Journal of Network Intelligence ©2016 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **2**, Number **3**, August **2017**

Equational Security of a Lattice-based Oblivious Transfer Protocol

Mo-Meng Liu

State Key Laboratory of Integrated Service Networks
Xidian University
No.2 South Taibai Road, Xian, Shaanxi, 710071, China
liumomeng@gmail.com

Yu-Pu Hu

State Key Laboratory of Integrated Service Networks
Xidian University
No.2 South Taibai Road, Xian, Shaanxi, 710071, China
yphu@mail.pxidian.edu.cn

Received March 2017; Revised April 2017.

Abstract. *In this paper, we apply a novel framework, called* equational security framework*, to an efficient lattice-based oblivious transfer (OT) protocol which is built upon the hardness of the* learning with errors *(LWE) problem that is believed quantum resistant, and provable secure in the standard* universally composability *(UC) model. This novel framework can provide a concrete mathematical model of communication, and a concise syntax to describe protocol in terms of a set of mathematical equations. This distinguishing feature makes it more easily to analyze the security of protocols when they are deployed in some complex environments (e.g., when composed with other protocols). Our initial motivation intends to prove the equational security of this lattice-based OT protocol to claim its usability in a fully asynchronous environment. However, we found a timing bug during the security proof of this lattice-based OT protocol in the equational security framework. Thus, we accordingly make twice modifications on its traditional OT functionality. Unfortunately, we still cannot obtain the equational security of this lattice-based OT protocol. We remark that our negative result does not mean that we find some specific faults in the original security proof of this OT protocol, but it implies that this protocol is not suitable to be used in the fully asynchronous execution environments.*

Keywords: Oblivious transfer, Lattice-based cryptography, Equational security framework, Universally composability

1. **Introduction.**

As a fundamental cryptographic primitive, *oblivious transfer* (OT) was firstly introduced by Rabin [1] to depict a cryptographic scenario between two parties, where one party (called the sender) sends a message to the other party (called the receiver) with requiring that the receiver can only obtain this message with probability $\frac{1}{2}$ and the sender remains oblivious as to whether the message has been received or not. In addition, an OT protocol based on RSA cryptosystem was proposed to implement this tricky conception. Afterwards, a more useful form called *1-out-of-2* OT [2] was developed to build protocols for secure multiparty computation. 1-out-of-2 OT is also an execution of a two-party computation, where the sender takes as input a message pair and the receiver chooses one bit as his input. After the interaction between two parties, the receiver can retrieve his chosen message without knowing any knowledge of the other message and the sender is unaware of the receiver's message choice. Moreover, 1-out-of-2 OT can be trivially transformed into *1-out-of-N* OT, where the sender runs a database consisting of N message entries and the receiver can retrieve one out of N messages by his choice. In the rest of paper, unless otherwise noted, we abbreviate 1-out-of-2 OT as OT for clarity.

Due to the above simple functionality of OT, it can be utilized to securely and efficiently implement some mathematical operations (e.g., secure two-party multiplication [3]) among several parties. Therefore, OT is usually taken as the basic building block to construct more complex cryptographic protocols in countless cryptographic literatures. In [4] and [5], efficient OT protocols are built upon the decisional Diffie-Hellman (DDH) assumption. However, their security is shown in *half-simulation* paradigm, where an ideal simulator can only be constructed for a cheating receiver. Therefore, this type of OT protocols cannot be sufficiently secure when integrated into a larger protocol. With this concern, many works attempt to achieve *fully-simulation* security, i.e., the ideal simulators can be built for both parties. There are several results (e.g., [6]) which aim to construct fully-simulation secure OT protocol, however, most of them are secure in the *stand-alone model* (cf. [7]), which guarantees that the protocol is secure under sequential composition instead of *parallel* or *concurrent* composition. However, concurrent composability can bring great advantages on efficiency when protocols are executed in a complex environment, such as the Internet, where it can help to save communication rounds by running protocols parallelly. This kind of security requirement is captured by a well-known framework, called *universally composability* (UC) model [8]. That is, the protocol which is provable secure in the UC model can maintain its security when running concurrently with arbitrary other secure and insecure protocols.

However, one main feature of the UC framework is that any cryptographic task can be expressed as an ideal functionality (e.g., OT functionality) and almost any network environment (e.g., fully asynchronous execution environments) is possible in this UC model. This highly generality makes it hard to use this UC formalization when specifying and analyzing concrete protocols in practice. This limitation motivates researchers to explore several variants for simplifying and specializing of the general UC model [9, 10, 11, 12]. For the same prospective, a novel framework called *equational security framework* [13] is proposed to provide a concrete mathematical model and a concise syntax to describe secure computation protocols, i.e., specifying cryptographic protocols by a system of mathematical equations. The framework makes the study of secure computation protocols in a concise, rigorous, notation-wise lightweight manner.

Motivation. In this paper, by applying the equational security framework, we revisit the security of an efficient lattice-based OT protocol [14] which is built upon the hardness of the *learning with errors* (LWE) problem and provable secure in the standard UC framework. Actually, our work is initially inspired by [15], which analyzes two concrete OT protocols in the equational security framework for their equational security. In [15], they first show the equational security of OT extension protocol [16], however, they cannot prove the equational security of an efficient UC secure OT [17] even after some modifications. Therefore, this motivates us to explore whether this lattice-based OT protocol [14] can achieve its equational security or not. Moreover, to the best of our knowledge, the lattice-based OT protocol proposed in [14] is the most efficient OT protocol which is believed quantum resistant (relying on the LWE hardness) and secure in the UC framework (offering the guarantee for arbitrary composition). Thus, if we achieve the equational security of this lattice-based OT protocol, it will make the protocol easily understandable and promisingly utilized in a complex environment (might be quantum adversarial and fully asynchronous).

Our Results. Here we briefly present our main findings in this paper. First, it is clear to see that the UC-security proof of this lattice-based OT protocol [14] is made by using the traditional OT functionality (see later in Section 4 Figure 2). When we apply this traditional OT functionality to the equational security proof of this lattice-based OT protocol, we find a timing bug in the case that when the sender is corrupted, so that there always exists an environment which can distinguish the real world from the ideal world with overwhelming probability. Therefore, we accordingly modified the OT ideal functionality twice to fix this bug, however, the result is still negative, i.e., this lattice-based OT protocol [14] cannot obtain its security in the equational security framework.

Roadmap. We introduce some basic notions and backgrounds in Section 2, and recall the lattice-based OT protocol [14] we mentioned in Section 3. Our work is mainly presented in Section 4 to explore the equational security of this lattice-based OT protocol. Moreover, we give the conclusion of this paper in Section 5.

2. Preliminaries.

In this section, some relevant backgrounds are introduced in two parts, where we briefly introduce the equational security framework in Section 2.1, and lattice-based cryptography and UC framework which are related to the lattice-based OT protocol [14] in Section 2.2. For clarity, most notations and concepts we recall respect their definitions shown in [8, 13, 14, 15]. In addition, we assume that the readers are familiar with the simulation-based security [7].

2.1. Equational security framework.

To study protocols in a concise, rigorous, notation-wise lightweight manner, *equational security framework* [13] is proposed to provide a concrete mathematical model and a concise syntax to describe cryptographic protocols by a system of mathematical equations. Its underlying mathematical foundation relies on *domain theory* [18, 19, 20]. Since to explain how domain theory can be applied to the equational security framework is not essential to the readers, we skip this illustration and refer to [13, 15] for a detailed treatment. Here, we only introduce some necessary backgrounds and notions of the equational security framework.

In a nutshell, the equational security framework can be viewed as a network with nodes representing computation units and edges modeling communication channels. Each channel is associated with a *partially ordered set* (i.e., poset) which represents all possible messages or sequences of messages that may be transmitted on the channel over time, where the temporal evolution is described by the *partial order relation.*

For example, if a channel is allowed to transmit an arbitrary number of messages from a basic set X, then it can be modeled by a poset $(X^*, \leq)$, where $X^* = \{(x_1, ..., x_k) : k \geq 0, \forall i, x_i \in x\}$ represents a set of finite messages in X, and the partial ordering relation $\leq$ (reflexive, transitive and antisymmetric) is used to describe a sequence of transmission observations at different points in time, denoted by a chain $x_1 \leq x_2 \leq ... \leq x_k$. Here x_{i+1} is a possible extension or future of x_i if for any two histories $x_i \leq x_{i+1}$.

Another common example of poset, denoted by $X_\perp = X \bigcup \{\perp\}$, consisting of all messages in X, and a bottom element $\perp$ representing the case that no message has been transmitted yet, where its flat partial order satisfies that $x \leq y$ if and only if $x = \perp$ or $x = y$. This kind of poset $X_\perp$ models a communication channel which is capable of transmitting only a single message from X.

In the equational security framework, each node can be viewed as a unit with one input and one output, where the input (*resp.* the output) is the Cartesian product of incoming channels (*resp.* outgoing channels). Each computational unit at each node can be modeled as a function $f : C_1 \to C_2$ from the incoming channels to the outgoing channels such that $f(h_1) \leq f(h_2)$ in C_2 if for any $h_1 \leq h_2$ in C_1. This natural monotonicity implies that once a message has been transmitted, the sender cannot take it back in time.

Now we give a more detailed introduction of equational security framework [13] in terms of its application in this paper. All considered posets in this paper are *complete partial orders* (CPOs), i.e., for any chain $x_1 \leq x_2 \leq ...$ in poset $(X, \leq)$, it has a least upper bound $\sup_i x_i$. In addition, all CPOs have a minimal element $\perp = \sup \emptyset$, i.e., for all $x \in X$, it satisfies the relation $\perp \leq x$. These posets are naturally endowed with the *Scott topology*, where a subset $A \subseteq X$ is *closed* if for all $x \in A$, $y \leq x$ implies $y \leq A$, and any chain in A has a least upper bound in A. *Open* sets are defined as the complements of closed sets. Note that all CPOs considered in this paper are *flat* CPOs. That is, for any finite set X, it is extended with $\perp$ to form a flat CPO $X_\perp$, where the partial order in this flat CPO consists of $\perp \leq x$ for all $x \in X$. In addition, for any set X, $x \leftarrow X_\perp$ represents selecting an element $x \neq \perp$ uniformly at random from X. Then the functions $f : X \to Y$ between sets can be lifted into the functions $f : X_\perp \to Y_\perp$ between the corresponding flat CPOs by setting $f(\perp) = \perp$.

Moreover, the equational framework also specifies the probabilistic behaviors of protocols, represented by the probabilistic functions between sets of distributions with proper ordering relation. A *probability distribution* on a CPO X is a function $p : X \to [0, 1]$ such that $p(A) + p(B) = p(A \cup B)$ for all disjoint $A, B \subseteq X$ and $p(X) = 1$. We let n be a security parameter. If for all $x \in X$, $p(x) < n^{-c}$ for any constant $c > 1$, we say such a probability p is *negligible*. Then p is *overwhelming* if $1 - p$ is negligible. Note that the set of *probability distributions* over a CPO X, denoted by $D(X)$, is also a CPO. For any two distributions p and q over $D(X)$, we have $q \leq q$ (in $D(X)$) if and only if $p(A) \leq q(A)$ for any open subset $A \subseteq X$.

Now we recall some notations defined in [15] for the application of equational security framework in our work. A flat CPO is denoted by $\mathbb{X}$ with a bottom element $\perp \in \mathbb{X}$, where its partial order satisfies that $x_1 \leq x_2$ if and only if $x_1 = \perp$ or $x_1 = x_2$. This flat CPO $\mathbb{X}$ models a simple communication channel which can transmit a single message from $\mathbb{X}/\{\perp\}$, where $\perp$ represents the state that no message has been transmitted yet. We use $\mathbb{X}^{\times 2} = \{(x, y) : x, y \in \mathbb{X}, x \neq \perp; y \neq \perp\}_\perp$ to represent the set of strict pairs over $\mathbb{X}$. For any pair $z \in \mathbb{X}^{\times 2}$, z consists of two elements, i.e., $z[0]$ and $z[1]$ such that $z[i] = \perp$ when $z = \perp$ or $i = \perp$. Now we let $\langle x, y\rangle$ denote the operation of combining two elements into a strict pair, thus z can be written as the form of $\langle z[0], z[1]\rangle$. Note that $\langle x, \perp\rangle = \langle \perp, y\rangle = \perp$, and therefore $\langle x, \perp\rangle[0] = \langle \perp, y\rangle[1] = \perp$ even when $x, y \neq \perp$. In addition, for any pairs $\langle x_0, x_1\rangle$, $\langle y_0, y_1\rangle$, strict function f and strict binary operation $\odot$, it is easily to verify that $f(\langle x_0, x_1\rangle[i]) = \langle f(x_0), f(x_1)\rangle[i]$ and $\langle x_0, x_1\rangle[i] \odot \langle y_0, y_1\rangle[i] = \langle x_0 \odot y_0, x_1 \odot y_1\rangle[i]$. For clarity, in the rest of the paper we let the CPO $\mathbb{S} = \{\top\}_\perp$ represent signals, i.e., the messages without information content, the CPO $\mathbb{B} = \{0, 1\}_\perp$ represent single bit message (used for selecting an element from a pair), and the CPO $\mathbb{M}_\ell = \{0, 1\}^\ell_\perp$ describe bit-strings of length ℓ. Moreover, we utilize $x!y = \langle x, y\rangle[1]$ to represent the operation of guarding y by x such that $x!y = y$ except for the case $x = \perp$. This expression can be used for delaying the transmission of y until after x is received. In addition, the expression $x! = x!\top$ can be used for testing $x > \perp$.

Advantages of equational security framework. The most outstanding feature of the equational framework is that it can provide a mathematically clean syntax to specify the processes of computation nodes by a set of mathematical equations. This makes it easy to represent composition by combining equations together. Since composition is independent of the composition order, the behavior of a network can be described by the equations of each node with the edges connecting the nodes. with using equations in this way, it can also provide a concise method to deduce transformations as well. For instance, equivalent components can be replaced by each other if they have equivalent equations. When considering probabilistic behaviors, if a component is indistinguishable from another component, then they can be interchangeably used with negligible impact on the behavior of the entire system.

2.2. **Some backgrounds related to the lattice-based OT protocol.** In this section, we introduce some necessary backgrounds related to the lattice-based OT protocol [14] we later introduce in Section 3, including lattice-based cryptography and UC framework. We first recall some notations defined in [8, 14].

2.2.1. *Notations.* We use bold lowercase letters to denote vectors, e.g., $\mathbf{v}$, and bold capital letters to denote matrices, e.g, $\mathbf{M}$. The notation $\mathbf{v}^T$ represents the transpose of vector $\mathbf{v}$. 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 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 normal 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.

Throughout this paper, $n \in \mathbb{N}$ denotes a security parameter which is taken as an implicit input to all cryptographic algorithms. We classify the growth of functions by 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 $f(n) = O(g(n) \log^c n)$ for some fixed constant c. Let $\mathrm{poly}(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant c. We say that some unspecified function $f(n)$ is *negligible* in n if $f(n) = o(n^{-c})$ for any constant c. Let $\mathrm{negl}(n)$ denote such a function $f(n)$ which is negligible in n. We say that a probability is *overwhelming* if it is $1 - \mathrm{negl}(n)$. Let $X = \{X(n, x)\}_{n \in \mathbb{N}, x \in \{0,1\}^*}$ denote a binary distribution ensemble (i.e., an infinite set of probability distributions over $\{0, 1\}$), where a distribution $X(n, x)$ is associated with each security parameter $n \in \mathbb{N}$ and input $x \in \{0, 1\}^*$. Now we have two such ensembles $X = \{X(n, x)\}_{n \in \mathbb{N}, x \in \{0,1\}^*}$ and $Y = \{Y(n, x)\}_{n \in \mathbb{N}, x \in \{0,1\}^*}$. If $|\Pr(X(n, x) = 1) - \Pr(Y(n, x) = 1)| \leq \mathrm{negl}(n)$, we say that X and Y are *indistinguishable*, denoted by $X \approx Y$.

2.2.2. *Lattice-based cryptography.* Lattice-based cryptography [21, 22] is a promising post-quantum cryptography candidate [23] 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 introduced by Regev [24] is widely utilized and regarded 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 that cannot be efficiently solved by quantum algorithms. Referring to [24], the LWE problem is described as follows:

Definition 2.1 (LWE). *Let $n \geq 1, q \geq 2$ be positive integers, $\mathcal{X}$ be an error distribution over $\mathbb{Z}_q$, and $\mathbf{s}$ be a secret vector following the uniform distribution over $\mathbb{Z}_q^n$. Let $A_{\mathbf{s},\mathcal{X}}$ denote the probability distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at* *random, choosing* $e \in \mathbb{Z}_q$ *according to* $\mathcal{X}$*, and returning the sample* $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*. The LWE problem has two versions, the* search*-version and the* decision*-version.*

Search-*LWE is to find* $\mathbf{s}$ *given access to arbitrarily many independent samples* $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ *from* $A_{\mathbf{s},\mathcal{X}}$*.*

Decision-*LWE is to distinguish an oracle that returns independent samples from* $A_{\mathbf{s},\mathcal{X}}$ *from an oracle that returns independent samples from the uniform distribution on* $\mathbb{Z}_q^n \times \mathbb{Z}_q$*.*

In [24], Regev gave a quantum reduction from worst-case lattice problems to the search-LWE problem. Moreover, he also established the equivalences of the search-LWE problem by using elementary reductions, including a reduction from the (average-case) decision-version to the search-version. Note that in the rest of this paper we denote this decision-LWE problem as the LWE problem. Here we state the result [24] with regarding to the average-case decision-LWE problem, which is denoted by $\mathrm{LWE}_{q,\mathcal{X}}$ and used in the following cryptographic application (see Section 3) as underlying hardness. That is, for certain choices of q and $\mathcal{X}$:

Proposition 2.1 (see [24] Theorem 1.1 and Lemma 4.2)**.** *Let* n*,* 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* $\mathrm{LWE}_{q,\mathcal{X}}$*, then there exists an efficient quantum algorithm for solving the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP) to within* $\tilde{O}(n/\alpha)$ *in the worst case.*

2.2.3. *UC framework.* Since the lattice-based OT protocol proposed in [14] achieves its security in the standard *universally composability* (UC) framework [8], we begin by a brief overview of this model which can provide a strong guarantee that ensuring the protocol remains secure when run concurrently with arbitrary other secure and insecure protocols. The security defined in this model is made by comparison: given a certain task, we assume that there exists a specific machine called the *ideal functionality*, denoted by $\mathcal{F}$, which can securely implement this task. This ideal functionality obtains the inputs of the computation participants and provides them with the desired outputs, where all communication between $\mathcal{F}$ and the involved computation parties are done over secure channels and the whole interaction process can be regarded as an ideal protocol/process (denoted by $\rho^{\mathcal{F}}$) to securely realize this given task.

However, in the real world, we cannot obtain this ideal protocol $\rho^{\mathcal{F}}$. We need a real protocol π to implement $\mathcal{F}$ and intuitively expect that π is no less secure than $\mathcal{F}$. It implies that if there exists an adversary that can do anything during the execution of π, then the adversary can also do the same thing during an execution of $\rho^{\mathcal{F}}$. Due to the required security of $\mathcal{F}$, this adversary cannot attack successfully π either. This is captured by requiring that for any real *adversary* Adv, there exists an ideal adversary, called the *simulator* Sim (constructed from Adv), such that an execution of π with Adv (called the *real world*) is indistinguishable from an execution of $\mathcal{F}$ with Sim (called the *ideal world*). Moreover, in the UC framework there is a new algorithmic entity called *environment* Env, which models the external environment of the current honest parties. The distinguishing feature of the UC framework is that Env can freely interact with both worlds through the whole execution process. If no machine Env can distinguish its interaction with the real world from the one with the ideal world, then we say this real protocol π can *securely UC-emulate* the ideal functionality $\mathcal{F}$ or the ideal process $\rho^{\mathcal{F}}$, i.e., π can achieve *UC-security.*

Referring to [8], we let $\mathcal{F}$ be an ideal functionality. $\mathrm{EXEC}_{\pi,\mathsf{Adv},\mathsf{Env}}(n, x)$ denotes the random variable describing the single bit output of Env when interacting with Adv and running π on an input $x \in \{0, 1\}^*$, where n represents the security parameter. Likewise, the random variable describing the single bit output of Env when interacting with $\mathcal{F}$

and Sim on the same input x is denoted by $\mathrm{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathsf{Env}}(n,x)$. Let $\mathrm{EXEC}_{\pi,\mathsf{Adv},\mathsf{Env}}$ (*resp.* $\mathrm{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathsf{Env}}$) denote the ensemble $\{\mathrm{EXEC}_{\pi,\mathsf{Adv},\mathsf{Env}}(n,x)\}$ (*resp.* $\{\mathrm{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathsf{Env}}(n,x)\}$) of distributions over $\{0,1\}$, where $n \in \mathbb{N}, x \in \{0,1\}^*$. Then the general notion that π can *securely UC-emulate* $\mathcal{F}$ is formally defined as follows.

Definition 2.2 (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* Env *it holds that*

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

where $\approx$ denotes indistinguishability.

3. Lattice-based OT Protocol.

In this section, we will introduce an efficient lattice-based OT protocol [14] which is provable secure in the UC framework. This lattice-based OT is extracted from a *dual-mode* cryptosystem constructed by using some technical tools proposed in [25]. Once such a dual-mode cryptosystem under some certain standard assumption 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 instantiated with the L-WE problem. Then we introduce the derived OT protocol in Section 3.2. In addition, for a clear application of equational framework to this lattice-based OT protocol, we briefly recall the simulator constructions for the UC-security proof of this OT protocol in Section 3.3.

3.1. Dual-mode cryptosystem based on LWE hardness.

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. However, the distinguishing feature of this particular encryption scheme is that it can operate in two modes, i.e., *messy mode* and *decryption mode*, where the mode that the scheme will run in is decided by the trusted setup phase of the dual-mode cryptosystem. In the trusted setup phase, a *common reference string* (denoted by crs) and the corresponding *trapdoor* information τ[1] are created, where crs may be chosen either uniformly random or from a specified distribution. Note that the distribution of crs decides the mode that the dual-mode cryptosystem will run in and also specifies the type of public key used in the encryption.

The instantiation of the dual-mode cryptosystem with the LWE problem relies on some techniques developed in [25], including an LWE-based encryption (a variant of Regev's CPA-secure LWE-based cryptosystem [24]) and an efficient algorithm called $\mathsf{IsMessy}$ (which is used to build the simulator in UC-security proof). In [25], it shows that how to securely embed a trapdoor into the public matrix $\mathbf{A}$ used in this LWE-based encryption cryptosystem, so that when the corresponding trapdoor information is given, the algorithm $\mathsf{IsMessy}$ 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.

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

[1]Note that the trapdoor τ in different modes is different, it will be illustrated later in the description of the dual-mode construction.

approximately r^2. All operations are performed over $\mathbb{Z}_q$.

LWE-Based Encryption

- $\mathsf{LWEKeyGen}(1^n)$: choose a secret key $\mathbf{s} \leftarrow \mathbb{Z}_q^{n\times 1}$ uniformly at random. To generate the public key, choose a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ uniformly at random and an error vector $\mathbf{x} \in \mathbb{Z}_q^{1\times m}$ according to the error distribution $\mathcal{X} = \bar{\Psi}_\alpha$ for some parameter $\alpha = \alpha(n) \in (0,1)$ (here each x_i is chosen independently for all $i \in [m]$). Then compute $(\mathbf{A}, \mathbf{p} = \mathbf{s}^T\mathbf{A} + \mathbf{x})$, where each entry $(\mathbf{a}_i, p_i)$ is a sample from $A_{s,\mathcal{X}}$ and m denotes the number of samples.
- $\mathsf{LWEEnc}(pk = (\mathbf{A}, \mathbf{p}), \mu)$: to encrypt a message $\mu \in \mathbb{Z}_2$, choose a vector $\mathbf{e} \in \mathbb{Z}^m$ from $D_{\mathbb{Z}^m,r}$ and output the ciphertext $(\mathbf{u}, c) = (\mathbf{Ae}, \mathbf{pe} + t(\mu)) = (\mathbf{Ae}, \mathbf{pe} + \mu \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
- $\mathsf{LWEDec}(sk = \mathbf{s}, (\mathbf{u}, c))$: to recover the message μ from the ciphertext, compute $d = c - \mathbf{s}^T\mathbf{u} \in \mathbb{Z}_q$. Output 0 if d is closer to 0 than to $\lfloor q/2 \rfloor$modulo q, otherwise output 1.

Remark 3.1. *The above LWE-based encryption scheme can also be applied when the message is a vector $\boldsymbol{\mu} \in \mathbb{Z}_2^\ell$, where $\ell = poly(n) \geq 1$. The main advantage that the matrix $\mathbf{A}$ can be reused over ℓ different bits enables to fulfill a* multi-session OT *protocol, where it can implement ℓ individual OT executions by using the same $\mathbf{A}$ and obliviously transfer one bit to the receiver in each subsession.*

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

LWE-Based Dual-Mode Cryptosystem

- $\mathsf{SetupMessy}(1^n)$: choose a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ uniformly at random, together with the trapdoor information $\tau = (\mathbf{S}, \mathbf{A})$ as shown in [25]. For each $b \in \{0,1\}$, choose a vector $\mathbf{v}_b \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$. Output (crs, τ).
- $\mathsf{SetupDec}(1^n)$: choose a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random. For each $b \in \{0,1\}$, choose a secret $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ uniformly at random and an error vector $\mathbf{x}_b \leftarrow \mathcal{X}^{1\times m}$, where all entries are chosen independently from the error distribution $\mathcal{X}$. Let $\mathbf{v}_b = \mathbf{s}_b^T\mathbf{A} + \mathbf{x}_b - \mathbf{w}$, $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$, and $\tau = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$. Output (crs, τ).
- $\mathsf{KeyGen}(\mathsf{crs}, \sigma)$: choose a secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random and a vector $\mathbf{x} \leftarrow \mathcal{X}^{1\times m}$. Let $pk = \mathbf{s}^T\mathbf{A} + \mathbf{x} - \mathbf{v}_\sigma$, where $\sigma \in \{0,1\}$ denotes the *decryptable* branch[3]. Let $sk = \mathbf{s}$ and output (pk, sk).
- $\mathsf{Enc}(pk, b, \mu)$: output $y \leftarrow \mathsf{LWEEnc}((\mathbf{A}, \mathbf{p} = pk + \mathbf{v}_b), \mu)$, where $b \in \{0,1\}$ is chosen by the encrypter. Here y is the pair $(\mathbf{u}, c)$.
- $\mathsf{Dec}(sk, y)$: output $\mu \leftarrow \mathsf{LWEDec}(sk = \mathbf{s}, (\mathbf{u}, c))$.
- $\mathsf{FindMessy}(\tau, pk)$: parse τ as $(\mathbf{S}, \mathbf{A})$, run $\mathsf{IsMessy}(\mathbf{S}, \mathbf{A}, pk + \mathbf{v}_b)$ for each $b \in \{0,1\}$, and output b such that $\mathsf{IsMessy}$ can output "messy" on at least one branch correctly with overwhelming probability.
- $\mathsf{TrapdoorKeyGen}(\tau)$: parse τ as $(\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$, and output (pk, sk_0, sk_1)=$(\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$.

[2]Referring to [25], the particular value of the Gaussian parameter r is a parameter of the scheme. When the trapdoor information is embed in the scheme, the value of r will be related to the length of the trapdoor for $\mathbf{A}$.

[3]The ciphertext of a message can be produced in two branches $b = 0$ or $b = 1$. This ciphertext only on the branch $\sigma = b$ can be correctly decrypted. We call the branch σ decryptable.

Remark 3.2. SetupMessy *generates a* crs *together with trapdoor* $\tau = (\mathbf{S}, \mathbf{A})$, *where* $\mathbf{S}$ *is an embedded trapdoor in* $\mathbf{A}$ *so that* IsMessy *can be used in* FindMessy *to reveal that whether* $pk + \mathbf{v}_b$ *is a messy key for each* $b \in \{0,1\}$ *when the trapdoor* τ *is given.*

The dual-mode cryptosystem has three main security properties: (1) in messy mode, for each base public key pk, at least one of the derived public keys $pk + \mathbf{v}_b$ 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 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, the instantiation of the dual-mode cryptosystem with the LWE problem only satisfies a slightly relaxed version of the above security properties. Fortunately, a UC-secure OT protocol based on LWE hardness can still be derived, however, it leads to a slightly weaker security of the honest receiver when running in decryption mode, i.e., *computational* security.

3.2. **LWE-based OT protocol.** Once this LWE-based dual-mode cryptosystem is built well, an LWE-based OT protocol denoted by $\mathsf{dm}^{\mathsf{mode}}$ can be obtained directly (see Figure 1). This $\mathsf{dm}^{\mathsf{mode}}$ can securely UC-emulate a multi-session OT functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ in the $\mathcal{F}_{\mathrm{CRS}}$-hybrid model (cf. [8]), where $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ serves as a shell around a bounded number (corresponding to ℓ) of independent $\mathcal{F}_{\mathrm{OT}}$ executions. Here $\mathcal{F}_{\mathrm{OT}}$ denote the traditional OT ideal functionality, where the sender $\mathbf{S}$ and the receiver $\mathbf{R}$ just forward their inputs (m_0, m_1) and $\sigma \in \{0,1\}$ to $\mathcal{F}_{\mathrm{OT}}$ and $\mathcal{F}_{\mathrm{OT}}$ will return m_σ to $\mathbf{R}$ without any output to $\mathbf{S}$. Here $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ specifies its interaction with two parties in a single session by sid, and coordinates each subsession of the same session by ssid. $\mathcal{F}_{\mathrm{CRS}}$ is an ideal functionality which generates a crs for both parties. Once crs is obtained, the execution of $\mathsf{dm}^{\mathsf{mode}}$ mainly follows the procedure of the dual-mode cryptosystem. Note that $\mathsf{dm}^{\mathsf{mode}}$ can also operate in two modes. Since $\mathcal{F}_{\mathrm{CRS}}$ can run two different setup algorithm, we denote it by $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$. When $\mathcal{D} =$ SetupMessy, $\mathsf{dm}^{\mathsf{mode}}$ runs in the messy mode; When $\mathcal{D} =$ SetupDec,$\mathsf{dm}^{\mathsf{mode}}$ runs in the decryption mode. Note that the protocol $\mathsf{dm}^{\mathsf{mode}}$ is a 1-out-of-2 OT, it can be extended as a 1-out-of-2^k OT protocol by choosing $\mathbf{v}_b$ for each $b \in \{0,1\}^k$.

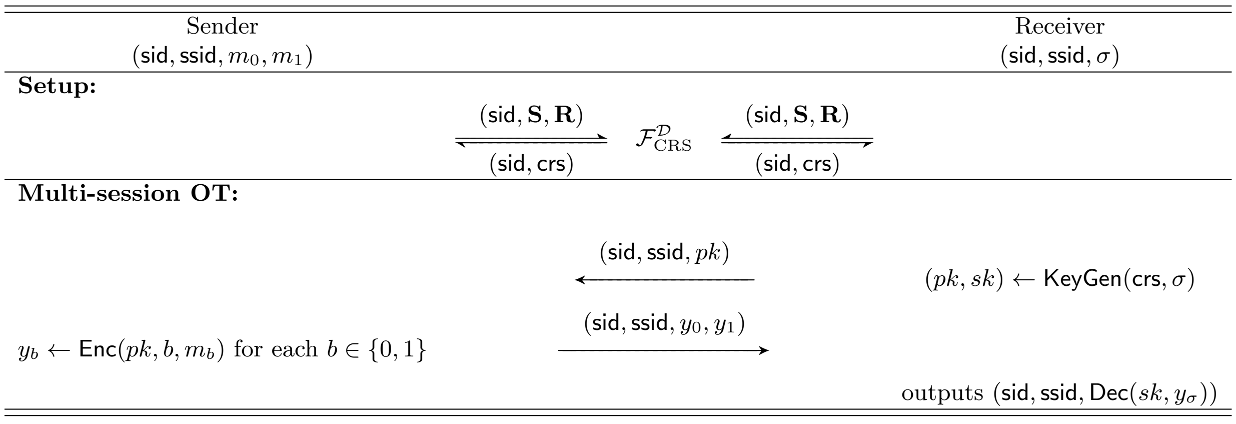


FIGURE 1. Lattice-based OT protocol $\mathsf{dm}^{\mathsf{mode}}$

3.3. **UC-security proof sketch.** As shown in [14], $\mathsf{dm}^{\mathsf{mode}}$ is provable secure against *static* corruptions in the UC framework, where static corruption means that the adversary can only make corruption decision before the execution of the protocol instead of during the course of the protocol execution. UC-security implies that for any *real-world* adversary

Adv, there exists an *ideal-world* adversary Sim, called *simulator*, interacting with the ideal functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$, such that no machine Env can distinguish its interaction with Adv in an execution of $\mathsf{dm}^{\mathsf{mode}}$ from an interaction with Sim by using $\widetilde{\mathcal{F}}_{\mathrm{OT}}$. The main ingredients of UC-security proof of $\mathsf{dm}^{\mathsf{mode}}$ lie in the simulator constructions in the corruption cases that when only the receiver is corrupted and when only the sender is corrupted, respectively.

In each corruption case, Sim starts by running a copy of Adv. Every incoming value received by Sim is written into Adv's input tape. Every outgoing value on Adv's output tape is copied to Sim's output tape. Note that regardless of which mode the protocol $\mathsf{dm}^{\mathsf{mode}}$ will run in, the construction of Sim only depends on corruption case.

When only R is corrupted: Sim runs the messy mode setup algorithm and generates $(\mathsf{crs}, \tau) \leftarrow \mathsf{SetupMessy}(1^n)$. Namely, Sim chooses a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ uniformly at random, together with a trapdoor $\tau = (\mathbf{S}, \mathbf{A})$. For each $b \in \{0,1\}$, Sim selects a vector $\mathbf{v}_b \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random, sets $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$, and outputs (crs, τ). When the parties query $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$, Sim returns $(\mathsf{sid}, \mathsf{crs})$ to them and stores τ privately. At some point, Adv produces a message $(\mathsf{sid}, \mathsf{ssid}, pk)$ and sends it to Sim, where $pk = \mathbf{s}^T\mathbf{A} + \mathbf{x} - \mathbf{v}_\sigma$. Then Sim runs $\mathsf{FindMessy}(\mathsf{crs}, \tau, pk)$ to find b for specifying the messy branch, and queries the ideal functionality $\widetilde{\mathcal{F}}_{OT}$ with $(\mathsf{sid}, \mathsf{ssid}, \mathsf{receiver}, 1-b)$ in the name of $\mathbf{R}$. Then Sim receives the output $(\mathsf{sid}, \mathsf{ssid}, m_{1-b})$ from $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ and stores the value (b, m_{1-b}). When $\mathbf{S}$ is activated on some subsession $(\mathsf{sid}, \mathsf{ssid})$, then Sim must play the role of the sender to interact with Adv like a real world execution. Sim firstly looks up the corresponding (b, m_{1-b}), then computes $y_{1-b} \leftarrow \mathsf{Enc}(pk, 1-b, m_{1-b})$ and $y_b \leftarrow \mathsf{Enc}(pk, b, 0^\ell)$. Finally Sim sends the message $(\mathsf{sid}, \mathsf{ssid}, y_0, y_1)$ to Adv as if it were from $\mathbf{S}$.

When only S is corrupted: Sim runs the decryption mode setup algorithm and generates $(\mathsf{crs}, \tau) \leftarrow \mathsf{SetupDec}(1^n)$, Namely, Sim chooses a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random. For each $b \in \{0,1\}$, Sim chooses a secret vector $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ uniformly at random and an error vector $\mathbf{x}_b \leftarrow \mathcal{X}^{1\times m}$, and then sets $\mathbf{v}_b = \mathbf{s}_b^T\mathbf{A} + \mathbf{x}_b - \mathbf{w}$, $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$, and $\tau = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$. When the parties query the ideal functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$, Sim returns $(\mathsf{sid}, \mathsf{crs})$ to them and stores τ privately. When $\mathbf{R}$ is activated on some subsession $(\mathsf{sid}, \mathsf{ssid})$, Sim runs $\mathsf{TrapKeyGen}(\mathsf{crs}, \tau)$ to generate (pk, sk_0, sk_1), where $(pk, sk_0, sk_1) = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$, and sends $(\mathsf{sid}, \mathsf{ssid}, pk)$ to Adv as if it were from $\mathbf{R}$, and s-tores $(\mathsf{sid}, \mathsf{ssid}, pk, sk_0, sk_1)$. When Adv replies with a tuple $(\mathsf{sid}, \mathsf{ssid}, y_0, y_1)$, Sim first looks up the corresponding (pk, sk_0, sk_1), then computes $m_b \leftarrow \mathsf{Dec}(sk_b, y_b)$ for each $b \in \{0,1\}$. Since $\mathbf{S}$ has been activated, Sim sends $(\mathsf{sid}, \mathsf{ssid}, \mathsf{sender}, m_0, m_1)$ to $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ as if it were from $\mathbf{S}$.

4. Equational Security of $\mathsf{dm}^{\mathsf{mode}}$.

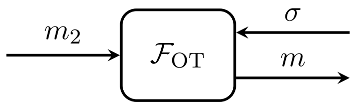
In this section, we analyze the equational security of $\mathsf{dm}^{\mathsf{mode}}$. As shown in Section 3, it is clear to see that in the UC-security proof of $\mathsf{dm}^{\mathsf{mode}}$ they use the traditional OT functionality (see Figure 2). When we keep using this traditional OT functionality in equational security framework, we found that there exists a timing bug in the case that when the sender is corrupted. Then we somehow modify the OT functionality into a revised one that outputs the sender one acknowledge bit a, which illustrates whether the chosen bit σ of the receiver has been transmitted yet or not without leaking any content of σ. However, we cannot achieve the security of the sender in this case. Regarding to this situation, we weaken this revised OT functionality by allowing to leak the bit a to the environment, but the result is still negative.

4.1. Equational description of $\mathsf{dm}^{\mathsf{mode}}$.

In this section, we first give the description of $\mathsf{dm}^{\mathsf{mode}}$ in the equational security framework. From Section 3, we know that $\mathsf{dm}^{\mathsf{mode}}$

can securely UC-emulate a multi-session OT functionality $\widetilde{\mathcal{F}}_{\mathrm{OT}}$ which can be regarded as a shell that wraps several traditional OT functionalities $\mathcal{F}_{\mathrm{OT}}$. In each subsession ssid of a session sid, one bit can be securely and obliviously transferred to the receiver $\mathbf{R}$. This feature of $\mathsf{dm}^{\mathsf{mode}}$ comes from the fact that the LWE-based encryption utilized for constructing the LWE-based dual-mode cryptosystem can encrypt an ℓ-bit message by reusing the public matrix $\mathbf{A}$ for ℓ times bit encryption. When $\ell = 1$, $\mathsf{dm}^{\mathsf{mode}}$ is a 1-out-of-2 OT protocol which UC-emulates a single-session OT functionality which securely and obliviously transfer one single bit, i.e., $\widetilde{\mathcal{F}}_{\mathrm{OT}} = \mathcal{F}_{\mathrm{OT}}$. For simplicity, we only consider the case that when $\ell = 1$, that is, we only use sid and omit ssid in the equational description of $\mathsf{dm}^{\mathsf{mode}}$.

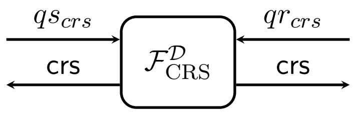
In the equational security framework, this traditional OT functionality $\mathcal{F}_{\mathrm{OT}}$ can be described as shown in Figure 2, where the sender $\mathbf{S}$ and the receiver $\mathbf{R}$ send their inputs $m_2 \in \mathbb{M}_\ell^{\times 2} = \mathbb{Z}_2^{\times 2}$ and $\sigma \in \mathbb{B} = \{0,1\}$ to $\mathcal{F}_{\mathrm{OT}}$, respectively. Then $\mathcal{F}_{\mathrm{OT}}$ will output $m = m_2[\sigma]$ to $\mathbf{R}$ without any output to $\mathbf{S}$.



$$\begin{aligned}
\mathsf{OT}(m_2,\sigma) &= m\\
m_2 &: \mathbb{Z}_2^{\times 2}\\
\sigma &: \mathbb{B}\\
m &: \mathbb{Z}_2\\
m &= m_2[\sigma]
\end{aligned}$$

FIGURE 2. Traditional OT functionality $\mathcal{F}_{\mathrm{OT}}$

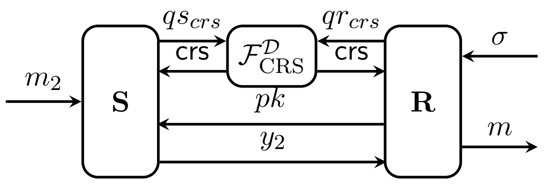
In addition, we know that $\mathsf{dm}^{\mathsf{mode}}$ can operate in two modes. In each mode, two parties first query $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$ with qs_{crs} and qr_{crs} (see Figure 3) for a crs. Here the distribution of crs is decided by setup algorithm in each mode, i.e., when running in messy mode (i.e., $\mathcal{D} = \mathsf{SetupMessy}(1^n)$), $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}} = \mathcal{F}_{\mathrm{CRS}}^{\mathsf{mes}}$; When running in decryption mode (i.e., $\mathcal{D} = \mathsf{SetupDec}(1^n)$), $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}} = \mathcal{F}_{\mathrm{CRS}}^{\mathsf{dec}}$.



$$\begin{aligned}
\mathsf{CRS}_{\mathcal{D}}(qs_{crs}, qr_{crs}) &= (\mathsf{crs},\mathsf{crs})\\
qs_{crs} &= (\mathsf{sid},\mathbf{S},\mathbf{R})\\
qr_{crs} &= (\mathsf{sid},\mathbf{S},\mathbf{R})\\
\mathsf{crs} &\leftarrow \mathsf{SetupMessy}(1^n)/\mathsf{SetupDec}(1^n)
\end{aligned}$$

FIGURE 3. CRS functionality $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$

As shown in Figure 4, $\mathsf{dm}^{\mathsf{mode}}$ is described by the notion of nodes and channels, where $\mathbf{S}$, $\mathbf{R}$ and $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$ are represented by three nodes with several input and output channels. It is easy to show the correctness of the protocol. With combining the equations of $\mathcal{F}_{\mathrm{CRS}}^{\mathcal{D}}$, Sender and $\mathsf{Receiver}$ programs form a real system $\mathsf{Real}(m_2,\sigma) = m$ which is perfectly equivalent to the defining equation $m = m_2[\sigma]$ of the ideal functionality $\mathcal{F}_{\mathrm{OT}}$, while $\mathcal{F}_{\mathrm{OT}}$ implies an ideal system $\mathsf{Ideal}(m_2,\sigma) = m_2[\sigma]$.



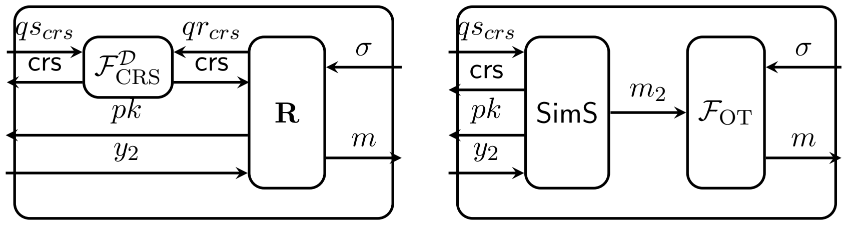
$$
\begin{aligned}
\mathsf{Sender}(m_2, \mathsf{crs}, pk) &= (qs_{crs}, y_2)\\
qs_{crs} &= (\mathsf{sid}, \mathbf{S}, \mathbf{R})\\
y_0 &\leftarrow \mathsf{LWEEnc}((\mathbf{A}, \mathbf{p} = pk + \mathbf{v}_0), m_2[0])\\
y_1 &\leftarrow \mathsf{LWEEnc}((\mathbf{A}, \mathbf{p} = pk + \mathbf{v}_1), m_2[1])\\
y_2 &= \langle y_0, y_1 \rangle\\
\mathsf{Receiver}(\mathsf{crs}, y_2, \sigma) &= (qr_{crs}, pk, m)\\
qr_{crs} &= (\mathsf{sid}, \mathbf{S}, \mathbf{R})\\
pk &\leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)\\
m &= \mathsf{LWEDec}(sk = \mathbf{s}, y_2[\sigma]).
\end{aligned}
$$

FIGURE 4. Equational description of $\mathsf{dm}^{\mathsf{mode}}$

4.2. **Equational security regarding to the traditional OT functionality $\mathcal{F}_{\mathbf{OT}}$.** Now we analyze the security of $\mathsf{dm}^{\mathsf{mode}}$ in the equational framework. The security of $\mathsf{dm}^{\mathsf{mode}}$ in the UC framework requires that for any adversary Adv, there always exists an efficient simulator program Sim such that for any Env, it cannot distinguish the real system (the execution of $\mathsf{dm}^{\mathsf{mode}}$ with Adv) from the ideal system (the execution of $\mathcal{F}_{\mathrm{OT}}$ with Sim) with overwhelming advantage. Here this environment Env connects to all input and output channels of the execution system. In addition, we let Env produce one additional output $t \in \mathbb{S}$. The distinguishing advantage of the environment Env is defined by

$$\mathsf{ADV}(\mathsf{Env}) = |\Pr\{\mathsf{Env}[\mathsf{Real}] = \top\} - \Pr\{\mathsf{Env}[\mathsf{Ideal}] = \top\}|.$$

When the sender is corrupted. In this attack scenario, since the sender is corrupted by the adversary, the real system is obtained by removing the Sender program from Figure 4. As shown in Figure 5(a), the real system is represented by $\mathsf{RealS}(qs_{crs}, y_2, \sigma) = (\mathsf{crs}, pk, m)$. The corresponding ideal system $\mathsf{IdealS}(qs_{crs}, y_2, \sigma) = (\mathsf{crs}, pk, m)$ is described as shown in Figure 5(b). The security in this case requires that two execution systems are indistinguishable for any Env, where Env connects all input and output channels of these two systems, it is denoted by $\mathsf{Env}(\mathsf{crs}, pk, m) = (qs_{crs}, y_2, \sigma, t)$.



(a) The real system (b) The ideal system

FIGURE 5. When the sender is corrupted (with respect to $\mathcal{F}_{\mathrm{OT}}$)

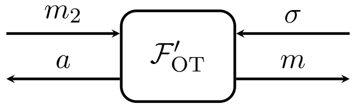
Now we define two distinguishing environments Env_0 and Env_1, and show that for any simulator SimS, at least one of these two environments can distinguish the real and ideal systems with non-negligible advantage.

- $\mathsf{Env}_0(\mathsf{crs}, pk, m) = (qs_{crs}, y_2, \sigma, t)$ sets $qs_{crs} = \top$, $y_2 = \perp$, and $\sigma = \perp$, and outputs $t = (pk > \perp)$.
- $\mathsf{Env}_1(\mathsf{crs}, pk, m) = (qs_{crs}, y_2, \sigma, t)$ sets $qs_{crs} = \top$, $y_2 = \perp$, and $\sigma \in \{0, 1\}$, and outputs $t = (pk > \perp)$.

The only difference between Env_0 and Env_1 is the value of σ. According to the receiver program $\mathsf{Receiver}$, we can see that $pk > \perp$ if and only if $\sigma > \perp$ in the real system. Thus, we have $\Pr\{\mathsf{Env}_0[\mathsf{RealS}] = \top\} = 0$ and $\Pr\{\mathsf{Env}_1[\mathsf{RealS}] = \top\} = 1$. In the ideal system, since the output value t is independent of σ when interacting with IdealS, we have $\Pr\{\mathsf{Env}_0[\mathsf{IdealS}] = \top\} = \Pr\{\mathsf{Env}_1[\mathsf{IdealS}] = \top\}$, which is denoted by p. Thus, these two environments have advantages $\mathsf{ADV}(\mathsf{Env}_0) = p$ and $\mathsf{ADV}(\mathsf{Env}_1) = 1 - p$, respectively. Therefore, either Env_0 or Env_1 has distinguishing advantage at least $1/2$.

4.3. **Equational security regarding to a revised OT functionality $\mathcal{F}'_{\mathbf{OT}}$.**

Based on the above analysis, we know that the protocol $\mathsf{dm}^{\mathsf{mode}}$ is not secure when the sender is corrupted since the environment can distinguish by setting $\sigma > \perp$ and observing $pk > \perp$. This is the only weakness, i.e., a timing bug, in the current corruption. With this concern, we can modify the traditional OT functionality $\mathcal{F}_{\mathrm{OT}}$ into a revised OT functionality which will additionally output a signal bit $a = (\sigma > \perp)$ to the sender. The signal $a \in \mathbb{S}$ only leaks the information that whether σ has been transmitted yet or not without revealing the content of σ. We denote this revised OT functionality as $\mathcal{F}'_{\mathrm{OT}}$, see Figure 6.



$$\begin{aligned}\mathsf{OT}'(m_2, \sigma) &= (a, m)\\ m &= m_2[\sigma]\\ a &= (\sigma > \perp)\end{aligned}$$

FIGURE 6. A revised OT functionality $\mathcal{F}'_{\mathrm{OT}}$

When the sender is corrupted. Due to this revised OT functionality, we accordingly modify the OT protocol $\mathsf{dm}^{\mathsf{mode}}$. Therefore, when the sender is corrupted, the real and ideal execution systems are described as shown in Figure 7.

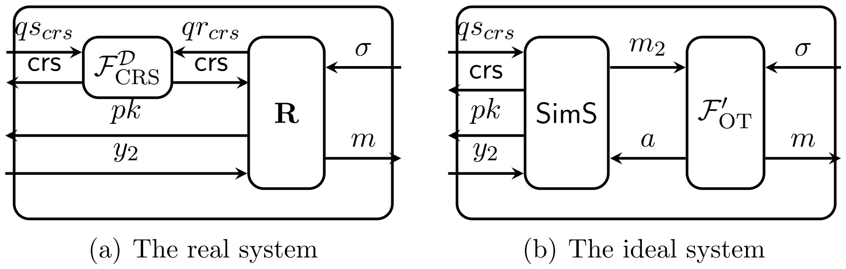
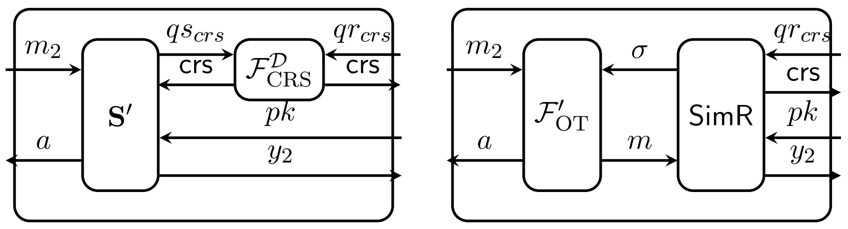


FIGURE 7. When the sender is corrupted (with respect to $\mathcal{F}'_{\mathrm{OT}}$)

Here the security against corrupted sender with respect to $\mathcal{F}'_{\mathrm{OT}}$ can be proved by the following simulator SimS, which takes the signal $a = (\sigma > \perp)$ as an additional input.

$$\begin{array}{rcl}
\mathsf{SimS}(qs_{crs}, y_2, a) & = & (\mathsf{crs}, pk, m_2) \\
(\mathsf{crs}, \tau) & \leftarrow & \mathsf{SetupDec}(1^n) \\
\mathsf{crs} & = & (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1) \\
\tau & = & (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1) \\
pk & \leftarrow & a!\mathbf{w} \\
qs_{crs} & = & (\mathsf{sid}, \mathbf{S}, \mathbf{R}) \\
m_2[0] & = & \mathsf{LWEDec}(sk = \mathbf{s}_0, y_2[0]) \\
m_2[1] & = & \mathsf{LWEDec}(sk = \mathbf{s}_1, y_2[1]) \\
m_2 & = & \langle m_2[0], m_2[1] \rangle
\end{array}$$

When the receiver is corrupted. Now we have to consider the security of $\mathsf{dm}^{\mathsf{mode}}$ when the receiver is corrupted. However, in order to match the case that $\mathcal{F}'_{\mathrm{OT}}$ leaks a signal a to the sender, we need to modify the sender program in the real system. The only possible way for this goal is that we let the sender produce an additional output $a = (pk > \perp)$ since the sender only receives one message pk which is the dependent of σ from the receiver. Therefore, the current real and ideal systems are shown in Figure 8.



(a) The real system (b) The ideal system

FIGURE 8. When the receiver is corrupted (with respect to $\mathcal{F}'_{\mathrm{OT}}$)

Note that the sender program is accordingly modified as Sender', which is represented by $\mathbf{S}'$ in Figure 8 and described as the following equations:

$$\begin{array}{rcl}
\mathsf{Sender}'(m_2, \mathsf{crs}, pk) & = & (a, qs_{crs}, y_2) \\
(qs_{crs}, y_2) & = & \mathsf{Sender}(m_2, \mathsf{crs}, pk) \\
a & = & (pk > \perp).
\end{array}$$

It is easy to verify the correctness of the revised OT protocol, where a real protocol execution $(\mathsf{Sender}'|\mathcal{F}^{\mathcal{D}}_{\mathrm{CRS}}|\mathsf{Receiver}) : (m_2, \sigma) \to (a, m)$ is equivalent to the ideal functionality $\mathcal{F}'_{\mathrm{OT}} : (m_2, \sigma) \to (a, m)$. Now we only need to show that the real and ideal systems described in Figure 8 are indistinguishable for any $\mathsf{Env}(a, \mathsf{crs}, y_2) = (m_2, qr_{crs}, pk)$, i.e., the security against the corrupted receiver. However, the result is negative and shown by Proposition 4.1.

Proposition 4.1. *For any simulator* SimR*, there is an environment* $\mathsf{Env}(a, \mathsf{crs}, y_2) = (m, qr_{crs}, pk, t)$ *such that the distinguishing advantage* $\mathsf{ADV}[\mathsf{Env}] = |\Pr\{\mathsf{Env}[\mathsf{RealR}] = \top\} - \Pr\{\mathsf{Env}[\mathsf{IdealR}] = \top\}|$ *is not negligible.*

Proof: We define two distinguishing environments Env_1 and Env_2, and show that for any simulator SimR, at least one of these two environments can distinguish the real system from ideal systems with non-negligible advantage.

- $\mathsf{Env}_1(a, \mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ sets $qr_{crs} = \top$, $m_2 = \perp$, $\sigma \leftarrow \{0, 1\}$, and $pk > \perp$, and outputs $t = (a > \perp)$.
- $\mathsf{Env}_2(a, \mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ sets $qr_{crs} = \top$, $m_2 \leftarrow \mathbb{Z}_2^{\times 2}$, $\sigma \leftarrow \{0, 1\}$, and $pk > \perp$, and outputs $t = (m_2[\sigma] = \mathsf{LWEDec}(sk = \mathbf{s}, y_2[\sigma]))$.

We let u_i denote the input distribution of SimR (i.e., $\mathsf{SimR}(m, qr_{crs}, pk) = (\sigma, \mathsf{crs}, y_2)$) when interacting with Env_i for $i = 1, 2$. For any input distribution $u = (m, qr_{crs}, pk)$, u^m denotes the distribution of m. Similarly, we have $u^{qr_{crs}}$, u^{pk}. In addition, we denote the distribution of σ as $\mathsf{SimR}(u)^\sigma$.

When interacting with Env_1, the real system sets $a = \top$. For matching the case that the ideal system is indistinguishable with the real system, SimR must output $\sigma \in \{0, 1\}$ with overwhelming probability. Let $p_1 = \Pr(\mathsf{SimR}(u_1)^\sigma > \bot)$ and $p_2 = \Pr(\mathsf{SimR}(u_2)^\sigma > \bot)$. Then $1 - p_1$ is negligible.

Note that $u_1^m = \bot$ with probability 1 since $m_2 = \bot$. Env_1 and Env_2 both set pk based on the same distribution, so $u_1 \leq u_2$. Since SimR is monotone, we have $p_1 \leq p_2$, and thus $1 - p_2$ is also negligible.

For $i = 1, 2$, we let $p_i^0 = \Pr(\mathsf{SimR}(u_i)^\sigma = 0)$ and $p_i^1 = \Pr(\mathsf{SimR}(u_i)^\sigma = 1)$, where $p_1^0 + p_1^1 = p_1$ and $p_2^0 + p_2^1 = p_2$. Since both $\{0\}$ and $\{1\}$ are open sets in $\mathbb{B}$, by monotonicity of SimR we have $p_1^0 \leq p_2^0$ and $p_1^1 \leq p_2^1$.

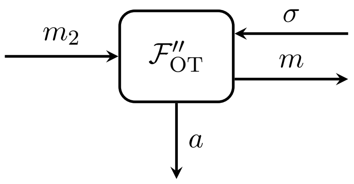
Now we consider the distinguishing advantage of Env_2. It is easy to see that $\Pr\{\mathsf{Env}_2[\mathsf{RealR}] = \top\} = 1$. When interacting with the ideal system, let $\tilde{\sigma}$ denote $\mathsf{SimR}(u_2)^\sigma$, then we have $\Pr(\mathsf{Env}_2[\mathsf{IdealR} = \top]) = \Pr(t = \top | \tilde{\sigma} = \bot)\Pr(\tilde{\sigma} = \bot) + \Pr(t = \top | \tilde{\sigma} = \sigma)\Pr(\tilde{\sigma} = \sigma) + \Pr(t = \top | \tilde{\sigma} = 1 - \sigma)\Pr(\tilde{\sigma} = 1 - \sigma)$ where $t = (m_2[\sigma] = \mathsf{LWEDec}(s, y_2[\sigma]))$.

We can deduce that $p_1^{1-\sigma}$ is close to $1/2$. For this claim, we fix σ and consider the case when $\tilde{\sigma} = 1 - \sigma$. By definition of $\mathcal{F}'_{\mathrm{OT}}$, $u_1^m = m_2[\tilde{\sigma}]$, and SimR cannot learn $m_2[\sigma]$. In addition, since $m_2[\sigma]$ is randomly selected from $\mathbb{Z}_2$, SimR cannot do better than randomly guessing and outputting a ciphertext $c_2[d]$ of it. Therefore, we have $\Pr(t = \top | \tilde{\sigma} = 1 - \sigma) \leq 1/2$. Since $p_1^{1-\sigma} \leq p_2^{1-\sigma} = p_2 - p_2^\sigma$, we have $p_2^\sigma \leq p_2 - p_1^{1-\sigma}$. Therefore, we can deduce the following inequations:

$$\begin{aligned}
&\Pr(\mathsf{Env}_2[\mathsf{IdealR} = \top]) \\
&= \Pr(t = \top | \tilde{\sigma} = \bot)\Pr(\tilde{\sigma} = \bot) + \Pr(t = \top | \tilde{\sigma} = \sigma)\Pr(\tilde{\sigma} = \sigma) \\
&\quad + \Pr(t = \top | \tilde{\sigma} = 1 - \sigma)\Pr(\tilde{\sigma} = 1 - \sigma) \\
&\leq (1 - p_2) + Pr(t = \top | \tilde{\sigma} = \sigma)Pr(\tilde{\sigma} = \sigma) + 1/2 \\
&\leq (1 - p_2) + p_2^\sigma + 1/2 \\
&= 1 - (p_2 - p_2^\sigma) + 1/2 \\
&\leq 1 - p_1^{1-\sigma} + 1/2
\end{aligned}$$

If we require $|\Pr(\mathsf{Env}_2[\mathsf{RealR}] = \top) - \Pr(\mathsf{Env}_2[\mathsf{IdealR}] = \top)| = |1 - \Pr(\mathsf{Env}_2[\mathsf{IdealR}] = \top)| \leq |p_1^{1-\sigma} - 1/2|$ to be negligible, then $p_1^{1-\sigma}$ should be close to $1/2$. This leads to a contradiction in Env_1, i.e., SimR cannot output $\tilde{\sigma} = \sigma$ with overwhelming probability. This fact contradicts the nature of $\mathsf{FindMessy}$ which can find correct messy branch $1 - \sigma$ on the information of (crs, τ) with overwhelming probability. Therefore, it implies that such a SimR is insufficient in Env_2 such that Env_2 can only distinguish between the real and ideal systems with negligible advantage, or our assumption that Env_1 can distinguish IdealR from RealR with negligible advantage is not the case.

4.4. **Equational security regarding to a weaker OT functionality $\mathcal{F}''_{\mathbf{OT}}$.** From the above analysis, we know that when using traditional OT functionality $\mathcal{F}_{\mathrm{OT}}$ the security under the corruption with the sender cannot be achieved. This is mainly due to the lack of information a. Then we revised $\mathcal{F}_{\mathrm{OT}}$ as $\mathcal{F}'_{\mathrm{OT}}$ which leaks a to the sender. However, in this case, the security under the corruption with the receiver cannot be achieved. It implies that we cannot build good simulators satisfying the security of $\mathsf{dm}^{\mathsf{mode}}$ when using neither $\mathcal{F}_{\mathrm{OT}}$ nor $\mathcal{F}'_{\mathrm{OT}}$, mainly because that it is hard to emulate (either directly or indirectly) the signal bit a for simulator. Now we would like to see what if we modify $\mathcal{F}'_{\text{OT}}$ into a weaker functionality $\mathcal{F}''_{\text{OT}}$ (see Figure 9), where the acknowledge bit $a = (\sigma > \perp)$ is leaked to the environment Env instead of seen by the sender in $\mathcal{F}'_{\text{OT}}$.



$$\begin{aligned} \mathsf{OT}''(m_2, \sigma) &= (a, m) \\ m &= m_2[\sigma] \\ a &= \sigma! \end{aligned}$$

FIGURE 9. A weaker OT functionality $\mathcal{F}''_{\text{OT}}$

Here we say $\mathcal{F}''_{\text{OT}}$ is weaker than $\mathcal{F}'_{\text{OT}}$ since $\mathcal{F}'_{\text{OT}}$ can emulate $\mathcal{F}''_{\text{OT}}$ by composing with an "ideal agent" which relays m_2 to $\mathcal{F}'_{\text{OT}}$ but releases a to Env, while $\mathcal{F}''_{\text{OT}}$ is unable to emulate $\mathcal{F}'_{\text{OT}}$ (a is given to Env, so the sender wont get it). When the ideal functionality leaks some information to Env, this usually means that in the real world, the party responsible for outputting that information is controlled by an adversary, and in the ideal world that information is given to a simulator to emulate the adversary's behavior. For the concrete case of $\mathcal{F}''_{\text{OT}}$, when the sender/receiver is corrupted, a is given to $\mathsf{SimS}/\mathsf{SimR}$, which may want to utilize a to decide when or what other information would be released to $\mathcal{F}''_{\text{OT}}$ and the environment Env. However, it is not necessary for $\mathsf{SimS}/\mathsf{SimR}$ to emulate the behaviour of a corrupted sender/receiver by taking advantage of this additional input.

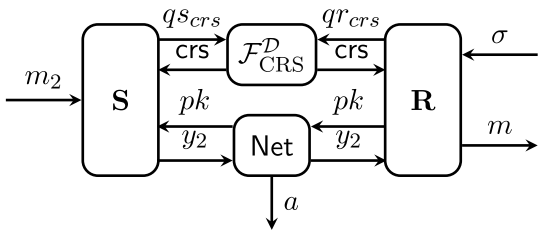
By the definition of $\mathcal{F}''_{\text{OT}}$, we accordingly modify the protocol $\mathsf{dm}^{\mathsf{mode}}$ (see Figure.10). Here we add an additional functionality Net between Sender and $\mathsf{Receiver}$, which is responsible outputting a to the environment.

Now we analyse the security of $\mathsf{dm}^{\mathsf{mode}}$ with respect to $\mathcal{F}''_{\text{OT}}$. Note that the program Net can be regarded as a part of the adversary, once one party is corrupted, we remove the corresponding corrupted program together with Net functionality from the real system.

When the sender is corrupted. When the sender is corrupted, we remove Sender program and Net from Figure 10, then the real system is $(\mathcal{F}^{\mathcal{D}}_{\text{CRS}}|\mathsf{Receiver})$. We can see that in the real system there is no interface of a any more, while in the ideal world, a is given to SimS. It is clear to see that the current protocol description is the same as the case shown in Figure 7, so the security can be achieved directly when the sender is corrupted.

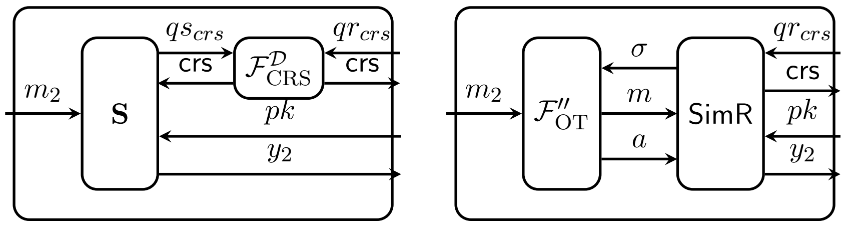
When the receiver is corrupted. When the receiver is corrupted, we remove $\mathsf{Receiver}$ program and Net from Figure 10, then the real system is $(\mathsf{Sender}|\mathcal{F}^{\mathcal{D}}_{\text{CRS}})$. The real and ideal systems are described as shown in Figure 11. Unfortunately, even though we already make use of a weaker OT functionality $\mathcal{F}''_{\text{OT}}$, the security under the corruption with the receiver still cannot be achieved. This is shown by the following Proposition 4.2.

Proposition 4.2. *For any simulator* SimR*, there is an environment* $\mathsf{Env}(\mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ *such that the distinguishing advantage* $\mathsf{ADV}[\mathsf{Env}] = |\Pr\{\mathsf{Env}[\mathsf{RealR}] = \top\} - \Pr\{\mathsf{Env}[\mathsf{RealR}] = \top\}|$ *is not negligible.*



$$
\begin{array}{rcl}
\mathsf{Sender}(m_2, \mathsf{crs}, pk) & = & (qs_{crs}, y_2) \\
qs_{crs} & = & (\mathsf{sid}, \mathbf{S}, \mathbf{R}) \\
y_0 & \leftarrow & \mathsf{LWEEnc}((\mathbf{A}, \mathbf{p} = pk + \mathbf{v}_0), m_2[0]) \\
y_1 & \leftarrow & \mathsf{LWEEnc}((\mathbf{A}, \mathbf{p} = pk + \mathbf{v}_1), m_2[1]) \\
y_2 & = & \langle y_0, y_1 \rangle \\
\mathsf{Receiver}(\mathsf{crs}, y_2, \sigma) & = & (qr_{crs}, pk, m) \\
qr_{crs} & = & (\mathsf{sid}, \mathbf{S}, \mathbf{R}) \\
pk & \leftarrow & \mathsf{KeyGen}(\mathsf{crs}, \sigma) \\
m & = & \mathsf{LWEDec}(sk = \mathbf{s}, y_2[\sigma]). \\
\mathsf{Net}(pk, y_2) & = & (pk, y_2, a) \\
a & = & pk!
\end{array}
$$

FIGURE 10. Equational description of modified $\mathsf{dm}^{\mathsf{mode}}$



(a) The real system (b) The ideal system

FIGURE 11. When the receiver is corrupted (with respect to $\mathcal{F}''_{\text{OT}}$)

Proof: We define three environments in the following and show that at least one of these environments can distinguish the real system from the ideal system with non-negligible advantage:

- $\mathsf{Env}_1(\mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ sets $qr_{crs} = \top$, $m_2 = \bot$, $\sigma \leftarrow \{0,1\}$, and $pk > \bot$, and outputs $t = (y_2 > \bot)$.
- $\mathsf{Env}_2(\mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ sets $qr_{crs} = \top$, $m_2 \leftarrow \mathbb{Z}_2^{\times 2}$, $\sigma \leftarrow \{0,1\}$, and $pk > \bot$, and outputs $t = (y_2 > \bot)$.
- $\mathsf{Env}_3(\mathsf{crs}, y_2) = (m_2, qr_{crs}, pk, t)$ sets $qr_{crs} = \top$, $m_2 \leftarrow \mathbb{Z}_2^{\times 2}$, $\sigma \leftarrow \{0,1\}$, and $pk > \bot$, and outputs $t = (m_2[\sigma] = \mathsf{LWEDec}(sk = \mathbf{s}, y_2[\sigma]))$.

We note that Env_3 is functionally equivalent to Env_2 defined in the proof of Proposition 4.1, where Env_2 in Proposition 4.1 does not make use of the input a.

Let p_i represent the probability that Env_i outputs $t = \top$ when interacting with IdealR, for $i = 1, 2, 3$. When interacting with Env_1, the Sender program sets $y_2 = \bot$, so p_1 must be negligible; When interacting with Env_2, the Sender program sets $y_2 = \bot$, so $1 - p_2$ must be negligible. Any simulator SimR in IdealR takes the tuple (m, a, qr_{crs}, pk) as input, and $(\sigma, \mathsf{crs}, y_2)$ as output. Let u_i denote the input distribution of SimR when interacting with Env_i for $i = 1, 2, 3$. It is clear that $u_1 \leq u_2 \leq u_3$.

Let p_σ denote the probability that SimR cannot output σ in Env_2, i.e., $\Pr(\mathsf{SimR}(u_2)^\sigma = \bot)$. We can see that both Env_1 and Env_2 set pk based on the same distribution. If $\mathsf{SimR}(u_2)^\sigma = \bot$, we have $\mathsf{SimR}(u_1)^\sigma = \bot$ by the monotonicity of SimR. Then we have $u_1^a = u_2^a = \bot$ and $u_1^m = u_2^m = \bot$, which imply that $u_1 = u_2$. Therefore, we can deduce that $p_\sigma = \Pr(\mathsf{SimR}(u_2)^\sigma = \bot) \leq \Pr(u_1 = u_2)$. However, we need to note that $\Pr(\mathsf{SimR}(u_1)^{y_2} = \mathsf{SimR}(u_2)^{y_2}) = \Pr(u_1 = u_2)$. Therefore, for both $p_1 = \Pr(\mathsf{SimR}(u_1)^{y_2} > \bot)$ and $1 - p_2 = \Pr(\mathsf{SimR}(u_1)^{y_2} = \bot)$ to be negligible, p_σ must be negligible.

Since $u_2 \leq u_3$, we have $1 - p_\sigma = \Pr(\mathsf{SimR}(u_2)^\sigma > \bot) \leq \Pr(\mathsf{SimR}(u_3)^\sigma > \bot)$. When interacting with Env_3, SimR should output $\sigma \in \{0, 1\}$ with overwhelming probability. Now the current case will go back to that was shown in Proposition 4.1, where it leads to a contradiction in Env_1, i.e., SimR cannot output $\tilde{\sigma} = \sigma$ with overwhelming probability. This fact contradicts the nature of $\mathsf{FindMessy}$ which can find correct messy branch $1-\sigma$ on the information of (crs, τ) with overwhelming probability. Therefore, the above analysis shows that even though we use a weaker OT functionality $\mathcal{F}''_{\mathrm{OT}}$, the security under the corruption with the receiver still cannot be obtained.

5. Conclusions.

In this paper, we analyze the security of a lattice-based OT protocol [14] in a novel framework, called equational security framework, which is a concise model especially for specifying and analyzing protocols in fully asynchronous environments. We found that when we keep using the notion of traditional OT functionality, we cannot prove their claimed security of this lattice-based OT protocol in the equational security framework. Therefore, we made twice modifications on OT functionality, but the result is still negative.

However, we claim that our result cannot be viewed as a cryptanalysis of this lattice-based OT protocol, since we do not point out any specific mistake in its original security proof except for a timing bug. We still believe that this lattice-based OT protocol can provide some meaningful form of security. One possibility is that the simulator constructed in the proof cannot be expressed within this equational security framework. If it is indeed the case, it would be very interesting to extend and upgrade the original equational security framework properly so that it allows to formally analyze this lattice-based OT protocol with respect to an appropriate security definition.

Acknowledgment. This work is supported by the National Natural Science Foundations of China (No.61472309, No.61672412, No.61572390), the National Key Research and Development Program of China under Grants (No.2017YFB0802002), and the China Scholarship Council (No.201406960041). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. O. Rabin, How to exchange secrets with oblivious transfer, (Technical report, 1981) available in *IACR Cryptology ePrint Archive*, 2005: 187, 2005.

[2] S. Even, O. Goldreich, and A. Lempel, A randomized protocol for signing contracts, *Communications of the ACM*, vol.28, no.6, pp. 637–647, 1985.

[3] N. Gilboa, Two party RSA key generation, *Annual International Cryptology Conference*, Springer Berlin Heidelberg, Germany, pp. 116–129, 1999.

[4] M. Naor, and B. Pinkas, Efficient oblivious transfer protocols, *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, pp. 448–457, 2001.

[5] B. Aiello, Y. Ishai, and O. Reingold, Priced oblivious transfer: How to sell digital goods, *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer Berlin Heidelberg, Germany, pp. 119–135, 2001.

[6] A. Y. Lindell. Efficient fully-simulatable oblivious transfer, *Topics in Cryptology-CT-RSA*, Springer Berlin Heidelberg, Germany, pp. 52–70, 2008.

[7] Y. Lindell, How to simulate it-A tutorial on the simulation proof technique, *IACR Cryptology ePrint Archive*, 2016: 46, 2016.

[8] R. Canetti, Universally composable security: A new paradigm for cryptographic protocols, *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, IEEE, pp. 136–145, 2001.

[9] J. Katz, U. Maurer, B. Tackmann, et al., Universally composable synchronous computation, *Theory of cryptography*, Springer Berlin Heidelberg, Germany, pp. 477–498, 2013.

[10] R. Küsters, and M. Tuengerthal, The IITM Model: a simple and expressive model for universal composability, *IACR Cryptology ePrint Archive*, 2013: 25, 2013.

[11] R. Canetti, A. Cohen, and Y. Lindell, A simpler variant of universally composable security for standard multiparty computation, *Annual Cryptology Conference*, Springer Berlin Heidelberg, Germany, pp. 3–22, 2015.

[12] D. Wikström, Simplified universal composability framework, *Theory of Cryptography Conference*, Springer Berlin Heidelberg, Germany, pp. 566–595, 2016.

[13] D. Micciancio, and S. Tessaro, An equational approach to secure multi-party computation, *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, ACM, pp. 355–372, 2013.

[14] C. Peikert, V. Vaikuntanathan, and B. Waters, A framework for efficient and composable oblivious transfer, *Annual International Cryptology Conference*, Springer Berlin Heidelberg, pp. 554–571, 2008.

[15] B. Li, and D. Micciancio, Equational security proofs of oblivious transfer protocols, *IACR Cryptology ePrint Archive*, 2016: 624, 2016.

[16] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently, *Annual International Cryptology Conference*, Springer Berlin Heidelberg, pp. 145–161, 2003.

[17] T. Chou, and C. Orlandi. The simplest protocol for oblivious transfer, *International Conference on Cryptology and Information Security in Latin America*, Springer International Publishing, pp. 40–58, 2015.

[18] D. S. Scott, Domains for denotational semantics, *International Colloquium on Automata, Languages, and Programming.* Springer Berlin Heidelberg, Germany, pp. 577–610, 1982.

[19] C. A. Gunter, and D. S. Scott, *Handbook of theoretical computer science (vol. B) chapter Semantic Domains*, MIT Press, Cambridge, MA, USA, pp. 633–674, 1990.

[20] V. Stoltenberg-Hansen, I. Lindström, and E. R. Griffor, *Mathematical theory of domains*, Cambridge University Press, 1994.

[21] D. Micciancio, and O. Regev, *Lattice-based cryptography*, Springer-Verlag Berlin Heidelberg, Germany, pp. 147–191, 2009.

[22] C. Peikert, Some recent progress in lattice-based cryptography, *TCC*, vol. 5444, p. 72, 2009.

[23] D. J. Bernstein, J. Buchmann, and E. Dahmen (eds.), *Post-quantum cryptography*, Springer-Verlag Berlin Heidelberg, Germany, 2009.

[24] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[25] C. Gentry, C. Peikert, and V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, *Proceedings of the fortieth annual ACM symposium on Theory of computing*, ACM, pp. 197–206, 2008.