) quantum-UC-emulates F_{ROT} in the CRS model, given that the HNF-RLWE assumption holds.*

Proof. Once again, we describe the behavior of the simulator $\mathcal{S}$ in each of the possible cases for the execution of the protocol when an adversary $\mathcal{A}$ is present.

Corrupted Sender. The simulator $\mathcal{S}$ simulates the view of the sender, meaning that it controls the communication with R as well as the inputs of $\mathcal{F}_{COM}$ and $\mathcal{F}_{NIZK}$. As in the proof of security for π_{QROT}, we will be replacing $\mathcal{F}_{COM}$ by the functionality $\mathcal{F}_{FakeCOM}$ and changing the receiver's implementation to match $\mathcal{F}_{FakeCOM}$.

$\mathcal{S}$ starts by receiving (M_0, M_1) from $\mathcal{F}_{ROT}$. It then samples $c \leftarrow_{\$} \{0,1\}$ and $t_0, t_1 \leftarrow_{\$} \{0,1\}^{\ell}$, as an honest receiver would. Next, it computes two RLWE samples, $p_{\mathsf{R}}^0 = ms_{\mathsf{R}}^0 + 2e_{\mathsf{R}}^0 \mod q$ and $p_{\mathsf{R}}^1 = ms_{\mathsf{R}}^0 + 2e_{\mathsf{R}}^0 \mod q$, sets $h = p_{\mathsf{R}}^1 - p_{\mathsf{R}}^0$, and programs $\mathcal{F}_{CRS}$ to return (m, h) when queried. Following that, it sends p_{R}^0 to $\mathcal{A}$ and sends COMMIT to $\mathcal{F}_{FakeCOM}$.

After receiving $(p_{\mathsf{S}}, \sigma_0, \sigma_1)$, $\mathcal{S}$ computes $\mathtt{sk}_{\mathsf{R}}^i = \mathsf{Ext}(s_{\mathsf{R}}^i p_{\mathsf{S}} + 2{e'_{\mathsf{R}}}^i, \sigma_i)$, for $i \in \{0,1\}$, and sends $\mathtt{sk}_{\mathsf{R}}^c$ to $\mathcal{F}_{NIZK}$. Finally, upon receiving a, r_0, r_1, S computes $t_a = M_0 \oplus \mathtt{sk}_{\mathsf{S}}^a \oplus r_a$ and $t_{a\oplus 1} = M_1 \oplus \mathtt{sk}_{\mathsf{S}}^{a\oplus 1} \oplus r_{a\oplus 1}$ and sends (OPEN, (t_0, t_1)) to $\mathcal{F}_{FakeCOM}$.

Corrupted Receiver. In this case, $\mathcal{S}$ simulates the view of the receiver, controlling the communication with S. The simulator starts by receiving (b, M) from $\mathcal{F}_{ROT}$. It computes p_{S} as an honest sender; after receiving p_{R}^0 as well as the receipt of the commitment, it computes $\mathtt{sk}_{\mathsf{S}}^i, \sigma_i$ honestly, for $i \in \{0,1\}$, and sends $p_{\mathsf{S}}, \sigma_0, \sigma_1$ to $\mathcal{A}$. After receiving the reply from $\mathcal{F}_{NIZK}$, if the test passed, $\mathcal{S}$ extracts c from observing the call made to $\mathcal{F}_{NIZK}$ and comparing $\mathtt{sk}_{\mathsf{R}}$ to $\mathtt{sk}_{\mathsf{S}}^0$ and $\mathtt{sk}_{\mathsf{S}}^1$. Finally, it computes $a = b \oplus c$ and $r_c = M \oplus \mathtt{sk}_{\mathsf{S}}^c \oplus t_c$, samples $r_{c\oplus 1} \leftarrow_{\$} \{0,1\}^{\ell}$ and sends a, r_0, r_1 to $\mathcal{A}$. At the end, it checks if t_0, t_1 match the initial commitment, aborting if they do not.

Both/None parties corrupted. Here, both cases work as in the previous UC security proof. When both parties are corrupted, the adversary is ran internally by $\mathcal{S}$. When neither of the parties are corrupted, $\mathcal{S}$ runs the honest receiver and sender, sending all the messages between them to $\mathcal{A}$.

Again, we now need to show that the real execution of the protocol is indistinguishable from the simulated ones.

Claim 3. *If the adversary $\mathcal{A}$ corrupts the sender, then the real execution of the protocol π_{ROT} is indistinguishable from the simulated one.*

Proof. The real world execution can be viewed as a game that proceeds as follows:

1. Sample bit $c \leftarrow_{\$} \{0,1\}$ and strings $t_0, t_1 \leftarrow_{\$} \{0,1\}^{\ell}$.
   Generate RLWE sample p_{R} and, if $c = 1$, compute $p_{\mathsf{R}}^0 = p_{\mathsf{R}}^1 - h$.
   Send p_{R}^0 and commit to values t_0, t_1.
2. Compute $\mathtt{sk}_{\mathsf{R}} = \mathsf{Ext}(s_{\mathsf{R}} p_{\mathsf{S}} + 2e'_{\mathsf{R}}, \sigma_c)$ and run the NIZK protocol with $\mathtt{sk}_{\mathsf{R}}$.
3. Open the commitment to values t_0, t_1.

The ideal world execution can be viewed as a game that proceeds as follows:

1. Sample bit $c \leftarrow_{\$} \{0,1\}$.
   Generate RLWE samples $p_{\mathsf{R}}^0, p_{\mathsf{R}}^1$ and program $\mathcal{F}_{CRS}$ to return $(m, p_{\mathsf{R}}^1 - p_{\mathsf{R}}^0)$.
   Send p_{R}^0 to $\mathcal{A}$ and send COMMIT to $\mathcal{F}_{FakeCOM}$.
2. Compute $\mathtt{sk}_{\mathsf{R}}^i = \mathsf{Ext}(s_{\mathsf{R}}^i p_{\mathsf{S}} + 2{e'_{\mathsf{R}}}^i, \sigma_i)$, for $i \in \{0,1\}$, and send $\mathtt{sk}_{\mathsf{R}}^c$ to $\mathcal{F}_{NIZK}$.
3. Send (OPEN,(t_0, t_1)) to $\mathcal{F}_{FakeCOM}$, where $t_a = M_0 \oplus \mathtt{sk}_{\mathsf{S}}^a \oplus r_a$ and $t_{a\oplus 1} = M_1 \oplus \mathtt{sk}_{\mathsf{S}}^{a\oplus 1} \oplus r_{a\oplus 1}$.

The first difference between both games is in p_{R}^0 and p_{R}^1. In the real world game, only p_{R}^c is an RLWE sample ($p_{\mathsf{R}}^{c\oplus 1}$ is a uniform random sample), while in the ideal world game, both p_{R}^0 and p_{R}^1 are RLWE samples. Given that the RLWE assumption holds, both situations are indistinguishable.

Once again, replacing $\mathcal{F}_{COM}$ by $\mathcal{F}_{FakeCOM}$ leads to an indistinguishable network, since the commitments are opened in the same way. Finally, in the real world, t_0, t_1 are uniform random values, while in the ideal world, they are not. However, since M_0, M_1 are uniform random values that come from $\mathcal{F}_{ROT}$, the values in the ideal world are statistically indistinguishable from uniform random values.

Thus, the two executions are indistinguishable, assuming the RLWE assumption holds. □

Claim 4. *If the adversary $\mathcal{A}$ corrupts the receiver, then the real execution of the protocol π_{ROT} is indistinguishable from the simulated one.*

Proof. The real world execution can be viewed as a game that proceeds as follows:

1. Generate RLWE sample p_{S}.
2. Compute $p_{\mathsf{R}}^1 = p_{\mathsf{R}}^0 + h \mod q$. Compute σ_i and $\mathtt{sk}_{\mathsf{S}}^i$, for $i \in \{0,1\}$.
   Send $(p_{\mathsf{S}}, \sigma_0, \sigma_1)$.
3. Run the NIZK protocol and check if the test passes; abort if it does not.
   Sample $a \leftarrow_{\$} \{0,1\}$ and $r_0, r_1 \leftarrow_{\$} \{0,1\}^{\ell}$. Send (a, r_0, r_1).
4. Check if the received values verify their commitment; abort if they do not.

The ideal world execution can be viewed as a game that proceeds as follows:

1. Generate RLWE sample p_{S}.
2. Compute $p_{\mathsf{R}}^1 = p_{\mathsf{R}}^0 + h \mod q$. Compute σ_i and $\mathtt{sk}_{\mathsf{S}}^i$, for $i \in \{0,1\}$.
   Send $(p_{\mathsf{S}}, \sigma_0, \sigma_1)$.
3. Check if the received answer from $\mathcal{F}_{NIZK}$ is 1; abort if it is not.
   Send (a, r_0, r_1), where $a = b \oplus c$, $r_c = M \oplus \mathtt{sk}_{\mathsf{S}}^c \oplus t_c$, and $r_{1-c} \leftarrow_{\$} \{0,1\}^{\ell}$.
4. Check if the received values verify their commitment; abort if they do not.

The games differ in how a and r_c are generated; however, since b and M are uniform random values that come from $\mathcal{F}_{ROT}$, both r_c and a are statistically indistinguishable from a uniform random string and a uniform random bit, respectively. Hence, the real world execution and the ideal world execution are indistinguishable, assuming that the RLWE assumption holds. □

It remains to be seen whether the simulated executions where both parties are corrupted and when no party is corrupted are also indistinguishable. As in the previous proof, both are trivial, which concludes the proof. □

5. Conclusions

In view of the usefulness of MPC and the steady evolution of both quantum technology and post-quantum cryptography techniques, as well as recognizing the potential threat quantum computers can present in the landscape of information security, we have proposed two potential solutions for quantum secure implementations of ROT.

Both of these protocols have in common that they use a commitment scheme based on quantum-secure pseudo-random generators, which is universally composable in the CRS model. The CRS assumption has the advantage of being weaker and better understood than the quantum random oracle, and it is independent of technological limitations as opposed to the noisy storage assumptions, which are two of the most common models in which the security of OT protocols is studied.

The first construction is based on a quantum OT protocol composed with a quantum secure bit commitment, which is then transformed into a ROT protocol. The usage of a PRNG, which is secure against any poly-time quantum distinguisher, is the key to the commitment scheme's quantum composability. The second construction is based on a highly efficient UC-secure ROT protocol from the RLWE assumption, initially proposed in the ROM. Our protocol differs in that we remove the random oracle's requirement, replacing it by a commitment scheme and non-interactive zero knowledge protocol, which allows us to make a quantum-secure UC protocol, but in the CRS model instead.

Potential future work directions include the following:

- Further optimization of the commitment scheme to reduce the number of CRS calls and PRNG computations per committed bit in the context of a string commitment scheme.
- The implementation of both protocols and a comparison of their performance, taking available (quantum) technologies into account. This poses a challenge, as the limitations of quantum technologies are much less known than traditional computational power and communication.

Author Contributions: Conceptualization, P.M.; investigation and formal analysis B.C., P.B., M.G., M.L. and P.M.; writing—original draft preparation, B.C.; writing—review and editing, M.G.; validation, M.G. and M.L.; supervision, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50008/2020 (Instituto de Telecomunicações via actions QuRUNNER, QUESTS) and Projects QuantumMining POCI-01-0145-FEDER-031826, PREDICT PTDC/CCI-CIF/29877/2017, and QuantumPrime PTDC/EEI-TEL/8017/2020. BC thanks Capgemini Engineering. PB gratefully acknowledges the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017. MG gratefully acknowledges the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135182/2017.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank Preeti Yadav for editorial improvements.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rabin, M.O. How To Exchange Secrets with Oblivious Transfer. *IACR Cryptol. ePrint Arch.* **2005**, *2005*, 187. Originally published as: Technical Report TR-81, Aiken Computation Lab, Harvard University, Cambridge, MA, USA, 1981.
2. Ishai, Y.; Prabhakaran, M.; Sahai, A. Founding Cryptography on Oblivious Transfer—Efficiently. In Proceedings of the Advances in Cryptology—CRYPTO 2008, Santa Barbara, CA, USA, 17–21 August 2008; Wagner, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 572–591.
3. Kilian, J. Founding Cryptography on Oblivious Transfer. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 4–6 May 1988; Association for Computing Machinery: New York, NY, USA, 1988; pp. 20–31. doi:10.1145/62212.62215.
4. Goldreich, O.; Micali, S.; Wigderson, A. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 307–328.
5. Blum, M. Coin flipping by telephone a protocol for solving impossible problems. *ACM SIGACT News* **1983**, *15*, 23–27.
6. Cramer, R.; Damgård, I.; Maurer, U. General secure multi-party computation from any linear secret-sharing scheme. In Proceedings of the Advances in Cryptology—EUROCRYPT 2000, Bruges, Belgium, 14–18 May 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 316–334.
7. Lindell, Y.; Pinkas, B. Secure two-party computation via cut-and-choose oblivious transfer. *J. Cryptol.* **2012**, *25*, 680–722.
8. Even, S.; Goldreich, O.; Lempel, A. A randomized protocol for signing contracts. *Commun. ACM* **1985**, *28*, 637–647.
9. Crépeau, C. Equivalence between two flavours of oblivious transfers. In Proceedings of the Advances in Cryptology—CRYPTO '87, Santa Barbara, CA, USA, 16–20 August 1987; Springer: Berlin/Heidelberg, Germany, 1987; pp. 350–354.
10. Yao, A.C. Protocols for secure computations. In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), Chicago, IL, USA, 3–5 November 1982; IEEE Computer Society: Washington, DC, USA, 1982; pp. 160–164. doi:10.1109/SFCS.1982.38.
11. Goldreich, O.; Micali, S.; Wigderson, A. How to Play ANY Mental Game. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 25–27 May 1987; Association for Computing Machinery: New York, NY, USA, 1987; pp. 218–229. doi:10.1145/28395.28420.
12. Ishai, Y.; Kilian, J.; Nissim, K.; Petrank, E. Extending Oblivious Transfers Efficiently. In Proceedings of the Advances in Cryptology—CRYPTO 2003, Santa Barbara, CA, USA, 17–21 August 2003; Boneh, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 145–161.
13. Orrù, M.; Orsini, E.; Scholl, P. Actively Secure 1-out-of-N OT Extension with Application to Private Set Intersection. In Proceedings of the Topics in Cryptology—CT-RSA 2017, San Francisco, CA, USA, 14–17 February 2017; Handschuh, H., Ed.; Springer International Publishing: Cham, Switzerland, 2017; pp. 381–396.

14. Pinkas, B.; Rosulek, M.; Trieu, N.; Yanai, A. Spot-light: Lightweight private set intersection from sparse ot extension. In Proceedings of the Advances in Cryptology—CRYPTO 2019, Santa Barbara, CA, USA, 18–22 August 2019; Springer International Publishing: Cham, Switzerland, 2019; pp. 401–431.
15. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; IEEE Computer Society: Washington, DC, USA, 1994; pp. 124–134. doi:10.1109/SFCS.1994.365700.
16. Lyubashevsky, V.; Peikert, C.; Regev, O. On Ideal Lattices and Learning with Errors over Rings. In Proceedings of the Advances in Cryptology—EUROCRYPT 2010, French Riviera, 30 May–3 June 2010; Gilbert, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–23.
17. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005; Association for Computing Machinery: New York, NY, USA, 2005; pp. 84–93. doi:10.1145/1060590.1060603.
18. Broadbent, A.; Schaffner, C. Quantum cryptography beyond quantum key distribution. *Des. Codes Cryptogr.* **2015**, *78*, 351–382. doi:10.1007/s10623-015-0157-4.
19. Renner, R.; Gisin, N.; Kraus, B. Information-theoretic security proof for quantum-key-distribution protocols. *Phys. Rev. A* **2005**, *72*, 012332.
20. Shenoy-Hejamadi, A.; Pathak, A.; Radhakrishna, S. Quantum Cryptography: Key Distribution and Beyond. *Quanta* **2017**, *6*, 1. doi:10.12743/quanta.v6i1.57.
21. Lo, H.K.; Chau, H.F. Is Quantum Bit Commitment Really Possible? *Phys. Rev. Lett.* **1997**, *78*, 3410–3413. doi:10.1103/physrevlett.78.3410.
22. Mayers, D. Unconditionally Secure Quantum Bit Commitment is Impossible. *Phys. Rev. Lett.* **1997**, *78*, 3414–3417. doi:10.1103/physrevlett.78.3414.
23. Erven, C.; Ng, N.; Gigov, N.; Laflamme, R.; Wehner, S.; Weihs, G. An experimental implementation of oblivious transfer in the noisy storage model. *Nat. Commun.* **2014**, *5*. doi:10.1038/ncomms4418.
24. Furrer, F.; Gehring, T.; Schaffner, C.; Pacher, C.; Schnabel, R.; Wehner, S. Continuous-Variable Protocol for Oblivious Transfer in the Noisy-Storage Model. *Nat. Commun.* **2018**, *9*. doi:10.1038/s41467-018-03729-4.
25. Ng, N.H.Y.; Joshi, S.K.; Chen Ming, C.; Kurtsiefer, C.; Wehner, S. Experimental implementation of bit commitment in the noisy-storage model. *Nat. Commun.* **2012**, *3*. doi:10.1038/ncomms2268.
26. Qiang, X.; Zhou, X.; Aungskunsiri, K.; Cable, H.; O'Brien, J.L. Quantum processing by remote quantum control. *Quantum Sci. Technol.* **2017**, *2*, 045002. doi:10.1088/2058-9565/aa78d6.
27. Long, G.L.; Liu, X.S. Theoretically efficient high-capacity quantum-key-distribution scheme. *Phys. Rev. A* **2002**, *65*, 032302. doi:10.1103/PhysRevA.65.032302.
28. Unruh, D. Universally Composable Quantum Multi-party Computation. In Proceedings of the Advances in Cryptology—EUROCRYPT 2010, French Riviera, 30 May–3 June 2010; Gilbert, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 486–505.
29. Branco, P.; Ding, J.; Goulão, M.; Mateus, P. A Framework for Universally Composable Oblivious Transfer from One-Round Key-Exchange. In Proceedings of the IMA International Conference on Cryptography and Coding, Oxford, United Kingdom, 15–17 December 2019; Albrecht, M., Ed.; Springer International Publishing: Cham, Switzerland, 2019; pp. 78–101.
30. Branco, P.; Fiolhais, L.; Goulão, M.; Martins, P.; Mateus, P.; Sousa, L. ROTed: Random Oblivious Transfer for Embedded Devices. IACR Transactions of Cryptographic Hardware and Embedded Systems. Available online: https://eprint.iacr.org/2021/935 (accessed on 7 June 2021).
31. Applebaum, B.; Cash, D.; Peikert, C.; Sahai, A. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In Proceedings of the Advances in Cryptology—CRYPTO 2009, Santa Barbara, CA, USA, 16–20 August 2009; Halevi, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 595–618.
32. Canetti, R. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, Las Vegas, NV, USA, 14–17 October 2001; IEEE Computer Society: Washington, DC, USA, 2001; p. 136.
33. Canetti, R.; Fischlin, M. Universally Composable Commitments. In Proceedings of the Advances in Cryptology—CRYPTO 2001, Santa Barbara, CA, USA, 19–23 August 2001; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 19–40.
34. Jintai Ding, Xiang Xie, X.L. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. Cryptology ePrint Archive, Report 2012/688. 2012. Available online: https://eprint.iacr.org/2012/688 (accessed on 7 June 2021).
35. Canetti, R.; Sarkar, P.; Wang, X. Triply Adaptive UC NIZK. Cryptology ePrint Archive, Report 2020/1212. 2020. Available online: https://eprint.iacr.org/2020/1212 (accessed on 7 June 2021).