Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Quantum security analysis of a lattice-based oblivious transfer protocol*

Mo-meng LIU[†‡1], Juliane KRÄMER[2], Yu-pu HU[1], Johannes BUCHMANN[2]

([1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China*)

([2]*Technische Universität Darmstadt, Darmstadt 64289, Germany*)

[†]E-mail: liumomeng@gmail.com

Received Jan. 12, 2017; Revision accepted Apr. 17, 2017; Crosschecked Sept. 24, 2017

Abstract: Because of the concise functionality of oblivious transfer (OT) protocols, they have been widely used as building blocks in secure multiparty computation and high-level protocols. The security of OT protocols built upon classical number theoretic problems, such as the discrete logarithm and factoring, however, is threatened as a result of the huge progress in quantum computing. Therefore, post-quantum cryptography is needed for protocols based on classical problems, and several proposals for post-quantum OT protocols exist. However, most post-quantum cryptosystems present their security proof only in the context of classical adversaries, not in the quantum setting. In this paper, we close this gap and prove the security of the lattice-based OT protocol proposed by Peikert *et al.* (CRYPTO, 2008), which is universally composably secure under the assumption of learning with errors hardness, in the quantum setting. We apply three general quantum security analysis frameworks. First, we apply the quantum lifting theorem proposed by Unruh (EUROCRYPT, 2010) to prove that the security of the lattice-based OT protocol can be lifted into the quantum world. Then, we apply two more security analysis frameworks specified for post-quantum cryptographic primitives, i.e., simple hybrid arguments (CRYPTO, 2011) and game-preserving reduction (PQCrypto, 2014).

Key words: Oblivious transfer; Post-quantum; Lattice-based; Learning with errors; Universally composable
https://doi.org/10.1631/FITEE.1700039 **CLC number:** TP309.7

1 Introduction

As a fundamental cryptographic primitive, oblivious transfer (OT) has been proposed to depict a scenario between two parties, where one party (called the sender) sends a message to the other party (called the receiver) with the requirement that the receiver can obtain only this message with probability 1/2 and the sender remains oblivious as to whether the message has been received or not. To build protocols for secure multiparty computation, a more useful flavor of OT, called 1-out-of-2 OT, was developed, where the receiver is allowed to retrieve one message from the sender's message pair without knowing anything about the other message, and the sender is required not to know about the receiver's message choice. Due to the concise functionality of OT, it can be used to securely implement some basic mathematical operations, e.g., secure two-party multiplication (Gilboa, 1999), in an efficient way. Therefore, OT is widely used as a building block for cryptographic construction and holds an important position in the development of modern cryptography.

However, most OT protocols are built from cryptosystems whose security relies on number theoretical problems, such as factoring and the discrete logarithm, which are only assumed to be hard for

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2017YFB0802000), the National Natural Science Foundation of China (Nos. 61672412, 61472309, and 61572390), and the China Scholarship Council (No. 201406960041)

ORCID: Mo-meng LIU, http://orcid.org/0000-0002-8545-5551

classical algorithms. Adversaries equipped with a quantum computer, however, can efficiently break those classical cryptographic constructions (Shor, 1997). Therefore, the security of OT protocols built upon these hardness assumptions will immediately break down in the quantum setting.

A natural countermeasure is to apply quantum resistant primitives as the cornerstone for OT constructions, in the sense that one uses cryptosystems whose underlying computational assumptions remain hard in the quantum setting. Developing and improving such cryptosystems is the scope of the research area of post-quantum cryptography (Bernstein *et al.*, 2009). It emerges from the challenge caused by quantum computers and includes subareas such as lattice-based (Micciancio and Regev, 2009; Peikert, 2009) cryptography and code-based (Sendrier, 2011) cryptography. Such post-quantum cryptosystems can be used as underlying primitives of cryptographic protocols and possibly make them secure against quantum adversaries. However, most existing security proofs of post-quantum constructions assume only classical adversaries and base the promised quantum security on the dependency on quantum resistant assumptions. Nonetheless, quantum resistant assumptions alone do not guarantee the quantum security of cryptographic constructions, since other fundamental issues in the security proof may be subtle and easily overlooked, especially in the complex proofs of protocols.

1.1 Related works

The concerns above have led to research on the security of classical cryptosystems in a quantum world. Damgård *et al.* (2014) introduced a new quantum attack model, i.e., the superposition attack model, on classical cryptographic protocols, where it allows adversaries to query in quantum superposition. Moreover, they revisited the security of several classical primitives in this stronger attack model. Unruh (2010) presented a quantum analogue of the universal composability (UC) model and constructed a statistically quantum-UC-secure OT protocol. In addition, he defined and constructed quantum proofs of knowledge (Unruh, 2012), where the main technique, i.e., a new quantum rewinding technique (Watrous, 2009), was used to extract witness in many classical proofs of knowledge. Hallgren *et al.* (2011) showed the existence of classical two-party protocols under reasonable computational assumptions that can keep their security in a quantum world. Fehr *et al.* (2013) explored the feasibility of realizing functionalities in the UC framework by moving from the classical to quantum world, regarding both computational and information-theoretic securities.

1.2 Contribution

By using three general security analysis results, we give a comprehensive quantum security analysis for an existing lattice-based OT protocol (Peikert *et al.*, 2008) which is secure in the UC framework. Our work relies mainly on the quantum lifting theorem (QLT) (Unruh, 2010), to explore the resilience of this latticed-based OT protocol against quantum adversaries. We present two additional discussions to supplement our analysis, one of which relies on a framework called 'simple hybrid arguments (SHA)' (Hallgren *et al.*, 2011; 2015), which supports the conjecture for the computational quantum lifting theorem proposed by Unruh (2010). The other one shows that the analysis we have made also falls into a framework termed 'game-preserving reduction' (Song, 2014), which is proposed to discuss the quantum resistance of post-quantum primitives from the perspective of a provable security reduction. Our work can be regarded as a concrete application of general quantum security analysis tools for post-quantum cryptographic constructions.

2 Preliminaries

For clarity, we first introduce some basic notations and relevant backgrounds. Most notations and concepts mentioned are taken from Canetti (2001), Peikert *et al.* (2008), and Unruh (2010). We assume that the reader is familiar with the basic concepts of quantum information theory (Nielsen and Chuang, 2010).

2.1 Notation

We use bold lowercase to denote vector, e.g., $\boldsymbol{v}$, and bold uppercase letter to denote matrix, e.g., $\boldsymbol{M}$. The notation $\boldsymbol{v}^{\mathrm{T}}$ represents the transpose of vector $\boldsymbol{v}$. For any $x, y \in \mathbb{R}$ with $y > 0$, $x \bmod y = x - \lfloor x/y \rfloor y$ and $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$. For any $n \in \mathbb{N}$, $[n]$ represents a set $\{1, 2, \ldots, n\}$.

For an integer $q \geq 1$, $\mathbb{Z}_q$ denotes the quotient ring $\mathbb{Z}/q\mathbb{Z}$. We let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the group of reals $[0, 1)$ with modulo 1 addition. For any $\alpha \in \mathbb{R}^+$, Ψ_α denotes the distribution on $\mathbb{T}$ obtained by sampling from a variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. Moreover, its discretization $\bar{\Psi}_\alpha : \mathbb{Z}_q \to \mathbb{R}^+$ defines the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_{\Psi_\alpha} \rceil \bmod q$, where X_{Ψ_α} has distribution Ψ_α.

If D is a probability distribution over $\mathbb{Z}_q$, then $x \leftarrow D$ denotes sampling $x \in \mathbb{Z}_q$ according to D. If $D(\cdot)$ is a probabilistic algorithm, $y \leftarrow D(x)$ denotes running D on input x and assigning the output to y.

Let $n \in \mathbb{N}$ denote a security parameter which is taken as an implicit input to all other quantities. We classify the growth of functions using standard asymptotic notation. Let $f(n)$ and $g(n)$ be two positive functions. We say $f(n) = O(g(n))$ if there exist two fixed positive constants c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$, and $f(n) = o(g(n))$ if for any arbitrarily positive constant c, there exists a positive constant n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$. We say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n)\log^c g(n))$ for some fixed constant c. Let $\mathrm{poly}(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant c. We let $\mathrm{negl}(n)$ denote a function $f(n)$ 'negligible' in n such that $f(n) = o(n^{-c})$ for any constant c. We say that an event happens with overwhelming probability if it happens with a probability of at least $1 - \mathrm{negl}(n)$.

Let $X = \{X(n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$ denote a binary distribution ensemble (i.e., an infinite set of probability distributions over $\{0, 1\}$), where a distribution $X(n, z)$ is associated with each security parameter $n \in \mathbb{N}$ and input $z \in \{0, 1\}^*$. Now we have two such ensembles $X = \{X(n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$ and $Y = \{Y(n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$. If $|\Pr(X(n, z) = 1) - \Pr(Y(n, z) = 1)| \leq \mathrm{negl}(n)$, we say that X and Y are indistinguishable, denoted by $X \approx Y$.

2.2 Oblivious transfer

Oblivious transfer was first introduced by Rabin (1981) who also proposed an OT construction based on the RSA cryptosystem. Then, a more useful form called '1-out-of-2 OT' (Even *et al.*, 1985) was developed to build protocols for secure multiparty computation. 1-out-of-2 OT is an execution of a two-party computation, where the sender (denoted by S) takes as input a message pair (m_0, m_1) and the receiver (denoted by R) chooses one bit $\sigma \in \{0, 1\}$ as its input. It requires that R will receive m_σ without knowing $m_{1-\sigma}$ and S has no knowledge about the receiver's choice after the execution of the protocol. Moreover, 1-out-of-N OT can be viewed as a generalization of 1-out-of-2 OT, where the sender S runs a database with N messages and the receiver R obliviously chooses one message from that database.

2.3 Lattice-based cryptography

Lattice-based cryptography is a promising post-quantum cryptography candidate since it has strong provable security guarantees and good asymptotic efficiency. As a versatile primitive for lattice-based cryptographic constructions, the learning with errors (LWE) problem (Regev, 2005) is widely used and regarded as quantum resistant due to its provable property that solving the average-case LWE problem is at least as hard as solving some standard worst-case lattice problem which cannot be efficiently solved by quantum algorithms. Referring to Regev (2005), the LWE problem is defined as follows:

Definition 1 (LWE) Let $n \geq 1$ and $q \geq 2$ be positive integers, χ an error distribution over $\mathbb{Z}_q$, and $\boldsymbol{s}$ a secret vector following the uniform distribution over $\mathbb{Z}_q^n$. Let $A_{\boldsymbol{s},\chi}$ denote the probability distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ, and returning $(\boldsymbol{a}, c) = (\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The LWE problem has two versions: search-version and decision-version. Search-LWE finds $\boldsymbol{s}$ given access to an arbitrary number of independent samples $(\boldsymbol{a}, c)$ from $A_{\boldsymbol{s},\chi}$. Decision-LWE distinguishs an oracle which returns independent samples from $A_{\boldsymbol{s},\chi}$ from an oracle which returns independent samples from the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Regev (2005) gave a quantum reduction from worst-case lattice problems to the search-LWE problem. Moreover, he established the equivalences of the search-LWE problem using elementary reductions, including a reduction from the (average-case) decision-version to the search-version. In the rest of this paper we denote this decision-LWE problem as the LWE problem. Here we state the result in Regev (2005) with regard to the average-case decision-LWE problem, which is denoted by $\mathrm{LWE}_{q,\chi}$ and used in the following cryptographic application (Section 3) as underlying hardness. That is, for certain choices of q and χ, we have:

Proposition 1 (Theorem 1.1 and Lemma 4.2 in Regev (2005)) Let n and q be integers and $\alpha = \alpha(n) \in (0,1)$ be such that $\alpha q > 2\sqrt{n}$. If there exists an efficient algorithm that can solve $\mathrm{LWE}_{q,\chi}$, then there exists an efficient quantum algorithm for solving the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP) within $\tilde{O}(n/\alpha)$ in the worst case.

2.4 Classical UC framework

Since the OT protocol that we intend to analyze in the quantum setting is proven to be secure in the standard UC framework proposed by Canetti (2001), we first give an overview of this model. The security in this model is defined by comparison: given a certain protocol task, e.g., a secure OT, we assume that there exists a specific machine, called the ideal functionality $\mathcal{F}_{\mathrm{OT}}$, which can securely implement this task. Two more involved parties S and R (called 'dummy parties') forward only their inputs (m_0, m_1) and σ to $\mathcal{F}_{\mathrm{OT}}$, and $\mathcal{F}_{\mathrm{OT}}$ will return m_σ to R and nothing to S. All communication between $\mathcal{F}_{\mathrm{OT}}$ and these two parties is performed over secure channels, and their interactions can be regarded as an ideal protocol (denoted by $\rho^{\mathcal{F}_{\mathrm{OT}}}$) which can securely realize OT.

However, in the real world we cannot obtain such an ideal protocol $\rho^{\mathcal{F}_{\mathrm{OT}}}$. We need a real protocol π to implement $\mathcal{F}_{\mathrm{OT}}$ and intuitively expect that π is at least as secure as $\mathcal{F}_{\mathrm{OT}}$. It implies that if there exists an adversary that can do something during the execution of π, then the adversary can do the same thing during an execution using $\mathcal{F}_{\mathrm{OT}}$. Due to the ideal security of $\mathcal{F}_{\mathrm{OT}}$, this adversary cannot successfully attack π either. In other words, we hope to build an ideal adversary in the execution of $\mathcal{F}_{\mathrm{OT}}$ using the power of the real adversary in the execution of π. If such an ideal adversary cannot be successful, then neither can the real adversary. The security is captured by comparing the execution of π to the execution of $\mathcal{F}_{\mathrm{OT}}$ within the existence of the adversary. That is to say, we compare what can be learned by a real adversary who receives the real information to what can be learned by an ideal adversary who receives nothing (due to the security of ideal functionality, the ideal adversary learns nothing). If they can both learn approximately the same amount of information (i.e., look indistinguishable), then the security of the real protocol is guaranteed.

Formally, it requires that for any adversary Adv, there is another adversary Sim called the 'simulator', such that an execution of π with Adv (called the 'real world') is indistinguishable from an execution of $\mathcal{F}_{\mathrm{OT}}$ with Sim (called the 'ideal world'). Moreover, in the UC framework, there is a new algorithmic entity called 'environment' $\mathcal{Z}$, which can freely interact with both worlds during the course of execution. If no machine $\mathcal{Z}$ can guess whether it is interacting with the real world or the ideal world, then we say this real protocol π can securely UC-emulate $\mathcal{F}_{\mathrm{OT}}$ or the ideal protocol $\rho^{\mathcal{F}_{\mathrm{OT}}}$; i.e., π can achieve UC-security.

Referring to Canetti (2001), let $\mathcal{F}$ be an ideal functionality. $\mathrm{EXEC}_{\pi,\mathrm{Adv},\mathcal{Z}}(n,z)$ denotes the random variable describing the 1-bit output of $\mathcal{Z}$ when interacting with Adv and running π on an input $z \in \{0,1\}^*$, where n represents the security parameter. Likewise, the random variable describing the single bit output of $\mathcal{Z}$ when interacting with $\mathcal{F}$ and Sim on the same input z is denoted by $\mathrm{IDEAL}_{\mathcal{F},\mathrm{Sim},\mathcal{Z}}(n,z)$. Let $\mathrm{EXEC}_{\pi,\mathrm{Adv},\mathcal{Z}}$ and $\mathrm{IDEAL}_{\mathcal{F},\mathrm{Sim},\mathcal{Z}}$ denote the ensembles $\{\mathrm{EXEC}_{\pi,\mathrm{Adv},\mathcal{Z}}(n,z)\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\{\mathrm{IDEAL}_{\mathcal{F},\mathrm{Sim},\mathcal{Z}}(n,z)\}_{n\in\mathbb{N},z\in\{0,1\}^*}$, respectively, of distributions over $\{0,1\}$. Then the general notion that π can securely UC-emulate $\mathcal{F}$ is formally defined as follows:

Definition 2 (UC-security) Let $\mathcal{F}$ be an ideal functionality. A protocol π is said to UC-emulate $\mathcal{F}$ if, for any adversary Adv, there exists a simulator Sim such that for all environments $\mathcal{Z}$ it holds that

$$\mathrm{IDEAL}_{\mathcal{F},\mathrm{Sim},\mathcal{Z}} \approx \mathrm{EXEC}_{\pi,\mathrm{Adv},\mathcal{Z}},$$

where '$\approx$' denotes indistinguishability. That is to say, π achieves UC-security.

Note that all machines involved in the above classical UC framework are modeled as interactive Turing machines (ITMs). When we restrict the adversary, simulator, and environment as classical polynomial-time machines, the notion of computational classical-UC-security will be derived from the above framework. Likewise, when we allow unbounded adversary, simulator, and environment, the notion of statistical classical-UC-security can be achieved. If we further require the complexities of Sim and Adv be polynomial-time related in the statistical setting, then the statistical UC-security will imply computational UC-security (see Canetti (2001) for details).

2.5 Quantum universal composability framework

A quantum analogue of Canetti's standard UC framework was proposed by Unruh (2010). In this quantum UC framework, all computation machines are modeled as quantum interactive machines (QIMs). For illustration, we briefly give an overview of the quantum machine model defined in the quantum UC model (Unruh, 2010), which presents the work pattern between QIMs and aims to mimic the counterpart in the classical UC model. Here we first introduce some basic notions of quantum information theory and refer the reader to Nielsen and Chuang (2010) for a detailed treatment.

A state in a quantum system is described by a vector $|\psi\rangle$ in a Hilbert space (a complex vector space with inner product) of the form $\mathcal{H}=\mathbb{C}^N$, where N is a countable set and $N=\{0,1\}$ for qubit or $N=\{0,1\}^*$ for multiple qubits. A joint quantum system is defined by the tensor product of several separate Hilbert spaces $\mathbb{C}^{N_1\times N_2\times\cdots\times N_n}=\mathcal{H}_1\otimes\mathcal{H}_2\otimes\cdots\otimes\mathcal{H}_n$, where $\mathcal{H}_i=\mathbb{C}^{N_i}$ for all $i\in[1,n]$ and $n\in\mathbb{N}$. Usually, we assume an orthonormal basis, i.e., the computation basis $\{|x\rangle:x\in N\}$ for each Hilbert space. Then the quantum state $|\psi\rangle$ can be represented as a state vector $|\psi\rangle=\sum_i\alpha_i|x_i\rangle$, where the measurement outcome $x_i\in N$ occurs with probability $|\alpha_i|^2$ following the normalization condition that $\sum_{i\in N}|\alpha_i|^2=1$. In addition, we use $\langle\psi|$ to represent a linear transformation mapping $|\varphi\rangle$ to the inner product $\langle\psi|\varphi\rangle$. $|\psi\rangle\langle\psi|$ denotes the orthogonal projector on $|\psi\rangle$. If a quantum system is not in a single pure state as defined above, but in a mixture of different pure states $|\psi_i\rangle\in\mathcal{H}$ with respective probability p_i, we use a density operator to describe this quantum system in the form of $\rho=\sum_i p_i|\psi_i\rangle\langle\psi_i|$ over $\mathcal{H}$. Note that this formulation is mathematically equivalent to the state vector representation. When a quantum system is in the pure state case, its density operator representation is simply $\rho=|\psi\rangle\langle\psi|$. In this study, we describe quantum states in the density operator representation.

Let $\mathcal{P}(\mathcal{H})$ denote the set of all density operators on $\mathcal{H}$, i.e., positive operators with trace 1 (see Theorem 2.5 in Nielsen and Chuang (2010) for the characteristic of the density operator). A composed system can be denoted by operators in $\mathcal{P}(\mathcal{H}_1\otimes\mathcal{H}_2\otimes\cdots\otimes\mathcal{H}_n)$. Any operation on a quantum system can be described by a superoperator (i.e., quantum operations). We say a map $\mathcal{E}:\mathcal{P}(\mathcal{H})\to\mathcal{P}(\mathcal{H})$ is a superoperator on $\mathcal{P}(\mathcal{H})$ if and only if $\mathcal{E}$ is a completely positive trace preserving map. That is to say, $\mathcal{E}(\rho)$ is positive (Hermitian with nonnegative eigenvalues) for all positive operators ρ over $\mathcal{H}$.

In this work, we perform measurements in the computational basis. Given a state $\rho\in\mathcal{P}(\mathcal{H})$ let $\rho_x=(|x\rangle\langle x|)\rho(|x\rangle\langle x|)$. Then the outcome of measuring ρ over $\mathcal{H}$ in the computational basis is x with probability $\mathrm{tr}(\rho_x)$ (i.e., the trace of ρ_x), and the resulting quantum state after measuring is $\rho_x/\mathrm{tr}(\rho_x)$. In addition, we can represent a classical state by a density operator $\rho=\sum_{x\in N}P(x)|x\rangle\langle x|\in\mathcal{P}(\mathbb{C}^N)$ that corresponds to a classical probability distribution $P:N\to[0,1]$, where $\{|x\rangle\}$ represents the computational basis. Accordingly, we can define a classical superoerator $\mathcal{E}:\mathcal{P}(\mathbb{C}^{N_1})\to\mathcal{P}(\mathbb{C}^{N_2})$ by the form $\mathcal{E}(\rho)=\sum_{\substack{x\in N_1\\y\in N_2}}\Pr(F(x)=y)\cdot\langle x|\rho|x\rangle\cdot|y\rangle\langle y|$ corresponding to a classical randomized function $F:N_1\to N_2$.

2.5.1 Machine model

In the classical UC framework, all computation elements are modeled as ITMs, denoted by M, while in the quantum setting they are modeled by QIMs, denoted by $\hat{M}$, where '$\hat{}$' represents a machine with quantum power.

A QIM is represented by an identity $\mathrm{id}_{\hat{M}}\in\{0,1\}^*$ and a sequence of superoperators $\mathcal{E}_{\hat{M}}^{(n)}$ on the tensor product of three Hilbert spaces $\mathcal{H}^{\mathrm{state}}\otimes\mathcal{H}^{\mathrm{class}}\otimes\mathcal{H}^{\mathrm{quant}}$, where $\mathcal{H}^{\mathrm{state}},\mathcal{H}^{\mathrm{class}},\mathcal{H}^{\mathrm{quant}}:=\mathbb{C}^{\{0,1\}^*}$. Here $\mathcal{H}^{\mathrm{state}}$, $\mathcal{H}^{\mathrm{class}}$, and $\mathcal{H}^{\mathrm{quant}}$ can be viewed as three registers. $\mathcal{H}^{\mathrm{state}}$ represents the state kept by $\hat{M}$ between invocations. $\mathcal{H}^{\mathrm{class}}$ and $\mathcal{H}^{\mathrm{quant}}$ store incoming and outgoing messages with the classical part stored in $\mathcal{H}^{\mathrm{class}}$ and the quantum message part stored in $\mathcal{H}^{\mathrm{quant}}$.

If a QIM $\mathrm{id}_{\mathrm{sender}}$ wants to send a message consisting of the classical part m and the quantum part $|\psi\rangle$ to another QIM $\mathrm{id}_{\mathrm{receiver}}$, $\mathrm{id}_{\mathrm{sender}}$ first initializes $\mathcal{H}^{\mathrm{class}}$ with $(\mathrm{id}_{\mathrm{sender}},\mathrm{id}_{\mathrm{receiver}},m)$ and $\mathcal{H}^{\mathrm{quant}}$ with $|\psi\rangle$. When we say that we initialize $\mathcal{H}$ with m, it means that we apply a superoperator $\mathcal{E}_{\mathrm{init}}^m$ on $\mathcal{P}(\mathcal{H})$, where $\mathcal{E}_{\mathrm{init}}^m(\rho)=|m\rangle\langle m|$. Similarly, it is applicable to the initialization with quantum part $|\psi\rangle$. If a QIM does not want to send a message, it initializes $\mathcal{H}^{\mathrm{class}}$ and $\mathcal{H}^{\mathrm{quant}}$ with an empty word.

Each QIM can be regarded as a special classical ITM which can perform classical linear operations. For example, we denote $\mathcal{E}_{\text{class}} : \rho \to \sum_x \langle x|\rho|x\rangle \cdot |x\rangle\langle x|$ as a classical superoperator in $\mathcal{E}_{\hat{M}}^{(n)}$ which measures the density operator ρ in the computational basis, thus removing all superpositions from ρ.

In the security proof, the corruption scenario is necessarily considered, where an adversary can corrupt some parties to control their current status and behave in the name of them. We denote a corrupted party with identity id by P_{id}^C, where C represents a corruption set that contains all identities of corrupted parties. The corrupted party P_{id}^C can be viewed as a special machine instructed by a quantum adversary, denoted by $\widehat{\text{Adv}}$. Once it is invoked, P_{id}^C will measure $\mathcal{H}^{\text{class}}$ first and parse the outcome as $(\text{id}_{\text{sender}}, \text{id}_{\text{receiver}}, m)$. If $\text{id}_{\text{sender}}$ is $\widehat{\text{Adv}}$, $\mathcal{H}^{\text{class}}$ is initialized with m. Otherwise, the message is forwarded to adversary $\widehat{\text{Adv}}$.

In addition, a notion called 'network', denoted by N, represents a set of machines with pairwise distinct identities containing a QIM $\hat{\mathcal{Z}}$ with identity $\text{id}_{\hat{\mathcal{Z}}}$. Given a network N and a corruption set C, N^C represents the network by substituting machines $\hat{M} \in N$ with their identities $\text{id} \in C$ with P_{id}^C.

2.5.2 Quantum-UC-security

Let π be a real protocol, and ρ be an ideal protocol which can securely realize some specific functionality. We denote parties_π as the set of all machines running in protocol π. From the above description of the quantum machine model, we can accordingly obtain the following two variants of quantum-UC-security in the quantum UC framework:

Definition 3 (Statistical quantum-UC-security, Definition 3 in Unruh (2010)) We say that π statistically quantum-UC-emulates ρ if and only if for every set $C \subseteq \text{parties}_\pi$ and for every adversary $\widehat{\text{Adv}}$, there is a simulator $\widehat{\text{Sim}}$ such that for every environment $\hat{\mathcal{Z}}$, two networks $\pi^C \cup \{\widehat{\text{Adv}}, \hat{\mathcal{Z}}\}$ (called the 'real world') and $\rho^C \cup \{\widehat{\text{Sim}}, \hat{\mathcal{Z}}\}$ (called the 'ideal world') are indistinguishable. That is to say, π achieves statistical quantum-UC-security. Furthermore, we require that if $\widehat{\text{Adv}}$ is quantum polynomial-time, so is $\widehat{\text{Sim}}$.

Definition 4 (Computational quantum-UC-security, cf. Definition 4 in Unruh (2010)) We say that π computationally quantum-UC-emulates ρ if and only if for every set $C \subseteq \text{parties}_\pi$ and for every quantum polynomial-time adversary $\widehat{\text{Adv}}$, there is a quantum polynomial-time simulator $\widehat{\text{Sim}}$ such that for every quantum polynomial-time environment $\hat{\mathcal{Z}}$, two networks $\pi^C \cup \{\widehat{\text{Adv}}, \hat{\mathcal{Z}}\}$ (called the 'real world') and $\rho^C \cup \{\widehat{\text{Sim}}, \hat{\mathcal{Z}}\}$ (called the 'ideal world') are indistinguishable. That is to say, π achieves computational quantum-UC-security.

Remark 1 The statement that two networks $\pi^C \cup \{\widehat{\text{Adv}}, \hat{\mathcal{Z}}\}$ and $\rho^C \cup \{\widehat{\text{Sim}}, \hat{\mathcal{Z}}\}$ are indistinguishable implies that, for all $z \in \{0,1\}^*$ and $n \in \mathbb{N}$, $|\Pr(\text{EXEC}_{\pi^C \cup \{\widehat{\text{Adv}}, \hat{\mathcal{Z}}\}}(n, z) = 1) - \Pr(\text{EXEC}_{\rho^C \cup \{\widehat{\text{Sim}}, \hat{\mathcal{Z}}\}}(n, z) = 1)| \leq \text{negl}(n)$.

3 Lattice-based oblivious transfer

In this section, we introduce an existing lattice-based OT protocol (Peikert *et al.*, 2008) which will be analyzed in the quantum setting (Section 4). This lattice-based OT protocol is extracted from a dual-mode cryptosystem proposed by Peikert *et al.* (2008), which can be instantiated with several standard theoretic assumptions, including the decisional Diffie-Hellman problem, the quadratic residuosity problem, and the worst-case lattice assumption. Once such a dual-mode cryptosystem under certain standard assumptions is built well, an efficient and UC-secure OT protocol based on the same hardness assumption can be directly derived. In Section 3.1, we first present the dual-mode cryptosystem in lattice setting, i.e., an instantiation with LWE hardness. Then we introduce the corresponding OT protocol derived from this LWE-based dual-mode cryptosystem in Section 3.2. In addition, for a clear quantum analysis (i.e., UC-security proof in quantum setting) of this LWE-based OT protocol shown in Section 4, we need to recall the simulator constructions (the main ingredients of (classical) UC-security proof) under different corruptions in Section 3.3.

3.1 Dual-mode cryptosystem based on LWE hardness

A dual-mode cryptosystem behaves like a normal encryption scheme which can correctly decrypt a message encrypted with a given public key by the corresponding secret key. The main feature of this encryption scheme is that it can operate in two modes, i.e., messy mode and decryption mode. The

mode in which the scheme runs is determined by a trusted setup phase of this dual-mode construction. In the trusted setup phase, a common reference string (denoted by crs) and the corresponding trapdoor information (τ) are created, where crs may be chosen, either uniformly at random or from a specified distribution. The distribution of crs decides the mode in which the dual-mode cryptosystem operates and specifies the type of public key used in the encryption.

The instantiation of the dual-mode cryptosystem with LWE hardness relies on some techniques developed by Gentry *et al.* (2008), including an LWE-based encryption which is a variant of Regev's CPA-secure LWE-based cryptosystem (Regev, 2005) and an efficient algorithm called IsMessy. Gentry *et al.* (2008) showed how to securely embed a trapdoor into the public matrix $\boldsymbol{A}$ of this LWE-based encryption cryptosystem, so that when this trapdoor information is given, the algorithm IsMessy can efficiently identify the messy public key, which has the property that a ciphertext produced by the messy public key carries no information (statistically) about the encrypted message.

3.1.1 LWE-based encryption

Here we first introduce this LWE-based encryption, where the message space is $\mathbb{Z}_2=\{0,1\}$. Let the modulus $q=\text{poly}(n)$ be a large prime. For every message $\mu\in\mathbb{Z}_2$, the center of μ is defined as $t(\mu)=\mu\cdot\lfloor q/2\rfloor\in\mathbb{Z}_q$. Let χ denote an error distribution over $\mathbb{Z}_q$ and $D_{\mathbb{Z}^m,r}$ a Gaussian-like distribution over $\mathbb{Z}^m$ with standard deviation approximately r. All operations are performed over $\mathbb{Z}_q$.

1. LWEKeyGen(1^n): choose $\boldsymbol{A}\leftarrow\mathbb{Z}_q^{n\times m}$ under a distribution statistically close to uniform, together with a trapdoor called $\boldsymbol{S}$ (Gentry *et al.*, 2008). Choose a secret key $\boldsymbol{s}\leftarrow\mathbb{Z}_q^{n\times 1}$ uniformly at random. To generate the public key, choose an error vector $\boldsymbol{x}\in\mathbb{Z}_q^{1\times m}$ according to the error distribution $\chi=\bar{\Psi}_\alpha$ for some parameter $\alpha=\alpha(n)\in(0,1)$ (here x_i is chosen independently for all $i\in[m]$ and m denotes the number of samples). Then compute $(\boldsymbol{A},\boldsymbol{p}=\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{x})$, where each entry $(\boldsymbol{a}_i,p_i)$ is a sample from $A_{s,\chi}$.

2. LWEEnc($\mathbf{pk}=(\boldsymbol{A},\boldsymbol{p}),\mu$): to encrypt a message $\mu\in\mathbb{Z}_2$, choose a vector $\boldsymbol{e}\in\mathbb{Z}^m$ from $D_{\mathbb{Z}^m,r}$ and output the ciphertext $(\boldsymbol{u},c)=(\boldsymbol{A}\boldsymbol{e},\boldsymbol{p}\boldsymbol{e}+t(\mu))=(\boldsymbol{A}\boldsymbol{e},\boldsymbol{p}\boldsymbol{e}+\mu\cdot\lfloor q/2\rfloor)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$.

3. LWEDec($\mathbf{sk}=\boldsymbol{s},(\boldsymbol{u},c)$): to recover message μ from the ciphertext, compute $d=c-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}\in\mathbb{Z}_q$. Output 0 if d is closer to 0 than to $\lfloor q/2\rfloor$ modulo q; otherwise, output 1.

Remark 2 The above LWE-based encryption scheme can also be applied when the message is a vector $\boldsymbol{\mu}\in\mathbb{Z}_2^\ell$, where $\ell=\text{poly}(n)\geq 1$. The main advantage that matrix $\boldsymbol{A}$ can be reused over ℓ different bits enables the fulfillment of a multi-session OT protocol, where it can implement ℓ individual OT executions using the same matrix $\boldsymbol{A}$ and obliviously transfer one bit to the receiver in each subsession.

3.1.2 Messy public keys (cf. Gentry *et al.* (2008), Section 6).

For an arbitrary fixed public key $(\boldsymbol{A},\boldsymbol{p})$, we define $\delta(\boldsymbol{p})$ to be the statistical distance between the uniform distribution over $\mathbb{Z}_q^n\times\mathbb{Z}_q$ and the distribution of $(\boldsymbol{A}\boldsymbol{e},\boldsymbol{p}\boldsymbol{e})$, where $\boldsymbol{e}\leftarrow D_{\mathbb{Z}^m,r}$. If $\delta(\boldsymbol{p})$ is negligible, we say $\boldsymbol{p}$ is a messy public key. It implies that the ciphertext produced by $\boldsymbol{p}$ will lose essentially all information (statistically) about the encrypted bit; i.e., LWEEnc($(\boldsymbol{A},\boldsymbol{p}),0$) and LWEEnc($(\boldsymbol{A},\boldsymbol{p}),1$) are statistically close since both distributions are statistically close to uniform. However, the correctness of LWEDec implies that a key $\boldsymbol{p}$ generated by LWEKeyGen typically has large $\delta(\boldsymbol{p})$. It means that when encrypting messages under the keys generated by LWEKeyGen, the encrypted data is computationally hidden. For a large enough m (the number of LWE samples), a public key $(\boldsymbol{A},\boldsymbol{p})$ chosen uniformly at random over $\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^{1\times m}$ is messy with overwhelming probability. There is no efficient adversary which can distinguish between messy public keys and the public keys generated by LWEKeyGen. However, if the corresponding trapdoor information about $\boldsymbol{A}$ is given, the algorithm IsMessy can identify messy public keys by taking $(\tau,\boldsymbol{p})$ as input, and then output messy if $\boldsymbol{p}$ is a messy public key.

3.1.3 LWE-based dual-mode cryptosystem

Now we introduce the LWE-based dual-mode cryptosystem constructed by the above LWE-based encryption and the messy key identification algorithm IsMessy. For simplicity, we present only this LWE-based dual-model cryptosystem for $\ell=1$.

1. SetupMessy(1^n): choose a matrix $\boldsymbol{A}\leftarrow\mathbb{Z}_q^{n\times m}$ uniformly at random, together with the trapdoor

information $\tau=(\boldsymbol{S},\boldsymbol{A})$. For each $b\in\{0,1\}$, choose a vector $\boldsymbol{v}_b\leftarrow\mathbb{Z}_q^{1\times m}$ uniformly at random. Let $\mathrm{crs}=(\boldsymbol{A},\boldsymbol{v}_0,\boldsymbol{v}_1)$. Output (crs,τ).

2. SetupDec(1^n): choose a matrix $\boldsymbol{A}\leftarrow\mathbb{Z}_q^{n\times m}$ and a vector $\boldsymbol{w}\leftarrow\mathbb{Z}_q^{1\times m}$ uniformly at random. For each $b\in\{0,1\}$, choose a secret $\boldsymbol{s}_b\leftarrow\mathbb{Z}_q^n$ uniformly at random and an error vector $\boldsymbol{x}_b\leftarrow\chi^{1\times m}$, where all entries are chosen independently from the error distribution χ. Let $\boldsymbol{v}_b=\boldsymbol{s}_b^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{x}_b-\boldsymbol{w}$, $\mathrm{crs}=(\boldsymbol{A},\boldsymbol{v}_0,\boldsymbol{v}_1)$, and $\tau=(\boldsymbol{w},\boldsymbol{s}_0,\boldsymbol{s}_1)$. Output (crs,τ).

3. KeyGen(crs,σ): choose a secret $\boldsymbol{s}\leftarrow\mathbb{Z}_q^n$ uniformly at random and a vector $\boldsymbol{x}\leftarrow\chi^{1\times m}$. Let $\mathbf{pk}=\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{x}-\boldsymbol{v}_\sigma$, where $\sigma\in\{0,1\}$ denotes the decryptable branch. Let $\mathbf{sk}=\boldsymbol{s}$ and output $(\mathbf{pk},\mathbf{sk})$.

4. Enc($\mathbf{pk},b,\mu$): output $y\leftarrow$LWEEnc($(\boldsymbol{A},\boldsymbol{p}=\mathbf{pk}+\boldsymbol{v}_b),\mu$), where $b\in\{0,1\}$ is chosen by the encrypter. Here y is pair $(\boldsymbol{u},c)$.

5. Dec($\mathbf{sk},y$): output $\mu\leftarrow$LWEDec($\mathbf{sk}=\boldsymbol{s},(\boldsymbol{u},c)$).

6. FindMessy($\tau,\mathbf{pk}$): parse τ as $(\boldsymbol{S},\boldsymbol{A})$, run IsMessy($(\boldsymbol{S},\boldsymbol{A}),\mathbf{pk}+\boldsymbol{v}_b$) for each $b\in\{0,1\}$, and output b such that IsMessy can output messy on at least one branch correctly with overwhelming probability.

7. TrapdoorKeyGen(τ): parse τ as $(\boldsymbol{w},\boldsymbol{s}_0,\boldsymbol{s}_1)$, and output $(\mathbf{pk},\mathbf{sk}_0,\mathbf{sk}_1)$=$(\boldsymbol{w},\boldsymbol{s}_0,\boldsymbol{s}_1)$.

Remark 3 SetupMessy generates crs and trapdoor $\tau=(\boldsymbol{S},\boldsymbol{A})$, where $\boldsymbol{S}$ is an embedded trapdoor in $\boldsymbol{A}$ so that IsMessy can be used in FindMessy to reveal whether $\mathbf{pk}+\boldsymbol{v}_b$ is a messy key for each $b\in\{0,1\}$ with the given trapdoor τ.

The dual-mode cryptosystem has three required security properties: (1) in messy mode, for each base public key $\mathbf{pk}$, at least one of the derived public keys $\mathbf{pk}+\boldsymbol{v}_b$ for $b\in\{0,1\}$ can statistically hide its encrypted message; (2) in decryption mode, the honest receiver's chosen bit σ is statistically hidden by its choice of base key $\mathbf{pk}$; (3) given crs, no adversary can efficiently distinguish two modes (i.e., satisfying computational indistinguishability). These security properties guarantee that a UC-secure OT protocol can be derived from this dual-mode cryptosystem.

However, this LWE-based dual-mode cryptosystem satisfies only a slightly relaxed version of the above security requirements. Fortunately, a UC-secure OT protocol based on LWE hardness can still be derived from this relaxed LWE-based dual-mode cryptosystem, although it leads to a slightly weaker security of the honest receiver when running in decryption mode, i.e., computational security.

3.2 LWE-based oblivious transfer

Once this LWE-based dual-mode cryptosystem is built well, an LWE-based OT protocol denoted by $\mathrm{dm}^{\mathrm{mode}}$ can be derived directly (Fig. 1). This $\mathrm{dm}^{\mathrm{mode}}$ can securely UC-emulate a multi-session OT functionality (denoted by $\widetilde{\mathcal{F}}_{\mathrm{OT}}$) in the common reference string model (Canetti, 2001), where two computation parties query a trusted party for a common reference string crs. Here $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ serves as a shell around a bounded number of independent $\mathcal{F}_{\mathrm{OT}}$ executions, where it specifies the interaction with two parties in a single session by sid and coordinates each subsession included in a single session by ssid. Let $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$ denote the common reference string functionality, where it produces crs for two parties by running PPT algorithm $\mathcal{D}$ before the interaction between two parties. Note that $\mathrm{dm}^{\mathrm{mode}}$ can also operate in two modes by two instantiations of $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$: when $\mathcal{D}=$ SetupMessy (i.e., $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}=\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$), $\mathrm{dm}^{\mathrm{mode}}$ runs in the messy mode (i.e., $\mathrm{dm}^{\mathrm{mode}}=\mathrm{dm}^{\mathrm{mes}}$); when $\mathcal{D}=$ SetupDec (i.e., $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}=\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$), $\mathrm{dm}^{\mathrm{mode}}$ runs in the decryption mode (i.e., $\mathrm{dm}^{\mathrm{mode}}=\mathrm{dm}^{\mathrm{dec}}$). The procedure of $\mathrm{dm}^{\mathrm{mode}}$ is described in Fig. 1. Note that protocol $\mathrm{dm}^{\mathrm{mode}}$ is a UC-secure 1-out-of-2 OT and it can be easily extended as a 1-out-of-2^k OT protocol by choosing $\boldsymbol{v}_b$ for each $b\in\{0,1\}^k$.

3.3 Construction of the simulator

As shown in Peikert *et al.* (2008), $\mathrm{dm}^{\mathrm{mode}}$ is proven to be secure against static corruptions in the standard UC model. Static corruption means that the adversary can decide only which parties will be corrupted before the execution of the protocol instead of during the course of protocol execution. Then the adversary will control all the behavior of the corrupted party when interacting with the honest parties. Referring to Section 2.4, the obtained UC-security of $\mathrm{dm}^{\mathrm{mode}}$ implies that for any real adversary Adv, there exists an ideal adversary Sim interacting with the ideal functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$, such that no machine $\mathcal{Z}$ can distinguish its interaction with Adv in an execution of $\mathrm{dm}^{\mathrm{mode}}$ from an interaction with Sim using $\widetilde{\mathcal{F}}_{\mathrm{OT}}$.

Since the UC-security proof of $\mathrm{dm}^{\mathrm{mode}}$ is proven against the static corruptions, four different

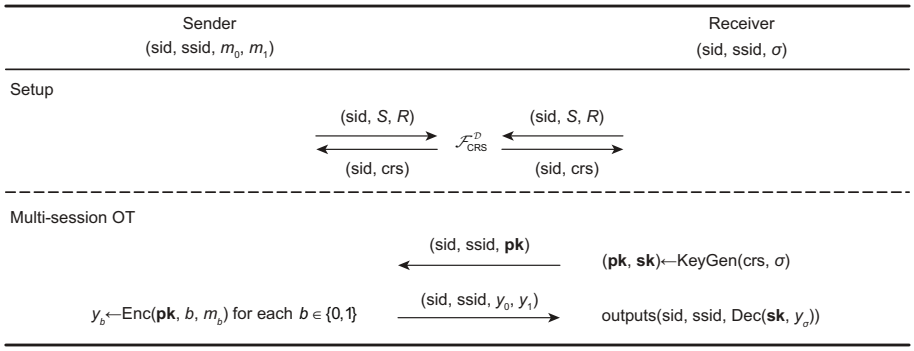

Fig. 1 Protocol $\mathrm{dm}^{\mathrm{mode}}$ for oblivious transfer

corruption cases should be considered in each mode:

Case 1: Only the receiver is corrupted;

Case 2: Only the sender is corrupted;

Case 3: Both parties are corrupted;

Case 4: Neither party is corrupted.

Therefore, we recall the constructions of Sim under these four corruption cases, respectively. As folklore, the non-trivial parts of the UC-security proof lie in the constructions of Sim when only the receiver or the sender is corrupted. The security under another two corruptions can be achieved trivially.

In Section 4, we will give the quantum security analysis of $\mathrm{dm}^{\mathrm{mode}}$ by modifying machines step by step, i.e., changing all adversarial ITMs into QIMs. For clarity, when we show that the security of $\mathrm{dm}^{\mathrm{mode}}$ in the classical setting can be preserved in the quantum setting, we divide the construction of two simulators (i.e., under corruptions cases 1 and 2) into four steps, denoted by $\mathrm{Sim}(S1, S2, S3, S4)$.

When we look into the UC-security proof of $\mathrm{dm}^{\mathrm{mode}}$, each participant involved in the proof is modeled by an ITM which has its own input and output tapes for recording the incoming value and outgoing value from and to its outer environment. In each corruption case, Sim starts by running a copy of Adv. Every incoming value that Sim receives from $\mathcal{Z}$ is written into Adv's input tape. Every outgoing value written by Adv on its output tape is copied to Sim's output tape. Note that regardless of the mode in which the protocol runs, the construction of Sim differs only according to the current corruption case.

3.3.1 When only R is corrupted

$S1$: Sim runs the messy mode setup algorithm and generates $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupMessy}(1^n)$. Namely, Sim chooses a matrix $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$ uniformly at random, together with a trapdoor $\tau = (\boldsymbol{S}, \boldsymbol{A})$. For each $b \in \{0,1\}$, Sim selects a vector $\boldsymbol{v}_b \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random, sets $\mathrm{crs} = (\boldsymbol{A}, \boldsymbol{v}_0, \boldsymbol{v}_1)$, and outputs (crs, τ). When the parties start to query $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$, Sim returns $(\mathrm{sid}, \mathrm{crs})$ to them and stores τ privately.

$S2$: Adv produces a message $(\mathrm{sid}, \mathrm{ssid}, \mathbf{pk})$, where $\mathbf{pk} = \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{x} - \boldsymbol{v}_\sigma$, and sends it to Sim. This communication between Adv and Sim can be regarded as a message produced by $\mathcal{Z}$ if Adv is dummy (denoted by $\mathrm{Adv}_{\mathrm{dummy}}$); i.e., Adv is fully controlled by $\mathcal{Z}$ and works as the communication channel for delivering messages between $\mathcal{Z}$ and honest parties.

$S3$: Sim runs $\mathrm{FindMessy}(\mathrm{crs}, \tau, \mathbf{pk})$ to find b which specifies the messy branch, and queries the ideal functionality $\tilde{\mathcal{F}}_{\mathrm{OT}}$ with $(\mathrm{sid}, \mathrm{ssid}, \mathrm{receiver}, 1-b)$ in the name of R. Then Sim receives output $(\mathrm{sid}, \mathrm{ssid}, m_{1-b})$ from $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ and stores (b, m_{1-b}).

$S4$: When S is activated on some subsession $(\mathrm{sid}, \mathrm{ssid})$, then Sim must play the role of the sender to interact with Adv as an execution of the real world. Sim first looks up the corresponding (b, m_{1-b}), and then computes $y_{1-b} \leftarrow \mathrm{Enc}(\mathbf{pk}, 1-b, m_{1-b})$ and $y_b \leftarrow \mathrm{Enc}(\mathbf{pk}, b, 0^\ell)$. Finally, Sim sends message $(\mathrm{sid}, \mathrm{ssid}, y_0, y_1)$ to Adv as if it were from S.

3.3.2 When only S is corrupted

$S1$: Sim runs the decryption mode setup algorithm and generates $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupDec}(1^n)$. Namely, Sim chooses a matrix $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and a vector $\boldsymbol{w} \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random. For each $b \in \{0,1\}$, Sim chooses a secret vector $\boldsymbol{s}_b \leftarrow \mathbb{Z}_q^n$ uniformly at random and an error vector $\boldsymbol{x}_b \leftarrow \chi^{1\times m}$, and then sets $\boldsymbol{v}_b = \boldsymbol{s}_b^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{x}_b - \boldsymbol{w}$, $\mathrm{crs} = (\boldsymbol{A}, \boldsymbol{v}_0, \boldsymbol{v}_1)$, and $\tau = (\boldsymbol{w}, \boldsymbol{s}_0, \boldsymbol{s}_1)$. When the parties query the ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$, Sim returns $(\mathrm{sid}, \mathrm{crs})$ to them and stores τ privately.

$S2$: When R is activated on some subsession $(\text{sid}, \text{ssid})$, Sim runs TrapKeyGen$(\text{crs}, \tau)$ to generate $(\mathbf{pk}, \mathbf{sk}_0, \mathbf{sk}_1)$, where $(\mathbf{pk}, \mathbf{sk}_0, \mathbf{sk}_1) = (\boldsymbol{w}, \boldsymbol{s}_0, \boldsymbol{s}_1)$, and sends $(\text{sid}, \text{ssid}, \mathbf{pk})$ to Adv (or $\text{Adv}_{\text{dummy}}$ when Adv is dummy) as if it were from R, and stores $(\text{sid}, \text{ssid}, \mathbf{pk}, \mathbf{sk}_0, \mathbf{sk}_1)$.

$S3$: When Adv replies with a tuple $(\text{sid}, \text{ssid}, y_0, y_1)$, Sim first looks up the corresponding $(\mathbf{pk}, \mathbf{sk}_0, \mathbf{sk}_1)$, and then computes $m_b \leftarrow \text{Dec}(\mathbf{sk}_b, y_b)$ for each $b \in \{0, 1\}$.

$S4$: Since S has been activated, Sim sends $(\text{sid}, \text{ssid}, \text{sender}, m_0, m_1)$ to $\widetilde{\mathcal{F}}_{\text{OT}}$ as if it were from S.

3.3.3 Remaining corruption cases

When both parties are corrupted, Sim just internally runs a copy of Adv which controls S and R and generates the messages from them both by itself.

When neither party is corrupted, Sim internally runs the honest R on input $(\text{sid}, \text{ssid}, \sigma = 0)$ and honest S on input $(\text{sid}, \text{ssid}, m_0 = 0^\ell, m_1 = 0^\ell)$ by performing the interaction specified by protocol dm^{mode} and delivering all communication messages between internals R and S to Adv.

4 Quantum security analysis

Based on different simulator constructions in different corruption cases, the protocol dm^{mode} is provably secure in the standard UC framework (Peikert *et al.*, 2008). However, its security is considered only within the existence of classical adversaries instead of a quantum one. Due to the fact that the protocol dm^{mode} is built upon LWE hardness, we conjecture that it may be quantum resistant. Therefore, in this section, by using some results from Unruh (2010), we revisit the security proof of dm^{mode} by considering the existence of a quantum adversary and show that dm^{mode} can also preserve its security in the quantum setting. In Section 4.1, we first introduce some results proposed by Unruh (2010). Then we use these results to make quantum analysis for dm^{mode} in Section 4.2.

4.1 Quantum lifting theorem

The quantum analogue of the classical UC framework was proposed by Unruh (2010). Furthermore, an additional result, called the 'quantum lifting theorem (QLT)', is presented.

Theorem 1 (Quantum lifting theorem (statistically), cf. Theorem 15 in Unruh (2010)) Let π and ρ be classical protocols. If π statistically classical-UC-emulates ρ, then π also statistically quantum-UC-emulates ρ.

To the best of our knowledge, this is the first result (cf. Unruh (2010) for its proof) to discuss the classical security in the quantum setting, although it is restricted to the statistical case. One might expect that a computational analogue of such a QLT exists. However, computational QLT (that is, π can computationally quantum-UC-emulate ρ if π computationally classical-UC-emulates ρ) cannot be directly obtained. The reason is that if the hardness on which protocol π relies is not quantum resistant, then such a computational QLT does not exist and cannot be used to lift the classical security of protocols into the quantum setting. However, Unruh (2010) proposed an additional restriction for the computational QLT, i.e., requiring that the classical adversary have the same computational power as a quantum polynomial-time machine in the classical UC framework. This type of adversary is called a 'quantum-strong probabilistic polynomial-time (QPPT)' machine, which is defined as follows:

Definition 5 (Quantum-strong probabilistic polynomial-time machine, cf. Definition 16 in Unruh (2010)) For a classical machine M, if there is a quantum polynomial-time machine $\hat{M}$ such that for any network N, $N \cup \{M\}$ and $N \cup \{\hat{M}\}$ are perfectly indistinguishable, then we say that M is a quantum-strong probabilistic polynomial-time machine, i.e., a QPPT machine. If a classical machine M is a QPPT machine, then we denote it by M'.

Remark 4 Now we recall the machine model as described in Section 2.5 for illustrating Definition 5 (Unruh, 2010). Given a QIM $\hat{M}$, let $\mathcal{C}(\hat{M})$ denote a machine, which behaves like $\hat{M}$, but measures incoming messages in the computational basis before processing them and measures outgoing messages in the computational basis. More precisely, its superoperater $\mathcal{E}^{(n)}_{\mathcal{C}(\hat{M})}$ first invokes $\mathcal{E}_{\text{class}}$ on $\mathcal{H}^{\text{class}} \otimes \mathcal{H}^{\text{quant}}$, then invokes $\hat{M}$'s superoperator $\mathcal{E}^{(n)}_{\hat{M}}$ on $\mathcal{H}^{\text{state}} \otimes \mathcal{H}^{\text{class}} \otimes \mathcal{H}^{\text{quant}}$, and finally invokes $\mathcal{E}_{\text{class}}$ on $\mathcal{H}^{\text{class}} \otimes \mathcal{H}^{\text{quant}}$ again. $\mathcal{C}(\hat{M})$ can be regarded as a special QIM which operates with quantum computation inside but classical communication outside. Since it is possible to simulate QIMs on ITMs with

an exponential overhead, for every machine $\hat{M}$, there exists a classical machine M such that $\mathcal{C}(\hat{M})$ and M are perfectly indistinguishable. Then such a machine M is a QPPT machine, denoted by M'.

Based on Definition 5, the notion of QPPT classical-UC-security can be derived directly, where the adversaries are required to be QPPT machines. Thus, a computational QLT with the restriction on the computation power of adversaries can be achieved.

Definition 6 (QPPT classical-UC-security, cf. Definition 17 in Unruh (2010)) Let π and ρ be classical protocols. We say π QPPT classical-UC-emulates ρ if and only if for every corruption case C and for every QPPT adversary Adv′, there is a QPPT simulator Sim′ such that for every QPPT environment $\mathcal{Z}'$ the networks $\pi^C \cup \{\text{Adv}', \mathcal{Z}'\}$ and $\rho^C \cup \{\text{Sim}', \mathcal{Z}'\}$ are indistinguishable. That is to say, π can achieve QPPT classical-UC-security.

Theorem 2 (Quantum lifting theorem (computationally), cf. Theorem 18 in Unruh (2010)) Let π and ρ be classical protocols. If π QPPT classical-UC-emulates ρ, then π also computationally quantum-UC-emulates ρ.

However, the above computational QLT cannot be directly applied to the classical security proof of dm^{mode}. We first have to prove that dm^{mode} can achieve the QPPT classical-UC-security. Therefore, we carefully revisit its original classical-UC-security proof and show that its security can be preserved when all adversarial machines are QPPT machines. We point out that all QIMs involved in our analysis match the description of the machine model shown in Section 2.5, so do the communication and corruption behaviors between QIMs.

4.2 Quantum security analysis of dm^{mode}

In the classical-UC-security proof of dm^{mode}, the achieved security is described as follows:

1. When in messy mode, the security of S is statistical and the security of R is computational;

2. When in decryption mode, the security of S and R are both computational.

We view the interaction between $\widetilde{\mathcal{F}}_{\text{OT}}$ and two dummy parties (S and R) as an ideal protocol, denoted by $\rho^{\widetilde{\mathcal{F}}_{\text{OT}}}$, which securely realizes a multi-session OT. If dm^{mode} classical-UC-emulates $\widetilde{\mathcal{F}}_{\text{OT}}$, it implies that dm^{mode} classical-UC-emulates the ideal protocol $\rho^{\widetilde{\mathcal{F}}_{\text{OT}}}$.

If the protocol dm^{mode} operates in a specific corruption case, e.g., when only R is corrupted, then we denote it with $\text{dm}_R^{\text{mode}}$. When only R is corrupted, the interaction between environment $\mathcal{Z}$ and the real world will be denoted by $\text{dm}_R^{\text{mode}} \cup \{\mathcal{Z}, \text{Adv}\}$. Analogously, the interaction between environment $\mathcal{Z}$ and the ideal world will be denoted by $\rho_R^{\widetilde{\mathcal{F}}_{\text{OT}}} \cup \{\mathcal{Z}, \text{Sim}\}$. In addition, if $\mathcal{Z}$ cannot distinguish with which world it is interacting, then we say these two networks are indistinguishable, i.e., $\text{dm}_R^{\text{mode}} \cup \{\text{Adv}, \mathcal{Z}\} \approx \rho_R^{\widetilde{\mathcal{F}}_{\text{OT}}} \cup \{\text{Sim}, \mathcal{Z}\}$.

Now we give a new security analysis for dm^{mode} by restricting the adversary, simulator, and environment as polynomial-time QIMs, denoted by $\widehat{\text{Adv}}$, $\widehat{\text{Sim}}$, and $\hat{\mathcal{Z}}$, respectively. In the quantum setting, we still consider the security of dm^{mode} under static corruptions, where the corruption decision is made before the execution of the protocol and cannot be changed during the execution of the protocol. However, if the classical-UC-security of dm^{mode} is achieved under adaptive corruptions (i.e., the corruption is probabilistically decided during the protocol execution), we cannot separate different corruption cases in a case distinction when we consider its security within the existence of a quantum adversary. This is caused by the fact that it is not clear at the beginning of the protocol in which corruption case it will be. However, thanks to the UC-security of dm^{mode} achieved under static corruptions, we can analyze its quantum security in a case distinction by using different QLTs as presented before. With this strategy, we first show our analysis of dm^{mode} in messy mode, and then step into the case in decryption mode.

4.2.1 In messy mode

In this part, we first analyze dm^{mode}'s quantum security in messy mode and under static corruptions: (1) when only the receiver is corrupted; (2) when only the sender is corrupted; (3) when both parties are corrupted; (4) when neither party is corrupted.

1. When only R is corrupted

In messy mode, the classical security of the sender is statistical if only R is corrupted, i.e., $\text{dm}_R^{\text{mes}} \cup \{\text{Adv}, \mathcal{Z}\} \overset{\text{s}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\text{OT}}} \cup \{\text{Sim}, \mathcal{Z}\}$, where '$\overset{\text{s}}{\approx}$' denotes that both networks are statistically indistinguishable.

Since Adv is a static adversary who decides its

corruption with R before the execution of $\mathrm{dm}_R^{\mathrm{mes}}$, we can achieve Lemma 1 by using Theorem 1. Here, we require $\widehat{\mathrm{Adv}}$, $\widehat{\mathrm{Sim}}$, and $\hat{\mathcal{Z}}$ to be quantum polynomial-time machines by following Unruh (2010).

Lemma 1 If $\mathrm{dm}_R^{\mathrm{mes}} \cup \{\mathcal{Z}, \mathrm{Adv}\} \overset{\mathrm{s}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}\}$, and assuming that the adversary, simulator, and environment are quantum polynomial-time machines, denoted by $\widehat{\mathrm{Adv}}$, $\widehat{\mathrm{Sim}}$, and $\hat{\mathcal{Z}}$, respectively, we have $\mathrm{dm}_R^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\} \overset{\mathrm{s}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}\}$.

Remark 5 When only R is corrupted in messy mode, the output distributions of the real world and the ideal world are statistically close. Actually, an unbounded M and an unbounded $\hat{M}$ can be regarded as having the same probability to distinguish any distribution. In particular, if two distributions are statistically close, then neither a polynomial-time or unbounded M, nor polynomial-time or unbounded $\hat{M}$ can distinguish them.

2. When only S is corrupted

In messy mode, the classical security of the receiver is computational when only S is corrupted, i.e., $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathrm{Adv}, \mathcal{Z}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathrm{Sim}, \mathcal{Z}\}$, where '$\overset{\mathrm{c}}{\approx}$' denotes that both networks are computationally indistinguishable. As illustrated before, we cannot directly use Theorem 2 to lift this computational security into the quantum setting. Thus, we revisit its classical security proof to check whether its security satisfies Definition 6. If so, we obtain Lemma 2:

Lemma 2 If $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{Z}, \mathrm{Adv}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}\}$, and assuming that the adversary, simulator, and environment are quantum polynomial-time machines, denoted by $\widehat{\mathrm{Adv}}$, $\widehat{\mathrm{Sim}}$, and $\hat{\mathcal{Z}}$, respectively, we have $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}\}$.

Proof If Adv corrupts only S in messy mode, then $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathrm{Adv}, \mathcal{Z}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathrm{Sim}, \mathcal{Z}\}$ in the classical setting. Now we consider the security of $\mathrm{dm}^{\mathrm{mes}}$ in the same corruption case except for substituting the adversary, simulator, and environment with QIMs.

In the execution of the real world, each party will query the ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$ for crs, where $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$ runs SetupMessy(1^n) to generate crs. All interactions will be executed in the following two phases:

Phase 1 In this phase, the environment machine chooses σ for honest R. Then R runs $(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathrm{KeyGen}(\mathrm{crs}, \sigma)$ and sends $\mathbf{pk}$ to $\mathcal{Z}$.

Now we assume that the environment and adversary are both quantum polynomial-time machines, i.e., $\hat{\mathcal{Z}}$ and $\widehat{\mathrm{Adv}}$. Since $\mathrm{dm}_S^{\mathrm{mes}}$ is classical, all messages sent from $\hat{\mathcal{Z}}$ to $\mathrm{dm}_S^{\mathrm{mes}}$ should be measured before being used. Then honest R will return message $\mathbf{pk}$ to environment $\hat{\mathcal{Z}}$. Here, $\mathbf{pk}$ can also be regarded as a communication message sent from R and then forwarded by adversary $\widehat{\mathrm{Adv}}_{\mathrm{dummy}}$ to environment $\hat{\mathcal{Z}}$, where adversary $\widehat{\mathrm{Adv}}$ is dummy to be controlled by the environment. Since $\mathbf{pk}$ is generated as an LWE instance, for quantum PPT machine $\hat{\mathcal{Z}}$ and classical PPT machine $\mathcal{Z}$ (even equipped with polynomial-time quantum computation power), they both have negligible probability to break LWE hardness. Thus, in this phase, the view of a quantum polynomial-time machine $\hat{\mathcal{Z}}$ and the view of a PPT classical machine $\mathcal{Z}$ are perfectly indistinguishable. Here $\hat{\mathcal{Z}}$ behaves like $\mathcal{C}(\hat{\mathcal{Z}})$ described in Remark 4, and then machine $\mathcal{Z}$ is a QPPT machine $\mathcal{Z}'$.

Phase 2 In this phase the environment machine will produce (y_0, y_1) using $\mathbf{pk}$ and then send it to R. The honest R will output $m_\sigma \leftarrow \mathrm{Dec}(sk, y_\sigma)$.

Actually (y_0, y_1) is a communication message sent from S to R, and it can be regarded as a message sent from $\widehat{\mathrm{Adv}}$ or forwarded by $\widehat{\mathrm{Adv}}_{\mathrm{dummy}}$ which is controlled by $\hat{\mathcal{Z}}$. Since $\widehat{\mathrm{Adv}}$ classically communicates with R in the name of S, it would additionally measure the forwarded messages in the computational basis first. Thus, the view of $\hat{\mathcal{Z}}$ will not be modified; i.e., $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\}$ and $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \mathrm{Adv}\}$ are perfectly indistinguishable, where $\widehat{\mathrm{Adv}}$ behaves like $\mathcal{C}(\widehat{\mathrm{Adv}})$. Then, Adv is a QPPT machine Adv′ as defined by Definition 5. Furthermore, both Adv′ and $\mathrm{dm}_S^{\mathrm{mes}}$ measure all messages upon sending and receiving, and then $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \mathrm{Adv}'\}$ and $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{C}(\hat{\mathcal{Z}}), \mathrm{Adv}'\}$ are perfectly indistinguishable, where $\mathcal{C}(\hat{\mathcal{Z}})$ is perfectly indistinguishable from a QPPT machine $\mathcal{Z}'$; thus, $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{C}(\hat{\mathcal{Z}}), \mathrm{Adv}'\}$ and $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{Z}', \mathrm{Adv}'\}$ are perfectly indistinguishable.

Based on the above two-phase execution of the real world, the views of $\hat{\mathcal{Z}}$ and $\mathcal{Z}$ are perfectly indistinguishable. Thus, $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{Z}, \mathrm{Adv}\}$ and $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{Z}', \mathrm{Adv}'\}$ are perfectly indistinguishable when two QPPT machines Adv′ and $\mathcal{Z}'$ participate in the execution of the real world.

In the execution of the ideal world, we revisit the construction of the simulator in four steps. For clarity, we define the simulator as $\mathrm{Sim}(S1, S2, S3, S4)$ and a quantum polynomial-time simulator as $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)$.

$S1$: First the simulator runs $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupDec}(1^n)$ and sends only crs to the environment machine. In this step the simulator simulates the classical ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$ which allows only classical state access and returns crs as output.

Now we assume that the simulator has polynomial-time quantum computation power in $S1$, denoted as $\widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)$. After running $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupDec}(1^n)$, $\widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)$ will measure the output before sending it to $\mathcal{Z}$ as if the output came from $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$. Thus, the view of $\mathcal{Z}$ would not be modified, and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}(S1, S2, S3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ are perfectly indistinguishable.

The simulation in this step will lead to the difference between the distributions of simulated crs and the real one that comes from $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{mes}}$. Since distinguishing these two crs' distributions amounts to solving a decisional LWE problem, classical polynomial-time $\mathcal{Z}$ and quantum polynomial-time $\hat{\mathcal{Z}}$ have negligible probability to distinguish these two distributions. Then $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ are perfectly indistinguishable; i.e., $\mathcal{Z}$ is a QPPT machine $\mathcal{Z}'$ running in this network.

$S2$: In this step, the simulator will run $(\mathbf{pk}, \mathbf{sk}_0, \mathbf{sk}_1) \leftarrow \mathrm{TrapKeyGen}(\tau)$ and send $\mathbf{pk}$ to the environment. Here we assume that the simulator has polynomial time quantum computation power in $S2$, denoted by $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)$. Since $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)$ simulates the behavior of R, it will communicate classically outside but perform quantum computation inside. The generation of $\mathbf{pk}$ comes from the trapdoor information that is related to crs whose generation is based on LWE hardness. Thus, for both classical PPT $\mathcal{Z}$ and quantum polynomial-time $\hat{\mathcal{Z}}$, $\widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)$ and $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)$ are perfectly indistinguishable.

Summarizing the above, $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)$ is perfectly indistinguishable with classical simulator $\mathrm{Sim}(S1, S2, S3, S4)$. It means that for a quantum polynomial-time $\hat{\mathcal{Z}}$, there exists a classical PPT $\mathcal{Z}$ which has a perfectly indistinguishable view with that of $\hat{\mathcal{Z}}$, such that $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}(S1, S2, S3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)\}$ are perfectly indistinguishable.

$S3$: In this step, the environment machine will produce (y_0, y_1) for R. As a communication message, (y_0, y_1) will be received by the simulator which plays the role of R to interact with the environment. Message (y_0, y_1) can be generated by invoking the real adversary as a black box, where the simulator works as the shell of the real adversary. As soon as the simulator receives (y_0, y_1) from the output tape of the adversary, it will compute $m_b \leftarrow \mathrm{Dec}(sk_b, y_b)$ for each $b \in \{0, 1\}$.

Based on the analysis of Phase 2 in the real world, we substitute Adv by a polynomial-time QIM $\widehat{\mathrm{Adv}}$ in the simulator's construction, i.e., $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)$. All messages sent by $\widehat{\mathrm{Adv}}$ will be measured before being used, and then $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)\}$ are perfectly indistinguishable.

$S4$: Since R has been activated, the simulator has to send (m_0, m_1) to the ideal functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ as if it were from S. In this step, as there is no interaction between the simulator and the environment machine, then $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ are perfectly indistinguishable.

Summarizing the above analysis in the ideal world, we know that $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}(S1, S2, S3, S4)\}$ and $\rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ are perfectly indistinguishable. By the definition of QPPT machines, $\mathcal{Z}$ and $\mathrm{Sim}(S1, S2, S3, S4)$ are both QPPT machines in the ideal world, i.e., $\mathcal{Z}'$ and Sim'.

Based on the above analyses in both worlds, we have $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}\}$. By the definition of QPPT machines, we also have $\mathrm{dm}_S^{\mathrm{mes}} \cup \{\mathcal{Z}', \mathrm{Adv}'\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}', \mathrm{Sim}'\}$. Lemma 2 has been proven.

3. When both parties are corrupted

In the case where both the sender and the receiver are corrupted, the simulator can internally run a copy of the adversary to perfectly simulate a protocol execution by generating the real messages exchanged between two parties. In this corruption case, the security in the classical setting can be trivially preserved in the quantum setting since the quantum adversary cannot distinguish between two perfectly indistinguishable worlds.

4. When neither party is corrupted

In the case where neither party is corrupted, the simulator internally runs the honest R and S on inputs $(\mathrm{sid}, \mathrm{ssid}, \sigma = 0)$ and $(\mathrm{sid}, \mathrm{ssid}, m_0 = 0^\ell, m_1 =$

0^ℓ). When some dummy party is activated in the ideal execution, the simulator will activate the appropriate algorithm for the corresponding dummy party, and deliver all communications between its internal S and R to the adversary Adv. Peikert *et al.* (2008) showed the computational security in messy mode when neither party is corrupted and its proof is sketched as follows:

Let $\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(m_0,m_1,b)$ denote $\mathcal{Z}$'s output in protocol $\mathrm{dm}^{\mathrm{mes}}$ where it sets the inputs of S and R as (m_0,m_1) and b. By the messy property of dual-mode encryption, we have

$$\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(m_0,m_1,1) \overset{\mathrm{s}}{\approx} \mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,m_1,1)$$

and

$$\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,m_1,0) \overset{\mathrm{s}}{\approx} \mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,0^\ell,0).$$

Due to the computational security of R in messy mode, we also have

$$\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,m_1,1) \overset{\mathrm{c}}{\approx} \mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,m_1,0).$$

Here $\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(m_0,m_1,1)$ is an actual real-world execution, and $\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(0^\ell,0^\ell,0)$ is the execution of the simulator. In addition, thanks to the completeness of the dual-mode cryptosystem, $\mathrm{EXEC}_{\mathrm{dm}^{\mathrm{mes}},\mathrm{Adv},\mathcal{Z}}(m_0,m_1,1)$ is statistically indistinguishable from the ideal-world execution on inputs (m_0,m_1,b). Thus, the computational security is achieved in this corruption case.

Note that the security in the case where neither party is corrupted is deduced by the security in the cases where only S is corrupted and where only R is corrupted. Thus, each indistinguishability derived from each two adjacent protocol executions above can be lifted into the quantum setting, thanks to the quantum analysis in the cases where only S is corrupted and where only R is corrupted. Therefore, the computational security of $\mathrm{dm}^{\mathrm{mes}}$ when neither party is corrupted is preserved in the quantum setting.

4.2.2 In decryption mode

In this part, we analyze $\mathrm{dm}^{\mathrm{mode}}$'s quantum security in decryption mode under four static corruptions. However, since the security of $\mathrm{dm}^{\mathrm{dec}}$ in the cases where both parties are corrupted and neither party is corrupted is similar to that we have shown in messy mode, here we analyze only $\mathrm{dm}^{\mathrm{dec}}$'s quantum security in the cases where only the receiver is corrupted and where only the sender is corrupted.

1. When only R is corrupted

In decryption mode, the security of the sender is computational if only R is corrupted, i.e., $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z},\mathrm{Adv}\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z},\mathrm{Sim}\}$. Now under this corruption, we revisit the original security proof of $\mathrm{dm}_R^{\mathrm{dec}}$ considering the existence of quantum adversaries. Then we can obtain Lemma 3:

Lemma 3 If $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z},\mathrm{Adv}\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z},\mathrm{Sim}\}$, and when we assume that the adversary, simulator, and environment are quantum polynomial-time machines, denoted by $\widehat{\mathrm{Adv}}$, $\widehat{\mathrm{Sim}}$, and $\hat{\mathcal{Z}}$, respectively, then $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}},\widehat{\mathrm{Adv}}\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}},\widehat{\mathrm{Sim}}\}$.

Proof If Adv corrupts only with R in decryption mode, then $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z},\mathrm{Adv}\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z},\mathrm{Sim}\}$ in the classical setting.

In the execution of the real world, all parties first query ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$ for crs, where $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$ runs $\mathrm{SetupDec}(1^n)$ to generate crs. All interactions will proceed as follows: the environment machine first chooses an arbitrary $\mathbf{pk}$ and inputs (m_0,m_1) for the honest S, and then S sends $y_b \leftarrow \mathrm{Enc}(\mathbf{pk},b,m_b)$ for each $b\in\{0,1\}$ to the environment machine.

Now we assume that the environment and the adversary are both quantum polynomial-time machines, i.e., $\hat{\mathcal{Z}}$ and $\widehat{\mathrm{Adv}}$. Since $\mathrm{dm}_R^{\mathrm{dec}}$ is classical, all messages sent from $\hat{\mathcal{Z}}$ to $\mathrm{dm}_R^{\mathrm{dec}}$ should be measured before being used. As a communication message sent from R to S, $\mathbf{pk}$ can be viewed as a message sent from $\widehat{\mathrm{Adv}}$ or forwarded by $\widehat{\mathrm{Adv}}_{\mathrm{dummy}}$ which is controlled by $\hat{\mathcal{Z}}$. Since $\widehat{\mathrm{Adv}}$ classically communicates with S in the name of R, it would additionally measure the forwarded messages in the computational basis first. Thus, the view of $\hat{\mathcal{Z}}$ would not be modified; i.e., $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}},\widehat{\mathrm{Adv}}\}$ and $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}},\mathrm{Adv}\}$ are perfectly indistinguishable, where $\widehat{\mathrm{Adv}}$ behaves like $\mathcal{C}(\widehat{\mathrm{Adv}})$. Then Adv can be regarded as a QPPT machine Adv′. Furthermore, both Adv′ and $\mathrm{dm}_R^{\mathrm{dec}}$ measure all messages upon sending and receiving. Then $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}},\mathrm{Adv}'\}$ and $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{C}(\hat{\mathcal{Z}}),\mathrm{Adv}'\}$ are perfectly indistinguishable, where $\mathcal{C}(\hat{\mathcal{Z}})$ is perfectly indistinguishable from a QPPT machine $\mathcal{Z}'$. Thus, $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{C}(\hat{\mathcal{Z}}),\mathrm{Adv}'\}$ and $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z}',\mathrm{Adv}'\}$ are perfectly indistinguishable.

Since messages (y_0, y_1) sent from $\mathrm{dm}_R^{\mathrm{dec}}$ to $\hat{\mathcal{Z}}$ are ciphertexts of (m_0, m_1) which are generated using the LWE-based encryption scheme described in Section 3, quantum PPT machine $\hat{\mathcal{Z}}$ and classical PPT $\mathcal{Z}$ (even equipped with polynomial-time quantum computation power) have negligible probabilities to break the LWE hardness. Thus, the view of a quantum polynomial-time machine $\hat{\mathcal{Z}}$ and the view of a PPT classical machine $\mathcal{Z}$ are perfectly indistinguishable. By the definition of QPPT machine, we know that $\mathcal{Z}$ is a QPPT machine $\mathcal{Z}'$. Then $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z}, \mathrm{Adv}\}$ and $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z}', \mathrm{Adv}'\}$ are perfectly indistinguishable when two QPPT machines Adv' and $\mathcal{Z}'$ participate in the execution of the real world.

In the execution of the ideal world, we revisit the construction of the simulator in four steps. For clarity, we denote the classical simulator as $\mathrm{Sim}(S1, S2, S3, S4)$ and a quantum polynomial-time simulator as $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)$.

$S1$: The simulator runs $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupMessy}(1^n)$ and outputs only crs when the query for crs is asked. In this step, the simulator simulates the classical ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$ which allows only classical state access and returns crs as the output.

Now we assume that the simulator has polynomial-time quantum computation power in $S1$, denoted by $\widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)$. After running $(\mathrm{crs}, \tau) \leftarrow \mathrm{SetupMessy}(1^n)$, $\widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)$ will measure the output before sending it to $\mathcal{Z}$ as if it were from $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$. Thus, the view of $\mathcal{Z}$ would not be modified; i.e., $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}(S1, S2, S3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ are perfectly indistinguishable.

The simulation in this step will lead to the difference in the distribution from the simulated crs and the real one generated by $\mathcal{F}_{\mathrm{CRS}}^{\mathrm{dec}}$. Since distinguishing these two crs distributions amounts to solving a decisional LWE problem, classical PPT $\mathcal{Z}$ and quantum polynomial-time $\hat{\mathcal{Z}}$ have negligible probabilities to distinguish these two distributions. Then we know that $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ are perfectly indistinguishable; i.e., $\mathcal{Z}$ is a QPPT machine $\mathcal{Z}'$ running in this network by Definition 5.

$S2$: In this step, the environment machine will produce $\mathbf{pk}$ for S. As a communication message, $\mathbf{pk}$ will be received by the simulator which plays the role of S. Message $\mathbf{pk}$ can be generated by invoking the real adversary as a black box, where the simulator works as the shell of the real adversary. Now we substitute the classical PPT machine Adv by a quantum polynomial-time machine $\widehat{\mathrm{Adv}}$ in the simulator's construction, i.e., $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)$. Since all messages sent by $\widehat{\mathrm{Adv}}$ will be measured before being used, based on the analysis in the real world, $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, S2, S3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)\}$ are perfectly indistinguishable.

$S3$: In the third step, the simulator will run $b \leftarrow \mathrm{FindMessy}(\tau, \mathbf{pk})$ to find messy branch b with an overwhelming probability. Then the simulator queries ideal functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ with the chosen bit $1-b$ to obtain m_{1-b} in the name of R. Here we assume that as this simulator has polynomial-time quantum power in $S3$, it will communicate classically outside but perform quantum computation inside. Since FindMessy is a classical PPT algorithm, it is still efficient when a quantum PPT machine runs on it. Thus, $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, S3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)\}$ are perfectly indistinguishable.

$S4$: In this step the simulator will compute $(y_{1-b} = \mathrm{Enc}(\mathbf{pk}, 1-b, m_{1-b}), y_b = \mathrm{Enc}(\mathbf{pk}, b, 0^\ell))$ and send it to the adversary as if it were from S. We assume that the simulator has polynomial-time quantum power in $S4$, i.e., $\widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)$, and it simulates the behavior of honest S and sends (y_0, y_1) to the environment. Thus, $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ are perfectly indistinguishable.

Since simulated $y_b = \mathrm{LWEEnc}(\mathbf{pk}, b, 0^\ell)$ is statistically close to real $y_b = \mathrm{LWEEnc}(\mathbf{pk}, b, m_b)$, classical PPT $\mathcal{Z}$ and quantum polynomial-time $\hat{\mathcal{Z}}$ have negligible probabilities to distinguish these two ciphertexts. Then $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ are perfectly indistinguishable; i.e., $\mathcal{Z}$ is a QPPT machine $\mathcal{Z}'$ in this network.

Summarizing the above analysis in the ideal world, we have that $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}(S1, S2, S3, S4)\}$ and $\rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}(\hat{S}1, \hat{S}2, \hat{S}3, \hat{S}4)\}$ are perfectly indistinguishable. By the definition of QPPT machines, $\mathcal{Z}$ and $\mathrm{Sim}(S1, S2, S3, S4)$ are both QPPT

machines in the execution of the ideal world, i.e., $\mathcal{Z}'$ and Sim′.

Based on the above analysis in two worlds, we have $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}\}$. By Definition 5, we have $\mathrm{dm}_R^{\mathrm{dec}} \cup \{\mathcal{Z}', \mathrm{Adv}'\} \overset{\mathrm{c}}{\approx} \rho_R^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}', \mathrm{Sim}'\}$. Lemma 3 has been proven.

2. When only S is corrupted

In this corruption case, because the construction of the simulator is the same as that in messy mode, the quantum analysis is similar to the case when only the sender is corrupted in messy mode. The only difference between the real world and the ideal world lies in the generation of the public and secret keys. However, the trapdoor key generation $(\mathbf{pk}, \mathbf{sk}_\sigma)$ is computationally indistinguishable with $\mathrm{KeyGen}(\mathrm{crs}, \sigma)$ for any crs generated by SetupDec. By assuming the LWE hardness, it provides only computational security for the receiver in decryption mode (cf. Lemma 7.7 in Peikert *et al.* (2008)). The quantum security follows similarly. We can directly obtain Lemma 4:

Lemma 4 If $\mathrm{dm}_S^{\mathrm{dec}} \cup \{\mathcal{Z}, \mathrm{Adv}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\mathcal{Z}, \mathrm{Sim}\}$, assuming that the adversary, simulator, and environment are quantum polynomial-time machines, denoted by $\widehat{\mathrm{Adv}}$, $\widehat{\mathrm{Sim}}$, and $\hat{\mathcal{Z}}$, respectively, then we have $\mathrm{dm}_S^{\mathrm{dec}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Adv}}\} \overset{\mathrm{c}}{\approx} \rho_S^{\widetilde{\mathcal{F}}_{\mathrm{OT}}} \cup \{\hat{\mathcal{Z}}, \widehat{\mathrm{Sim}}\}$.

3. Remaining corruption cases

The security in the cases where both parties are corrupted and where neither party is corrupted follows symmetrically as shown in messy mode.

According to the above security analysis of $\mathrm{dm}^{\mathrm{mode}}$ considering the existence of polynomial-time quantum adversaries, we can claim that the security of $\mathrm{dm}^{\mathrm{mode}}$ can be lifted into a quantum world by Proposition 2:

Proposition 2 UC-secure OT protocol $\mathrm{dm}^{\mathrm{mode}}$ can preserve its achieved security in the quantum setting within the existence of polynomial-time quantum adversaries.

5 Simple hybrid arguments framework

In this section, we first introduce a framework proposed by Hallgren *et al.* (2011; 2015), called 'simple hybrid arguments (SHA)', which is developed to capture a large family of classical security in the UC model that can go through into the quantum setting as long as their underlying primitives are quantum resistant. As a supplement to our quantum security analysis shown in Section 4, we apply this framework to the OT protocol $\mathrm{dm}^{\mathrm{mode}}$ and show that our quantum analysis of $\mathrm{dm}^{\mathrm{mode}}$ is compatible with the SHA framework.

5.1 Simple hybrid arguments

To introduce the notion of SHA, some relevant notions are presented first. Here we first recall their description for the QIM, which respects the machine model proposed by Unruh (2010).

A QIM $\hat{M}$ is an ensemble of interactive circuits $\{\hat{M}_n\}_{n\in\mathbb{N}}$, where n is the security parameter. For some n, $\hat{M}_n$ consists of a sequence of circuits $\{\hat{M}_n^{(i)}\}_{i=1,2,\ldots,r(n)}$, where $\{\hat{M}_n^{(i)}\}$ defines the operations of $\hat{M}$ in the ith round and $r(n)$ is the round number for which $\hat{M}_n$ operates. $\hat{M}$ works on three registers, i.e., a state register S used for input and workspace, an output register O, and a network register N for interacting with other machines. The size or the running time of $\hat{M}$ is denoted by $t(n)$. We say a machine $\hat{M}$ is polynomial time if there exists a deterministic classical Turing machine that computes the description of $\hat{M}_n$ in $t(n) = \mathrm{poly}(n)$ on input 1^n. A non-interactive quantum machine, denoted by QTM, is a QIM without network register; i.e., its machine circuit is only $\{\hat{M}_n^{(i)}\}_{i=1}$. Note that a classical ITM can be viewed as a special QIM, where it stores only classical string and all circuits there are classical. Then we recall the definition of simply related machines as follows:

Definition 7 (Simply related machines, cf. Definition 4.2 in Hallgren *et al.* (2015)) Let $\hat{M}_a$ and $\hat{M}_b$ denote two QIMs. If there exists a $t(n)$-time QTM $\hat{M}$ and a pair of classical distributions (D_a, D_b) such that (1) $\hat{M}(D_a) \equiv \hat{M}_a$ (for two QIMs $\hat{M}_1$ and $\hat{M}_2$, $\hat{M}_1 \equiv \hat{M}_2$ implies that two machines can be described by the same circuits and behave identically on all inputs), (2) $\hat{M}(D_b) \equiv \hat{M}_b$, and (3) $D_a \approx_{\mathrm{qc}}^{2t,\varepsilon} D_b$ ($\varepsilon(n)$ denotes a negligible advantage for distinguishing D_a and D_b), we say that $\hat{M}_a$ and $\hat{M}_b$ are (t, ε)-simply related.

Note that here $\hat{M}(D_a)$ represents a machine $\hat{M}$ running on the inputs sampled from D_a. In addition, $D_a \approx_{\mathrm{qc}}^{2t,\varepsilon} D_b$ means that for any $2t(n)$-time quantum distinguisher, it can distinguish only these two distributions with a negligible advantage $\varepsilon(n)$. The condition $D_a \approx_{\mathrm{qc}}^{2t,\varepsilon} D_b$ is required for the derivation of Lemma 5 whose proof comes from the contradiction that if a $t(n)$-time distinguisher $\hat{\mathcal{Z}}$ that

can distinguish $\hat{M}_a$ and $\hat{M}_b$ exists, then a $2t(n)$-time distinguisher $\hat{\mathcal{D}}$ for distinguishing D_a and D_b can be constructed. Before we introduce Lemma 5, we first recall the notion of (t,ε)-interactively indistinguishable QIMs.

Definition 8 ((t,ε)-interactively indistinguishable QIMs, cf. Definition 2.5 in Hallgren *et al.* (2015)) Let $\hat{M}_a$ and $\hat{M}_b$ denote two QIMs. We say $\hat{M}_a$ and $\hat{M}_b$ are (t,ε)-interactively indistinguishable, denoted by $\hat{M}_a \approx_{\mathrm{i}}^{t,\varepsilon} \hat{M}_b$, if for any $t(n)$-time QIM $\hat{\mathcal{Z}}$ and any mixed state φ_n on $t(n)$ qubits,

$$|\Pr[\langle\hat{\mathcal{Z}}(\varphi_n),\hat{M}_a\rangle=1]-\Pr[\langle\hat{\mathcal{Z}}(\varphi_n),\hat{M}_b\rangle=1]|\leq\varepsilon(n),$$

where $\varepsilon(n)$ is negligible.

If for every $t(n)\leq \mathrm{poly}(n)$, there exists a negligible $\varepsilon(n)$ such that $\hat{M}_a$ and $\hat{M}_b$ are (t,ε)-interactively indistinguishable, $\hat{M}_a$ and $\hat{M}_b$ are called 'quantum computationally interactively indistinguishable', denoted by $\hat{M}_a \approx_{\mathrm{qci}} \hat{M}_b$. Likewise, for statistically interactively indistinguishable QIMs, this type of indistinguishability is denoted by $\hat{M}_a \approx_{\mathrm{qsi}} \hat{M}_b$, if an unbounded interactive quantum distinguisher $\hat{\mathcal{Z}}$ exists.

Based on Definitions 7 and 8, Lemma 5 can be achieved:

Lemma 5 (cf. Lemma 4.3 in Hallgren *et al.* (2015)) If two QIMs $\hat{M}_a$ and $\hat{M}_b$ are (t,ε)-simply related, then $\hat{M}_a \approx_{\mathrm{qci}}^{t,\varepsilon} \hat{M}_b$.

Now we can introduce the SHA framework, which uses the above notions.

Definition 9 (Simple hybrid arguments, cf. Definition 4.4 in Hallgren *et al.* (2015)) If there is a sequence of intermediate machines $\hat{M}_1,\hat{M}_2,\ldots,\hat{M}_{\ell-1}$ between $\hat{M}_0$ and $\hat{M}_\ell$ such that each two adjacent machines $\hat{M}_{i-1}$ and $\hat{M}_i$ $(i=1,2,\ldots,\ell)$ are $(t,\varepsilon/\ell)$-simply related, we say $\hat{M}_0$ and $\hat{M}_\ell$ are related by a (t,ε)-SHA of length ℓ.

Furthermore, we can obtain Lemma 6 (cf. Lemma 4.5 in Hallgren *et al.* (2015)):

Lemma 6 (cf. Lemma 4.5 in Hallgren *et al.* (2015)) For any t,ε, and ℓ, if $\hat{M}_0$ and $\hat{M}_\ell$ are related by a (t,ε)-SHA of length ℓ, then $\hat{M}_0$ and $\hat{M}_\ell$ are (t,ε)-interactively indistinguishable.

Lemma 6 is deduced by showing that if two QIMs can fall into the SHA framework, then these two QIMs are (t,ε)-interactively indistinguishable by Lemma 5; i.e., no $t(n)$-time distinguisher $\hat{\mathcal{Z}}$ can distinguish these two QIMs with overwhelming advantages.

5.2 Application of the SHA framework

Now we apply the SHA framework to protocol $\mathrm{dm}^{\mathrm{mode}}$. We first regard the execution of $\mathrm{dm}^{\mathrm{mode}}$ with Adv as a composed machine, denoted by $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$. The execution of $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ with two dummy parties in the ideal world can be regarded as an ideal protocol $\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}}$ which captures the security requirements of secure oblivious transfer. Likewise, the execution of $\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}}$ with simulator Sim is represented by a composed machine $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$. Using the SHA framework in Definition 9, we show that $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ and $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$ are still interactively indistinguishable after all adversarial ITMs; i.e., the adversary, simulator, and environment are changed into QIMs.

Now we recall the classical UC-security proof of $\mathrm{dm}^{\mathrm{mode}}$. The real world and the ideal world can be represented by $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ and $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$, respectively. These two composed classical ITMs can be regarded as two special QIMs, where the registers store only classical states and all circuits are classical. In Section 4, we show the quantum security of $\mathrm{dm}^{\mathrm{mode}}$ by modifying the adversarial machines Adv, Sim, and $\mathcal{Z}$ step by step in each corruption case. During the execution of the real world (or the ideal world), we modify the composed machine from $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ to $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ (or from $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$ to $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$), where these modifications will lead to a sequence of intermediate machines from $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ to $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ (or from $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$ to $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$).

Each two adjacent machines in the machine modification sequence from $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ to $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ are perfectly indistinguishable. It implies that these two adjacent machines behave identically on all inputs chosen from the same distribution. They can be viewed as a special pair of simply related machines with a negligible impact on (t,ε). Thus, $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ and $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ are related by a (t,ε)-SHA of length ℓ_{real} (the number of machine modifications in the real world), i.e., (t,ε)-interactively indistinguishable. Likewise, each two adjacent machines between $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$ and $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$ are simply related. Thus, $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$ and $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$ are related by a (t,ε)-SHA of length ℓ_{ideal} (the number of machine modifications in the real world). In addition, since the classical UC-security of $\mathrm{dm}^{\mathrm{mode}}$ shows that $M_{\mathrm{dm}^{\mathrm{mode}},\mathrm{Adv}}$ and $M_{\rho^{\widetilde{\mathcal{F}}_{\mathrm{OT}}},\mathrm{Sim}}$ are interactively indistinguishable, we finally have that $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}}$ and

$M_{\rho^{\tilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$ are interactively indistinguishable, i.e., $M_{\mathrm{dm}^{\mathrm{mode}},\widehat{\mathrm{Adv}}} \approx^{t,\varepsilon}_{\mathrm{qci}} M_{\rho^{\tilde{\mathcal{F}}_{\mathrm{OT}}},\widehat{\mathrm{Sim}}}$.

6 Game-preserving reduction

In this section, we introduce another framework, called 'game-preserving reduction' (Song, 2014), which is used to check whether the classical security proof of post-quantum cryptographic primitives can be preserved in the quantum setting. In other words, this framework explores whether the reductions derived from the classical security proof of some post-quantum cryptographic primitives are still available in the quantum setting from the perspective of the provable security paradigm. If so, we say these reductions are quantum-friendly, and this post-quantum cryptographic primitive is quantum resistant.

Intuitively, a reduction is a transformation from an adversary to another adversary. The notion of game-preserving reduction is used to depict the type of reductions that can still make sense when adversaries are assumed to have quantum power. In addition, a notion, called 'class-respectful reduction', is proposed as an instance to illustrate the game-preserving reductions.

In this section, we show that the classical reductions inherently included in the classical UC-security proof of $\mathrm{dm}^{\mathrm{mode}}$ satisfy the sufficient conditions of class-respectful reduction. Therefore, these reductions are quantum-friendly and this application is compatible with our analysis in Section 4. We first recall the notion of 'class-respectful reduction' (Song, 2014) in Section 6.1 and apply it to $\mathrm{dm}^{\mathrm{mode}}$ in Section 6.2.

6.1 Game-preserving reduction: class-respectful reduction

For a clear overview of class-respectful reduction, we introduce a formal definition of 'reduction', which is built upon the notion of 'game'. Then we list some expected properties that an effective reduction may have and show the procedure on which a reduction can be built. These notions (Song, 2014) are used for the check on whether an effective reduction already at hand respects the properties of a specified class of reductions. If it does, we call this reduction a 'class-respectful reduction' for that specified class. Finally, we recall the formal definition of class-respectful reduction which is specified as two sufficient conditions and the main tool for checking the security of $\mathrm{dm}^{\mathrm{mode}}$ in the quantum setting (Section 6.2).

A game G denotes a probabilistic process between two players, i.e., a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. After several rounds of interaction between them, $\mathcal{C}$ will output one bit to indicate if $\mathcal{A}$ wins game G. We denote the success probability of $\mathcal{A}$ in G by $\omega_G(\mathcal{A})$. In addition, if there exist two machines M and N such that $\omega_G(M) = \omega_G(N)$, then we call M and N G-equivalent. If for a classical machine M, there exists a machine $N \in \mathfrak{C}$ such that $\omega_G(M) = \omega_G(N)$, where $\mathfrak{C}$ denotes a class of machines, then we say that M is '$[G, \mathfrak{C}]$-realizable'. $E_G(\mathfrak{C})$ denotes the collection of classical machines that are $[G, \mathfrak{C}]$-realizable.

A reduction $\mathcal{R}$ is represented by $(G^{\mathrm{ext}}(\mathcal{B}), \mathcal{T}, G^{\mathrm{int}}(\mathcal{A}))$, where G^{ext} and G^{int} denote the external game and the internal game, respectively, and transformer $\mathcal{T}$ transforms an adversary $\mathcal{A}$ in G^{int} into an adversary $\mathcal{B} = \mathcal{T}(\mathcal{A})$ in G^{ext}. In particular, if $\mathcal{R}$ is a black-box reduction (i.e., transformation $\mathcal{T}$ does not look into the inner workings of adversary $\mathcal{A}$), then the output of $\mathcal{T}$ can be viewed as an oracle machine with access to $\mathcal{A}$, denoted by $T^{\mathcal{A}}$. A meaningful reduction $\mathcal{R} = (G^{\mathrm{ext}}(\mathcal{B}), \mathcal{T}, G^{\mathrm{int}}(\mathcal{A}))$ is supposed to have some properties as described below. Let $\mathfrak{A}$ and $\mathfrak{B}$ denote two classes of machines. We say $\mathcal{R}$ is

1. $\mathfrak{A}$-compatible, if $\forall \mathcal{A} \in \mathfrak{A}$, $G^{\mathrm{int}}(\mathcal{A})$ and $G^{\mathrm{ext}}(\mathcal{T}(\mathcal{A}))$ are well defined, i.e., $\mathcal{A}$ and $\mathcal{T}(\mathcal{A})$ respect the game specifications;

2. $(\mathfrak{A}, \mathfrak{B})$-consistent, if $\mathcal{R}$ is $\mathfrak{A}$-compatible and $\forall \mathcal{A} \in \mathfrak{A}$, $\mathcal{T}(\mathcal{A}) \in \mathfrak{B}$—we denote such a reduction $\mathcal{R}$ by $(G^{\mathrm{ext}}(\mathfrak{B}), \mathcal{T}, G^{\mathrm{int}}(\mathfrak{A}))$ or $\mathcal{R}(\mathfrak{A}, \mathfrak{B})$;

3. value-dominating, if $\omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{A})) = \omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{B}))$ whenever $\omega_{G^{\mathrm{int}}}(\mathcal{A}) = \omega_{G^{\mathrm{int}}}(\mathcal{B})$;

4. $(\alpha_{\mathrm{succ}}, \mathfrak{A})$-effective, if $\forall \mathcal{A} \in \mathfrak{A}, \omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{A})) \geq \alpha_{\mathrm{succ}}(\omega_{G^{\mathrm{int}}}(\mathcal{A}))$, where $\alpha_{\mathrm{succ}} : \mathbb{R}^+ \to \mathbb{R}^+$ denotes a function bounded by the global security parameters.

To analyze the security of a classical cryptographic scheme in the provable security paradigm, an effective reduction should be built and it proceeds mainly as follows:

1. A security requirement is formalized by a game G^{int}, i.e., $(G^{\mathrm{int}}(\mathfrak{A}), \varepsilon_{\mathfrak{A}})$, where the adversary in G^{int} is restricted to a particular class $\mathfrak{A}$ and $\varepsilon_{\mathfrak{A}}$ represents the upper bound on the success

probability that any adversary in $\mathfrak{A}$ can win G^{int}. Namely, $\omega_{G^{\text{int}}}(\mathfrak{A}) \leq \varepsilon_{\mathfrak{A}} \in (0,1]$.

2. A computational assumption is formalized by another game G^{ext}, i.e., $(G^{\text{ext}}(\mathfrak{B}), \varepsilon_{\mathfrak{B}})$, where the adversary in G^{ext} is restricted to class $\mathfrak{B}$ and $\varepsilon_{\mathfrak{B}}$ is the upper bound of the success probability in G^{ext}. Namely, $\omega_{G^{\text{ext}}}(\mathfrak{B}) \leq \varepsilon_{\mathfrak{B}} \in (0,1]$.

3. An $(\mathfrak{A},\mathfrak{B})$-consistent reduction $\mathcal{R} = (G^{\text{ext}}(\mathfrak{B}), \mathcal{T}, G^{\text{int}}(\mathfrak{A}))$ is constructed, if its security follows that $\mathcal{R}$ is α_{succ}-effective with $\alpha_{\text{succ}} \geq \varepsilon_{\mathfrak{B}}/\varepsilon_{\mathfrak{A}}$. It implies that if there exists an adversary $\mathcal{A} \in \mathfrak{A}$ with $\omega_{G^{\text{int}}}(\mathcal{A}) > \varepsilon_{\mathfrak{A}}$, then an adversary $\mathcal{T}(\mathcal{A}) \in \mathfrak{B}$ exists such that $\omega_{G^{\text{ext}}}(\mathcal{T}(\mathcal{A})) \geq \alpha_{\text{succ}} \cdot \omega_{G^{\text{int}}}(\mathcal{A}) > \varepsilon_{\mathfrak{B}}$.

Now let $\mathcal{R} = (G^{\text{ext}}(\mathfrak{B}), \mathcal{T}, G^{\text{int}}(\mathfrak{A}))$ be a classical reduction as described before. Let $\hat{G}^{\text{ext}}(\hat{\mathfrak{B}})$ and $\hat{G}^{\text{int}}(\hat{\mathfrak{A}})$ be two extended games in the quantum setting, where the adversaries are restricted to classes of quantum machines $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$. Here $\hat{G}^{\text{ext}}(\hat{\mathfrak{B}})$ and $\hat{G}^{\text{int}}(\hat{\mathfrak{A}})$ respect the game specifications of G^{ext} and G^{int}, respectively. Similarly, $(\hat{G}^{\text{int}}(\hat{\mathfrak{A}}), \varepsilon_{\hat{\mathfrak{A}}})$ formalizes a security requirement against quantum adversaries in $\hat{\mathfrak{A}}$ with $\omega_{\hat{G}^{\text{int}}}(\hat{\mathfrak{A}}) \leq \varepsilon_{\hat{\mathfrak{A}}}$, and $(\hat{G}^{\text{ext}}(\hat{\mathfrak{B}}), \varepsilon_{\hat{\mathfrak{B}}})$ formalizes a computational assumption against quantum adversaries in class $\hat{\mathfrak{B}}$. Now we would like to explore whether there exists a reduction $\hat{\mathcal{R}}(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$ that can keep similar properties to those of $\mathcal{R}(\mathfrak{A}, \mathfrak{B})$. To achieve this goal easily, we will apply the notion of class-respectful reduction, which is sketched by two conditions in Definition 10 for checking if a classical reduction $\mathcal{R}(\mathfrak{A}, \mathfrak{B})$ can be lifted into the quantum setting.

Definition 10 (β-$(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-respectful reduction, cf. Definition 3 in Song (2014)) Let $\mathcal{R}$ be a classical reduction $(G^{\text{ext}}(\mathfrak{B}), \mathcal{T}, G^{\text{ext}}(\mathfrak{A}))$ and $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$ denote two quantum machine classes. We say $\mathcal{R}$ is β-$(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-respectful for some $\beta \in \mathbb{R}^+$ if $\mathcal{R}$ satisfies the following two conditions:

1. $(\beta, \hat{\mathfrak{A}})$-extendable, if $\mathcal{R}$ is $E_{G^{\text{int}}}(\hat{\mathfrak{A}})$-compatible and $(\beta, E_{G^{\text{int}}}(\hat{\mathfrak{A}}))$-effective;

2. $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-closed, if $\mathcal{R}$ is $(E_{G^{\text{int}}}(\hat{\mathfrak{A}}), E_{G^{\text{ext}}}(\hat{\mathfrak{B}}))$-consistent.

Note that the above notions can be interpreted for checking whether a classical reduction can still make sense when the adversaries are restricted to the class of quantum machines β-$(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$, where the internal and external adversaries are restricted to $\hat{\mathfrak{A}}$ and $\hat{\mathfrak{B}}$, respectively, and β is a value related to the effectiveness of such a reduction. If a classical reduction $\mathcal{R}$ satisfies the above two conditions, by Theorem 3 (cf. Song (2014) for its proof), we can directly show that an effective quantum reduction exists. Furthermore, if all the classical reductions derived from the security proof of a post-quantum primitive satisfy Definition 10, then we say this primitive can preserve its security in the quantum setting.

Theorem 3 (Quantum lifting for game-preserving reductions, cf. Theorem 1 in Song (2014)) If $\mathcal{R}(\mathfrak{A}, \mathfrak{B})$ is β-$(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-respectful, then there exists an $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-consistent reduction $\hat{\mathcal{R}}(\hat{\mathfrak{A}}, \hat{\mathfrak{B}}) := (G^{\text{ext}}(\hat{\mathfrak{B}}), \hat{\mathcal{T}}, G^{\text{int}}(\hat{\mathfrak{A}}))$ that is $(\beta, \hat{\mathfrak{A}})$-effective.

6.2 Application of class-respectful reduction

In this section, we apply Definition 10 to the derived reductions of dm^{mode} and show that these reductions are class-respectful reductions. Therefore, by Theorem 3, the classical security of dm^{mode} can be lifted into the quantum setting.

We note that the provable security of dm^{mode} is achieved by the simulation-based paradigm, where the main ingredients of its UC-security proof are several hybrid arguments defined in terms of Adv and Sim. The hybrid argument is a proof technique to show that two distributions are computationally indistinguishable by building a sequence of polynomial distributions (called 'hybrids') between the original two distributions and showing that each two adjacent hybrids are indistinguishable.

When we recall the classical UC-security of dm^{mode}, we can find that there exists a reduction $\mathcal{R} = (G^{\text{ext}}, \mathcal{T}, G^{\text{int}})$ whenever the final computational indistinguishability is obtained by distinguishing between the real world and the ideal world in different corruption cases. In each corruption case, we can formalize the distinguishing game between the real world and the ideal world as a general form, which is executed between a challenger $\mathcal{C}$ and a distinguisher $\mathcal{D}$ and viewed as G^{int} in reduction $\mathcal{R}$. This internal game is denoted by $G_{\mathcal{Z},\text{Sim(Adv)}}$, where Sim(Adv) represents that Sim runs a copy of Adv as its subroutine. Game $G_{\mathcal{Z},\text{Sim(Adv)}}$ is defined as follows:

1. $\mathcal{C}$ flips a random coin $i \in_R \{0,1\}$. If $i = 0$, $\mathcal{C}$ runs an execution of the real world and sends the view of Adv to $\mathcal{D}$; if $i = 1$, $\mathcal{C}$ runs an execution of the ideal world, and sends the output of Sim to $\mathcal{D}$.

2. $\mathcal{D}$ collects all messages received from $\mathcal{C}$, and outputs one bit i' to $\mathcal{C}$.

3. $\mathcal{C}$ outputs success if $i = i'$ or failure otherwise.

We note that the environment machine $\mathcal{Z}$ plays the role of $\mathcal{D}$ in distinguishing game $G_{\mathcal{Z},\mathrm{Sim(Adv)}}$. Since $\mathrm{dm}^{\mathrm{mode}}$ is built upon LWE hardness, a classical reduction $\mathcal{R} := (G^{\mathrm{ext}}, \mathcal{T}, G^{\mathrm{int}} := G_{\mathcal{Z},\mathrm{Sim(Adv)}})$ can be built, where we define G^{ext} as a game of distinguishing the LWE distribution from the uniform distribution.

Now we can apply Definition 10 to check whether reduction $\mathcal{R} := (G^{\mathrm{ext}}, \mathcal{T}, G^{\mathrm{int}} := G_{\mathcal{Z},\mathrm{Sim(Adv)}})$ is a class-respectful reduction or not, where the class is restricted as a quantum machine class. It is clear that the class of internal adversaries $\mathfrak{A}$ and the class of external adversaries $\mathfrak{B}$ in $\mathcal{R} := (G^{\mathrm{ext}}, \mathcal{T}, G^{\mathrm{int}} := G_{\mathcal{Z},\mathrm{Sim(Adv)}})$ are both polynomial-time classical machine classes. In the application of Definition 10, we restrict the class of internal adversaries $\hat{\mathfrak{A}}$ and the class of external adversaries $\hat{\mathfrak{B}}$ in the quantum setting to both polynomial-time quantum machine classes, denoted by $\mathcal{Q}$.

During this check, the first condition of Definition 10 is usually easier to check than the second condition. However, if the classical reduction $\mathcal{R}$ is obtained in a black-box pattern with an additional condition that $\mathcal{R}$ is straight-line (i.e., the output machine of $\mathcal{T}$ on $\mathcal{A}$ is the form of $T^{\mathcal{A}}$ and $\mathcal{A}$ is run in a straight-line till completion without rewinding), then the second condition of Definition 10 can be checked easily by Theorem 4 (cf. Song (2014) for its proof).

Theorem 4 (Straight-line reduction) Let $\mathcal{R} = (G^{\mathrm{ext}}(\mathfrak{B}), \mathcal{T}, G^{\mathrm{int}}(\mathfrak{A}))$ be a classical reduction, where $\mathfrak{A}$ and $\mathfrak{B}$ are both classical polynomial-time machines. Let $\hat{\mathfrak{A}}$ and $\hat{\mathfrak{B}}$ be quantum polynomial-time machines $\mathcal{Q}$. If $\mathcal{R}$ is black-box, straight-line, $\hat{\mathfrak{A}}$-compatible, and value-dominating, we say $\mathcal{R}$ is $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$-closed.

Now we show that whenever the computational indistinguishability is obtained in the security proof of $\mathrm{dm}^{\mathrm{mode}}$, it will directly fall into the game-preserving reduction framework by checking whether these inherent reductions satisfy two conditions of class-respectful reduction. Specifically, in each corruption case, a classical reduction $\mathcal{R} := (G^{\mathrm{ext}}, \mathcal{T}, G^{\mathrm{int}} := G_{\mathcal{Z},\mathrm{Sim(Adv)}})$ can be obtained, where the environment machine $\mathcal{Z}$ plays the role of $\mathcal{A}$ in $G^{\mathrm{int}}(\mathcal{A})$. When we replace $\mathcal{Z}$ with a quantum polynomial-time machine $\hat{\mathcal{Z}}$, then $\mathcal{Z} \in E_{G^{\mathrm{int}}}(\mathcal{Q})$, which indicates that there is a machine $\hat{\mathcal{Z}} \in \mathcal{Q}$ such that $\omega_{G^{\mathrm{int}}}(\mathcal{Z}) = \omega_{G^{\mathrm{int}}}(\hat{\mathcal{Z}})$. Since $G^{\mathrm{int}}(\mathcal{Z})$ and $G^{\mathrm{ext}}(\mathcal{T}(\mathcal{Z}))$ are well-defined, $\mathcal{R}$ is $E_{G^{\mathrm{int}}}(\mathcal{Q})$-compatible. Furthermore, $\mathcal{R}$ is $(\beta, E_{G^{\mathrm{int}}}(\mathcal{Q}))$-effective, where β satisfies $\omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{Z})) \geq \beta(\omega_{G^{\mathrm{int}}}(\mathcal{Z})$ for any $\mathcal{Z} \in E_{G^{\mathrm{int}}}(\mathcal{Q})$. Thus, $\mathcal{R}$ is $(\beta, \mathcal{Q})$-extendable.

Based on the simulator construction of $\mathrm{dm}^{\mathrm{mode}}$ in each corruption case, we know that the corresponding reduction $\mathcal{R}$ is black-box and straight-line, i.e., $\mathcal{T}(\mathcal{Z}) = T^{\mathcal{Z}}$. Since $\forall \hat{\mathcal{Z}} \in \mathcal{Q}$ both $G^{\mathrm{int}}(\hat{\mathcal{Z}})$ and $G^{\mathrm{ext}}(\mathcal{T}(\hat{\mathcal{Z}}))$ are well-defined, $\mathcal{R}$ is $\mathcal{Q}$-compatible. In addition, $\mathcal{R}$ is value-dominating, which means that if $\omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{A})) = \omega_{G^{\mathrm{ext}}}(\mathcal{T}(\mathcal{B}))$, we always have $\omega_{G^{\mathrm{int}}}(\mathcal{A}) = \omega_{G^{\mathrm{int}}}(\mathcal{B})$. Then we know that $\mathcal{R}$ is $(\mathcal{Q}, \mathcal{Q})$-closed by Theorem 4. According to Definition 10, we can claim that such $\mathcal{R}$ is a $(\beta, \mathcal{Q})$-respectful reduction, i.e., a $(\beta, \mathcal{Q})$-effective reduction. It implies that in the current corruption case $\mathrm{dm}^{\mathrm{mode}}$ is secure against quantum adversaries restricted in $\mathcal{Q}$. After checking for all derived reductions, we can say that $\mathrm{dm}^{\mathrm{mode}}$ is still secure in the quantum setting.

7 Discussion and conclusions

The widespread use of oblivious transfer (OT) and the concerns about the quantum resistance of post-quantum cryptographic constructions motivate us to explore the feasible security of the post-quantum OT protocol in the quantum world.

In this work, we use three tools and give a comprehensive quantum security analysis of an existing lattice-based OT protocol. These three tools are all existing techniques that can be applied to the security analysis of classical protocols in a quantum world. We first prove the quantum security of this lattice-based OT protocol using the quantum lifting theorem in Section 4, and then apply another two tools (SHA framework and game-preserving reduction) to this OT protocol for a comprehensive analysis. From some points of view, the discussions shown in Sections 5 and 6 seem somewhat redundant since they do not contribute to the proof in Section 4 and simply work for the same argument using different tools. The purpose of what we have done in Sections 5 and 6 is to show that our analysis is compatible with that of the other two frameworks. Actually, the application of the quantum lifting theorem in our analysis also obeys the hybrid arguments technique, which is the main ingredient of the SHA framework; i.e., two adjacent machines are indistinguishable for

a quantum distinguisher. Game-preserving reduction attempts to achieve the same goal, but instead it interprets the security proof of classical protocols as several reductions, i.e., checking whether these reductions can meet some requirements (regarding the quantum adversary), and thus lifting the whole security proof into the quantum setting. It seems that they are two different kinds of methods to prove quantum security; however, these three tools give the same results for the lattice-based OT in our work. We think that one can use any tool as the circumstances may require; sometimes it may be easier to approach the desired security from one point of view rather than the other.

Our work can be regarded as an application of the quantum security analysis framework for protocols. Furthermore, it motivates us to do some more interesting work. The cryptographic applications based on the standard LWE problem are rather inefficient in practice due to the large key size, the error sampling, and large modulus operations. For this concern, an algebraic variant of LWE called 'ring learning with errors (RLWE)' was proposed, and is provably at least as hard as the worst-case problems in ideal lattice setting (Lyubashevsky *et al.*, 2013). Many applications based on the LWE problem can be made more efficient by transforming the original cryptosystem into a variant of the RLWE problem. Therefore, searching for an efficient RLWE-based OT protocol is an interesting research topic. One possible way is to transform $\mathrm{dm}^{\mathrm{mode}}$ into an RLWE-based variant using some results on trapdoor applications in ideal lattices (Lai *et al.*, 2014). We believe that the three security analysis frameworks used in this work will also be available for this RLWE-based OT variant.

For a practical deployment of the OT protocol, a more efficient method called 'OT extension' (Ishai *et al.*, 2003) has been proposed, which enables a relatively small number of base OTs to compute a very large number of OTs at low cost. This OT extension protocol can be viewed as a hybrid of asymmetric and symmetric constructions, i.e., extending a small number of base (asymmetric) OTs via symmetric cryptography. In addition, Zhandry (2012) showed three secure classical constructions of pseudorandom functions in a quantum setting. Based on these two results, an interesting question arises: if we use a post-quantum OT protocol that is provably secure in the quantum setting as the base OT in the OT extension protocol, and some proper quantum secure pseudorandom functions as the symmetric tools in the OT extension protocol, can we obtain a post-quantum OT extension protocol which is secure in the quantum setting? If this post-quantum OT extension indeed exists, then we will have an efficient and practical way to implement a large use of post-quantum OTs without loss of security in the quantum world.

References

Bernstein, D.J., Buchamann, J., Dahmen, E., 2009. Post-Quantum Cryptography. Springer, Berlin. https://doi.org/10.1007/978-3-540-88702-7

Canetti, R., 2001. Universally composable security: a new paradigm for cryptographic protocols. Proc. 42nd IEEE Symp. on Foundations of Computer Science, p.136-145. https://doi.org/10.1109/SFCS.2001.959888

Damgård, I., Funder, J., Nielsen, J.B., *et al.*, 2014. Superposition attacks on cryptographic protocols. *LNCS*, **8317**:142-161. https://doi.org/10.1007/978-3-319-04268-8_9

Even, S., Goldreich, O., Lempel, A., 1985. A randomized protocol for signing contracts. *Commun. ACM*, **28**(6):637-647. https://doi.org/10.1145/3812.3818

Fehr, S., Katz, J., Song, F., *et al.*, 2013. Feasibility and completeness of cryptographic tasks in the quantum world. *LNCS*, **7785**:281-296. https://doi.org/10.1007/978-3-642-36594-2_16

Gentry, C., Peikert, C., Vaikuntanathan, V., 2008. Trapdoors for hard lattices and new cryptographic constructions. Proc. 40th Annual ACM Symp. on Theory of Computing, p.197-206. https://doi.org/10.1145/1374376.1374407

Gilboa, N., 1999. Two party RSA key generation. *LNCS*, **1666**:116-129. https://doi.org/10.1007/3-540-48405-1_8

Hallgren, S., Smith, A., Song, F., 2011. Classical cryptographic protocols in a quantum world. *LNCS*, **6841**:411-428. https://doi.org/10.1007/978-3-642-22792-9_23

Hallgren, S., Smith, A., Song, F., 2015. Classical cryptographic protocols in a quantum world. Cryptology ePrint Archive, 2015/687. http://eprint.iacr.org/2015/687

Ishai, Y., Kilian, J., Nissim, K., *et al.*, 2003. Extending oblivious transfers efficiently. *LNCS*, **2729**:145-161. https://doi.org/10.1007/978-3-540-45146-4_9

Lai, R.W.F., Cheung, H.K.F., Chow, S.S.M., 2014. Trapdoors for ideal lattices with applications. *LNCS*, **8957**:239-256. https://doi.org/10.1007/978-3-319-16745-9_14

Lyubashevsky, V., Peikert, C., Regev, O., 2013. On ideal lattices and learning with errors over rings. *J. ACM*, **60**(6):43. https://doi.org/10.1145/2535925

Micciancio, D., Regev, O., 2009. Lattice-based cryptography. *In*: Bernstein, D.J., Buchmann, J., Dahmen, E. (Eds.), Post-Quantum Cryptography. Springer, Berlin, p.147-191. https://doi.org/10.1007/978-3-540-88702-7_5

Nielsen, M.A., Chuang, I.L., 2010. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

Peikert, C., 2009. Some recent progress in lattice-based cryptography. *LNCS*, **5444**:72. https://doi.org/10.1007/978-3-642-00457-5_5

Peikert, C., Vaikuntanathan, V., Waters, B., 2008. A framework for efficient and composable oblivious transfer. *LNCS*, **5157**:554-571. https://doi.org/10.1007/978-3-540-85174-5_31

Rabin, M.O., 1981. How to Exchange Secrets with Oblivious Transfer. Technical Report No. TR-81, Aiken Computation Lab, Harvard University, Cambridge, MA. http://eprint.iacr.org/2005/187

Regev, O., 2005. On lattices, learning with errors, random linear codes, and cryptography. Proc. 37th Annual ACM Symp. on Theory of Computing, p.84-93. https://doi.org/10.1145/1060590.1060603

Sendrier, N., 2011. Code-based cryptography. *In*: van Tilborg, H.C.A., Jajodia, S. (Eds.), Encyclopedia of Cryptography and Security. Springer, New York, p.215-216. https://doi.org/10.1007/978-1-4419-5906-5_378

Shor, P.W., 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, **26**(5):1484-1509. https://doi.org/10.1137/S0097539795293172

Song, F., 2014. A note on quantum security for post-quantum cryptography. *LNCS*, **8772**:246-265. https://doi.org/10.1007/978-3-319-11659-4_15

Unruh, D., 2010. Universally composable quantum multiparty computation. *LNCS*, **6110**:486-505. https://doi.org/10.1007/978-3-642-13190-5_25

Unruh, D., 2012. Quantum proofs of knowledge. *LNCS*, **7237**:135-152. https://doi.org/10.1007/978-3-642-29011-4_10

Watrous, J., 2009. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, **39**(1):25-58. https://doi.org/10.1137/060670997

Zhandry, M., 2012. How to construct quantum random functions. IEEE 53rd Annual Symp. on Foundations of Computer Science, p.679-687. https://doi.org/10.1109/FOCS.2012.37