Post-Quantum Universal Composable OT Based on Key Exchange

DING HANGCHAO1, HAN JIANG2, AND QIULIANG XU2

1School of Mathematics, Shandong University, Jinan 250100, China
2School of Software, Shandong University, Jinan 250101, China

Corresponding authors: Qiuliang Xu (xql@sdu.edu.cn) and Han Jiang (jianghan@sdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61632020, in part by the Major Scientific and Technological Innovation Project of Shandong Province under Grant 2018CXGC0702, in part by the Development and Construction Funds Project of National Independent Innovation Demonstration Zone in Shandong Peninsula under Grant S19010101001, and in part by the Primary Research and Development Plan of Shandong Province under Grant 2018GGX101037.

ABSTRACT We construct a universal composable framework for two-message oblivious transfer protocols based on lattice-assumption. Compared with the paper proposed by Liu and Hu, we modify a framework proposed by Liu and Hu by adding three tools, which are XOR, Bit Commitment and Smooth Projective Hash Function (SPHF). We instantiate Hash Function as SPHF, which can be more secure in practical application and can achieve full-simulatable in security proof. Compared with Hash Function used as Random Oracle in security proof, this construction is more secure and efficient in security proof. In particular, we mainly consider non-adaptively malicious adversary in Oblivious Transfer protocols.

INDEX TERMS Oblivious transfer (OT), key exchange (KE), universally composable (UC), lattice assumption, RLWE (Ring-LWE), smooth projective hash function (SPHF).

I. INTRODUCTION

Oblivious Transfer was proposed by Rabin in 1981 [1], which is based on integer factoring. Oblivious Transfer protocol has two participants, sender and receiver. We mainly focus on 1-out-of-2 OT, where sender holds two message $M_1$, $M_2$, and receiver selects any bit to obtain $M_i, i \in \{0, 1\}$. Sender transmits messages $M_1, M_2$ to receiver by OT protocol. The security of OT ensures that sender doesn’t know which message receiver obtains. Receiver can only obtain his chosen message $M_i$, and has no information about $M_{1-i}$.

Oblivious Transfer can be constructed by public key cryptosystem (PKC). Most OT are based on number theory assumption. And OT can be used to construct other MPC protocols, including Zero Knowledge Proof, Bit Commitment etc. In 1985, Even and Goldreich proposed a randomized OT protocol for signing contracts protocol, which can be implemented by PKC (public key cryptosystem) [2]. OT protocol can be used as main technical means to achieve privacy protection. As a basic cryptographic protocol, OT protocol can be widely used in MPC, including secret information retrieval, e-commerce (online ordering, payment browsing and securities trading, etc.), etc. As a basic cryptographic primitive, OT can be used to construct Oblivious Circuit Evaluation proposed by Kilian in 1988 [3]. Kilian also mentioned that OT can be useful to noninteractive zero-knowledge proof and commitment protocol. Bellar and Micali based on DH assumption and design non-interactive OT in 1989 [4]. In 1991, Bennett proposed a practical quantum OT, which is based on quantum physics. This quantum OT can implement bit commitment and oblivious circuit evaluation [5]. In 1995, Beaver proposed a pre-computing oblivious transfer, which had efficient computability by obtaining intensive online computation without unproven hard assumption [6]. Crépeau proposed a efficient committed OT and PMPC (private multi-party computation). BC (bit commitment) and 1-out-of-2 OT are used as basic tools to construct PMPC used in multi-party secure computation [7]. In 1999, Naor and Pinkas proposed k-out-of-n OT, which has various applications in protecting privacy combined with commitment. This OT_{k/n} based on sum consistent synthesizers, and make many invocations of OT_2 [8]. In 2000, Gertner and Kannam introduce the relationship between public key encryption and oblivious transfer, which is incomparable under the reduction of black box [9].

The security model of OT and extending-OT are based on random oracle, one-way function, homomorphism and universally composable etc. In 2001, Naor and Pinkas proposed...
efficient oblivious transfer protocols. They constructed a two round protocol based on DDH assumption, whose security proof wasn’t based on random oracles [10]. In 2002, Tzeng proposed efficient OT\textsuperscript{1} schemes, which achieves optimal round and obtain receiver’s security without any conditions. These schemes can apply (symmetric) private information retrieval (PIR/SPIR) between user and database manager (DBM) [11]. Mu and Zhang proposed OT\textsuperscript{m} based on DL assumption, which have better completeness, robustness and flexibility. Specially, they construct a non-interactive OT. These constructions have suitable application in electronic commerce [12]. In 2003, Ishai and Kilian proposed an extending OT technique in the random oracle. Compared with extending OT based on one-way function proposed by Beaver [13], this technique had better practicability [14]. Lipmaa combined HOT protocol with homomorphic public key cryptosystems to construct verifiable homomorphic oblivious transfer and private equality test [15]. In 2004, Crépeau and Morozov proposed an unconditional secure OT whose setting is from any noisy channel [16]. Ogata and Sasahara extend OT\textsuperscript{1} proposed by Naor and Pinkas whose security was based on random oracle, but the security of OT\textsuperscript{2} didn’t need random oracle [17]. Garay and Mackenzie extend committed oblivious transfer to obtain ECOT (extended committed oblivious transfer), which can achieve security in universal composability model. ECOT protocol can be used as building primitive to apply in some MPC’s function [18]. In 2005, Huang and Chang improved Mu’s OT\textsuperscript{m} protocol provided in 2002, which cannot provide security of receiver. And this improvement didn’t have impact on efficiency [19]. Chu and Tzeng proposed an efficient OT\textsuperscript{k} based on DDH and CT-CDH (chosen-target CDH) with universal parameters, which was most efficient in communication cost. The proposal adaptive OT\textsuperscript{k} with commitment and transfer phase [20]. Harnik and Kilian introduced a robust combiner which can combine OT with other primitives to obtain universal protocols [21].

Smooth Projective Hash Function (SPHF) originally was used to construct OT for half-simulatable security. Then modified SPHF can be used to construct full-simulation OT [48]. Full-simulation mainly matches non-adaptive adversary in most cases. Kalai constructed two-message OT protocols using modified smooth projective hash function (SPHF) provided by Gramer and Shoup in 2002 [22], whose security were based on N’th residuosity and quadratic residuosity [23]. In 2006, Wolf and Wulschlegler proposed a reduction and proved that OT is symmetric function in mathematical level [24]. Fischlin construct an universal composable OT schemes used in multi-party setting for honest participants [25]. In 2007, Huang proposed a OT\textsuperscript{k} which has lower bandwidth and speeds up calculation [26]. Green and Hohenberger proposed blind identity-based encryption (IBE) and applied this protocol to construct simulatable OT [27]. Camenisch and Neven constructed first OT based on blind signatures in the random oracles. And they proposed second OT which only need a constant number of group elements without random oracle model [28]. Meier and Braydatek constructed an uniform OT-combiner which can achieve optimal robustness [29]. In 2008, Green and Hohenberger proposed an universal composable secure adaptive OT\textsuperscript{k} in the static corruption model [30]. Lindell proposed an efficient OT which could obtain fully-simulatable security with assumptions of DDH, QR and FHE [31]. Haitner proposed an black-box technique that malicious OT protocol can be reduced to semi-honest OT [32]. Shakar and Srinathan proposed an alternative reduction used for generalized OT, combining with secret sharing to replace private protocols [33]. Dowsley and Graaf proposed an OT which reduced to Mceliece assumptions [34].

Key Exchange (KE) can be used to construct OT with the help of SPHF. In addition to KE, lossy function, dual-mode cryptosystems and cut-and-choose technique are used to construct OT. Lattice-based OT have been proposed in recent years. Lattice-based OT are believed to quantum-resistance. LWE as a average-case lattice assumption are always used to construct OT, which has better reduction property. Parakh proposed that key exchange (KE) can be used to implement OT [35]. Peikert and Vaikuntanathan proposed a framework to construct efficient and universally composable OT. Messy public keys had necessary role in dual-mode cryptosystems, which used in PVW framework to implement statistical security [36]. In 2009, camerich and Dubovitskaya proposed a OT protocol with anonymous access to database [37]. Rial and Kohlweiss proposed an priced oblivious transfer (POT) scheme between vendor and buyer, which was based on pairing-related assumptions in the standard model [37]. Lindell and Zarosim proposed adaptive zero-knowledge proofs and adaptively secure OT [38]. Wang and Ishwar proposed bootstrap string OT and secure two-party function computation [40]. Qin and Zhao proposed a non-interactive OT protocols [41]. Katz and Vaikuntanathan proposed an public key encryption scheme based on LWE assumption. This PKE schemes combined with SPHF to construct password-based authenticated key exchange (PAKE) based on lattice [42]. In 2010, Jain and Hair proposed a efficient OT\textsuperscript{k} protocol, which was based on Diffie-Hellman key exchange [43]. Isaka proposed an OT which used in additive white Gaussian noise channel [44]. In 2011, Green and Hohenberger proposed a practical adaptive oblivious transfer from simple assumptions [45]. Ma and Xu proposed an oblivious transfer with time-release receiver’s privacy [46]. Tassa proposed a generalized OT combined with secret sharing [47]. In 2012, Halevi and Kalai improved SPHF and two-message OT which first proposed by Kalai in 2005 [48]. Cannetti and Dachman-soled proposed a efficient password authenticated key exchange via OT [49]. Lindell and Pinkas proposed an secure two-party computation via cut and choose OT [50]. Zeng and Tartary proposed a practical framework for OT\textsuperscript{n} with security against covert adversaries [51]. In 2013, Asharov and Lindell proposed more efficient OT and some extensions which can speed up secure computation [52].
Canetti proposed an universally composable security model for cryptographic protocols [53] in 2001, which was based on simulation technique. In 2014, Guleria and Dutta proposed an efficient adaptive OT in universal composable framework [54]. David and Dowsley proposed an universally composable OT based on LPN assumption, which belongs to lattice assumption [55]. In 2015, Blazy and Chevalier proposed a generic construction of UC-secure OT [56]. Zhao and Jiang proposed an cut-and-choose bilateral OT. As a basic technique cut-and-choose has necessary role in designing OT protocols [57]. Asharov and Lindell proposed OT extensions which achieved security under malicious adversaries [58]. In 2017, Jannati and Baharak proposed an OT protocol based on elgamal encryption, which can apply preserving location privacy [59]. In 2018, Li and Xiang proposed an OT combined with lossy encryption under lattice-based assumption [60]. Li and Micciancio proposed an equalitarian security proofs for OT protocols [61]. Branco and Ding proposed an universal composable OT which was based on RLWE assumption [62]. In 2019, Liu and Hu proposed an UC-secure OT based on ideal lattice [63]. Li and Ma proposed an efficient OT by multiple bits dual-mode cryptosystem, which can apply in the cloud [64]. Li and Huang proposed a post-quantum OT protocol [65]. Bu and Agrawal proposed some cryptographic hardware primitives which were resist to quantum [66]. Döttling and Garg proposed a two-round OT which was based on CDH or LPN assumptions [67]. For better understanding key exchange based on LWE, Ding and Jiang proposed an research on key exchange based on LWE [68], including some relation between key exchange protocols.

A. OUR CONTRIBUTION

As an important cryptographic primitive, Oblivious Transfer (OT) plays an important role in Multi-party Secure Computation (MPC) protocols. As a basic protocol used in transporting key, key exchange (KE) can also be considered as a technique to construct OT. Most OT constructions are based on DDH, RSA, et al., those primitives may become unsafe as the development of quantum computing. Latticed-based cryptography are believed to be quantum-resistant, we use LWE (RLWE) assumption to construct post-quantum OT protocols. Combined with KE protocol, we mainly think about ‘universally composable’ security model.

We modify a framework proposed by Liu and Hu (2019) [63], which construct a UC-secure OT based on KE. Combined with XOR and Bit Commitment (BC), we can obtain better security with the help of BC’s privacy and blinding properties. With the help of SPHF, we can obtain full-simulation OT which can resist non-adaptive malicious adversary in all corruption cases.

Compared to OT proposed by Liu’s protocol, we present an OT which is secure in standard model. Liu proposed a UC-secure OT in random oracle (RO) model. However, practicability is negligible in RO model. To some extent, protocols based on RO model is insecure than protocols based on standard model.

B. ORGANIZATION

In Section 2, we describe notations used in this paper and some preliminaries about lattice assumption, which include LWE, error distribution and reduction relations. In Section 3, we mainly introduce several basic primitives, which include Key Exchange (KE), Smooth Projective Hash Function (SPHF), Bit Commitment (BC), Zero-Knowledge Proof (ZKP), Oblivious Transfer (OT) and Universally Composable (UC). In Section 4, we introduce a UC-secure OT protocol, which is based on Perkert’s KE, XOR, BC and SPHF. In Section 5, we propose security proof. It mainly include two phases, which are key initialization and section phase, key transfer and commitment phase. We mainly apply Simulation tools to prove security against non-adaptive malicious adversary. In Section 6, we give a conclusion.

II. PRELIMINARIES

At present, there are five cryptographic systems to resist quantum attacks, which are Lattice-based, Code-based, Multivatiable-based, Hash-based and Hypersingular Homology-based. Due to its flexible structure and rich functions, lattice cryptography is the most common type of cryptography system. Almost all classical cryptography primitives can be implemented in lattice cryptography.

A. BASIC NOTATION

As an certain algebraic structure, lattice has an important role in quantum cryptography. It mainly includes CVP, SVP, SIS, LWE and their variants. As a discrete additive subgroup, lattice cryptography is the most common type of cryptography system. Almost all classical cryptography primitives can be implemented in lattice cryptography.
as reduction oracle which can reduce protocols to lattice assumptions. In most cases, we mainly apply LWE and SIS assumptions. Because \textit{CVP}/\textit{SVP} assumptions in worst case can be reduced to LWE/SIS assumptions in average case.

A negligible function is always used to describe negligible success probability. A function \( f \) is negligible, meaning that the following conditions are satisfied. If for any positive polynomial \( p(n) \), existing an \( N \), holds \( f(n) < 1/p(n) \) for all \( n > N \).

Considering that adversary \( A \) corrupt normal participant in protocols, we denote a sequence of \textit{Game}_i (\( i \in \{0, 1, 2\} \)), which is used to simulate in security proof. In ideal world, we denote \( S \) as a simulator, which is used to simulate adversary \( A \). For better understanding the environment, we denote \( Z \) as an indistinguishable box, which has the view of indistinguishability between real world and ideal world. We denote real OT protocol as \( P \) in real environment. Denote ideal function \( F_{OT} \) in ideal environment, which can implement the function of protocol \( P \).

**B. LATTICE**

Lattice can be seen as discrete additive subgroup. Lattice can be regarded as a linear combination of integral coefficients under a set of linear independent lattice basis. \( B = \{b_1, b_1, \ldots b_n\} \) are linearly independent basis vectors, \( L(B) = \{bx \in \mathbb{Z} | x \in \mathbb{Z} \} \).

LWE has necessary variance which can be reduced from worst-case to average-case. Considering efficiency and practicability, some schemes based on LWE or LWE’s variants have weaker practicability on account of key and ciphertext size.

1) LWE

Learning with errors assumption mainly means that LWE’s pairs are indistinguishable from uniform distribution. LWE’s pair means that \( a \) and \( s \) are chosen from uniform distribution, \( e \) is chosen from certain distribution, such as Gaussian distribution, Centered binomial distribution, output \( (a, \langle a, s \rangle + e) \).

2) SEARCH-LWE

Given polynomial LWE’s pairs, find correct solution of \( s \), simplify as S-LWE. From one point of view, given a set of unary polynomial equations, it is difficult to solve \( s \).

3) DECISION-LWE

Given polynomial samples, it’s negligible to distinguish LWE’s pairs from uniform distribution, simplify as D-LWE.

4) REDUCTION FROM DECISION TO SEARCH [69]

Assuming that \( q = \text{poly}(n) \), there exist an algorithm, which can distinguish a certain distribution \( \chi \) from uniform distribution. Then there is an algorithm, which can solve \( s \) in LWE’s pairs. Considering the difficulty of solving \( s \), the probability of succeeding in distinguishing certain distribution from uniform distribution is negligible.

5) GAUSSIAN DISTRIBUTION

For \( s > 0 \), define function \( \rho_s(x) = \exp(-\pi ||x||^2/s^2) \). Gaussian distribution \( D_{Z,s} \) means that \( x \) samples on \( Z \) with probability of \( \rho_s(x) \).

6) CENTERED BINOMIAL DISTRIBUTION

For \( b_1, b_1' \in [0, 1]^n \), define distribution \( A_{\text{center}} \) means that output each bit is \( (b_1 - b_1') \in \{-1, 0, 1\} \), \( i \in \{1, 2, \ldots, n\} \). The probability of outputting 0 is \( \frac{1}{2} \), and the probability of outputting \( \pm 1 \) is \( \frac{1}{4} \).

**C. IDEAL LATTICE**

Considering that ideal lattice basis has the property of cyclic, lattice can be constructed by an ideal lattice basis. Ordinary lattice can be constructed by different basis, which need to choose a suitable basis in application. Comparatively speaking, ideal lattice is better applied than ordinary lattice in Public Key Cryptosystems (PKC), which is only constructed by ideal basis.

Ideal lattice as a variant of lattice, has many advantages applied in PKC, which shorten the length of key and ciphertext and has better efficiency in communication.

1) RLWE

Ring learning with errors assumption can be seen as LWE’s problem based on ideal lattice. Certain distribution \( \chi \) means sampling from \( \mathbb{R}_q \) in terms of distribution function, such as Gaussian Sampling. Centered Binomial Sampling. etc. For random \( s \in \mathbb{R}_q \), and certain distribution \( \chi \), input \( a \in \mathbb{R}_q, e \in \chi \), output \( (a, b = \langle a, s \rangle + e) \in \mathbb{R}_q \times \mathbb{R}_q \). When \( \mathbb{R} = \mathbb{Z} \), this RLWE is plain LWE.

2) SEARCH-RLWE

Given RLWE’s pairs \( (a, b = \langle a, s \rangle + e) \in \mathbb{R}_q \times \mathbb{R}_q \), find solution of \( s \), simply as S-RLWE.

3) DECISION-RLWE

Define RLWE’s pairs distribution as \( A_{s, \chi} \). Decision-RLWE means that distinguish distribution \( A_{s, \chi} \) from uniform distribution \( \mathbb{R}_q \times \mathbb{R}_q \).

**III. BASIC PRIMITIVES**

**A. KEY EXCHANGE**

1) DIFFIE-HELLMAN KEY EXCHANGE

Diffie and Hellman proposed that key exchange(KE) based on discrete logarithm. After choosing suitable parameters, we mainly introduce this KE cryptosystem, sender Alice sends \( A = g^a \) to receiver Bob, Bob sends \( B = g^b \) to Alice. Then Alice computes \( B^a \) and Bob computes \( A^b \). Finally both of them get same key \( g^{ab} \) [70].

2) POST-QUANTUM KE

More KE protocols were proposed based on number theory’s assumptions, such as RSA, bilinear pairings, elliptic curve, which were like Diffie-Hellman’s KE. To defend against
possible quantum attacks, cryptographist proposed some KE protocols which are resistant quantum attacks. NIST called for post-quantum schemes submission in 2016. After two-round competition, there are four kinds of schemes, lattice-based, code-based, multivariable-based, hash-based. Considering lattice cryptography to resist quantum attack, we mainly apply KE which is based on lattice assumptions.

3) PEIKERT’S KE

This reconciliation mechanism was proved by Peikert. For odd prime \( q \geq 2 \), define \( W_q = \{ \frac{q-1}{2}, \ldots, \frac{q-1}{2} \} \), \( Z_2 = \{ 0, 1 \} \) and define the subset \( W_q \), \( E = \{ -\frac{1}{2}, \ldots, \frac{1}{2} \} \) as the middle of \( W_q \), \( \tilde{v} =dbl(v) = 2v - \bar{e}, \bar{e} \in \chi \) is randomly uniform. Denote modular rounding function as \( \langle v \rangle_2 = \left\lfloor \frac{1}{q} \cdot v \right\rfloor \mod 2 \) and \( \lceil v \rceil_2 = \left\lceil \frac{1}{q} \cdot v \right\rceil \mod 2 \). Denote mod function as \( \text{mod}_2(v, b) = (v + b \cdot \frac{1}{q}) \mod 2 \).

Define \( \text{Cha} \) function as follows:

\[
\text{Cha}(v) = \begin{cases} 0, & v \in E. \\ 1, & \text{else} \end{cases}
\]

If \( q \) is even, corresponding function is \( \text{rec} : W_q \times Z_2 \rightarrow W_2 \); if \( q \) is odd, corresponding function is \( \text{rec} : W_q \times Z_2 \rightarrow Z_2 \). We introduce \( \text{rec} \) function when \( q \) is even, and another situation is similar to it.

\[
\text{rec}(w, g) = \begin{cases} 0, & w \in I_b + e, e \in E, b \in \{ 0, 1 \} \\ 1, & \text{else}. \end{cases}
\]

Considering that modulus is even during key exchange, we mainly use several functions to achieve key exchange by every bit. For better understanding key exchange under even modulus and odd modulus, we briefly introduce key exchange in even modulus \( q \geq 4 \). Considering that modular rounding function \( \lceil v \rceil_2 = \left\lceil \frac{1}{q} \cdot v \right\rceil \mod 2 \) mainly applies in \( I_b \) and \( \frac{q}{2} + I_b \), we calculate the values of the functions in different sets.

**TABLE 1. Peikert’s reconciliation mechanism when modulus is even.**

| Alice | Bob |
|-------|-----|
| \text{KEM.Gen}(a) | \text{KEM.Encaps}(a, b) |
| \begin{align*}
\text{s}, e \leftarrow \chi \\
\text{rec} = \left\lceil \frac{1}{q} \cdot v \right\rceil \\
\text{KEM.Decaps}(s, \langle v \rangle_2) \\
\mu = \text{rec}(w, \langle v \rangle_2)
\end{align*} |
| \begin{align*}
\text{s_1, s_2} \leftarrow \chi \\
\text{u = } s_1 + e_1 \\
\text{v = } s_2 + e_2 \\
\langle v', \mu \rangle
\end{align*} |

When \( v \in I_b \), we can obtain \( \lceil v \rceil_2 = 0 \), otherwise \( \lceil v \rceil_2 = 1 \).

When \( v \in \frac{q}{2} + I_b \), we can obtain \( \frac{q}{2} + I_0 = \{ \frac{q}{2}, \frac{q}{2} + 1, \ldots, \frac{q}{2} + \left\lfloor \frac{q}{2} \right\rfloor \} \) and \( \frac{q}{2} + I_1 = \{ \frac{q}{2} - \left\lfloor \frac{q}{2} \right\rfloor + 1, \ldots, \frac{q}{2} - 1 \} \).

Substitute \( \frac{q}{2} - \frac{q}{2} + \left\lfloor \frac{q}{2} \right\rfloor - 1, \frac{q}{2} - \left\lfloor \frac{q}{2} \right\rfloor - 1 \) into rounding function \( \left\lfloor \frac{1}{q} \cdot v \right\rfloor \mod 2 \), then we can obtain that \( \lceil v \rceil_2 = 1 \).

So we can get \( \lceil v \rceil_2 = \begin{cases} 0, & 0 \leq v \leq I_b \\ 1, & v \in \frac{q}{2} + I_b \end{cases} \).

For better understanding this reconciliation mechanism, several theorems have been cited from [71], as follows.

**Lemma 1:** When \( q \) is even integer, given \( v \) is uniformly distributed in \( \mathbb{R}_q \) and \( \langle v \rangle_2 \), obtain that \( \lceil v \rceil_2 \) is also uniformly distributed in \( \mathbb{R}_q \).

**Lemma 2:** When \( q \) is even integer, \( g = v + e \mod 2, v \in \mathbb{R}_q, e \in E \), obtain \( \text{rec}(g, \langle v \rangle_2) = \langle v' \rangle_2 \).

The above is the case of even modulus, we’ll nextly introduce the case of odd modulus. Considering practical application in security, we mainly use odd modulus. For better understand key exchange in modulus of odd integer, we briefly introduce key exchange in even modulus.

**TABLE 2. Peikert’s reconciliation mechanism when modulus is odd.**

| Alice | Bob |
|-------|-----|
| \text{KEM.Gen}(a) | \text{KEM.Encaps}(a, b) |
| \begin{align*}
\text{s, e} \leftarrow \chi \\
\text{b = } a + e \\
\text{KEM.Decaps}(s, \langle u, v \rangle) \\
\mu = \text{rec}(2u, v')
\end{align*} |
| \begin{align*}
\text{s_1, s_2} \leftarrow \chi \\
\text{b = } a + e_1 \\
\text{v = } s_1 + e_2 \\
\langle v', \mu \rangle
\end{align*} |

For better apply designing of even modulus to odd modulus, make use of \( \text{dbl}(v) = 2v - \bar{e} \) function to complete transition between even modulus and odd modulus. Notice sender should apply \( \text{rec} \) function in 2us.

**Lemma 3:** When \( q \) is odd integer, \( v \) is random uniform in \( \mathbb{R}_q \), and \( \tilde{v} = 2v - \bar{e} \) is uniformly random in \( \mathbb{R}_{2q} \). Given \( \langle \tilde{v} \rangle_2 \), obtain that \( \lceil \tilde{v} \rceil_2 \) is uniformly distributed in \( \mathbb{R}_{2q} \).

**B. SMOOTH PROJECTIVE HASH FUNCTION**

Smooth Projective Hash Function (SPHF) is based on hard subset membership problem. Given a set \( U \), which mainly refers to the distribution on \( U \), and an NP-language \( D \subset U \). Hardness assumption is that it’s indistinguishable between random element from \( D \) and random element from \( U \setminus D \).

SPHF has two keys, hash key and projection key. These keys are closely related with SPHF’s requirement and property. Define \( K_h \) as hash key. Define \( K_p \) as projection key. Define SPHF as a hash function \( H : U \rightarrow \{ 0, 1 \}^n \).

**Standard Property** Given hash key \( K_h \) and random element \( x \) from \( U \), user can obtain the value of hash function \( H(K_h, x) \).

**Projection Property** Given projection key \( K_p \) and random element \( y \) from \( D \), user can obtain the unique value of hash function \( H(K_p, x) \).

**Smoothness Property** Given projection key \( K_p \) and random element \( y \) from \( U \setminus D \), user can obtain indistinguishability between the value of hash function and random value. It means the indistinguishability between \( \{ y \in U \setminus D, H(K_p, y) \} \) and \( \{ y \in U \setminus D, v(0, 1) \}^n \).

**C. BIT COMMITMENT**

We introduce Bit Commitment schemes between sender and receiver. It mainly includes two phases: Commit Phase and Reveal (Decommit) Phase. Bit Commitment (BT) schemes
are used to commit on some message for sender in Commit Phase. Then this commitment will be opened by receiver in Reveal Phase.

**Commit Phase** Sender make a commitment on bit $\sigma$ and send this commitment to receiver. Receiver has no information about $\sigma$.

**Decommit(Reveal) Phase** Sender prove that commitment above mentioned is relevant to bit $\sigma$, and sender can’t modify the value of $\sigma$.

Bit Commitment schemes should satisfy three properties, which are correctness, privacy and binding.

**Correctness Property** If sender and receiver honestly run BC protocol, receiver will obtain correct bit $\sigma$ committed by sender. It holds $Pr[\text{Decommit}(Y_i) = M_i|Y_i = \text{Commit}(M_i)] = 1$.

**Privacy/Hiding Property** In Commit phase, receiver has no information about $\sigma$. Define event $E$ as adversary distinguish $Y_0$ from $Y_1$, it holds $Pr[E] < \text{negl}(n)$ for any adversary.

**Binding Property** In Decommit phase, receiver only obtain unique $\sigma$. And sender can’t modify the value of $\sigma$. It holds $Pr[\text{Commit}(M_i) = \text{Commit}(M_{i-1})] < \text{negl}(n)$ for any adversary.

Some symmetric algorithms and Hash function are always used to construct BC schemes. Considering some BC based on lattice, we can apply BC which is correlative with lattice assumptions.

**D. ZERO-KNOWLEDGE PROOF**

The interactive zero-knowledge proof was first proposed by Goldwasser, and then Blum modified it and proposed the non-interactive zero-knowledge proof. Currently, we learn more zero-knowledge proof protocols of Stern type and Fiat-Shamir with abort type(FSWA). In 2019, Yang et al. [72] proposed a zero-knowledge proof protocol for lattice-related relationships, which solve disadvantages of Stern type’s reliability error rate and low efficiency of FSWA’s type.

In brief, zero-knowledge proof protocol has two parties, certifier and verifier. It mainly includes the process of proof and the process of verification. Turing machines are often used in formal proof of security. We can get an evidence or assertion, which can become true. But it doesn’t disclose the process of finding the evidence.

A complete zero-knowledge proof protocol satisfies the following three properties. We mainly introduce interactive system $< P, V >$ for relation $L$, which includes prover $P$ and verifier $V$.

**Completeness** From the view of certifier, correct assertions can be proved, and wrong assertions would be invalid. It holds $\forall x \in L, Pr[< P, V > (x) = 1] = 1$

**Soundness** From the view of verifier, if certifier cheat and give deceptive argument, verifier will not be deceived by certifier’s proof. It holds $\forall x \notin L, Pr[< \land, V > (x) = 1] < \text{negl}(n)$ for any adversary $\land$.

**Zero Knowledge** Verifier can give correct statements and will not reveal any information. Verifier can only obtain the result of relevant statement, and can’t get any other useful information.

**E. OBLIVIOUS TRANSFER**

Oblivious transfer(OT) is a common basic and underlying protocol in secure multi-party computing(MPC). We briefly introduce $OT_N^1$. Sender sends $N$ messages to receiver, who only obtains relative message. Sender does not know what information receiver chooses to receive.

In general, when designing OT protocol, we need to use ideal functions based on OT function during security proof phase. Next, I’ll briefly introduce ideal function for implementing $OT_N^1$ functionality $F_{OT_N^1}$. We mainly consider the case of $N = 2$.

1) **IDEAL FUNCTION $F_{OT_N^1}$**

After a series of initialization settings, sender sends message $(M_1, M_2, \ldots, M_n)$ to $F_{OT_N^1}$. Receiver sends selected bit to $F_{OT_N^1}$. $F_{OT_N^1}$ and stores messages $(M_1, M_2, \ldots, M_n)$. Then receiver sends array information (sid,receiver,i) to $F_{OT_N^1}$. $F_{OT_N^1}$ checks for the its message in the stored information. If so, sender sends $M_i$ to receiver, and stop running program. If no information exists, receiver will receive nothing.

2) **ATTACK TYPES**

From the perspective of the participants, there are four attack types to consider: only sender is corrupted by adversary; only receiver is corrupted by adversary, both sender and receiver are corrupted adversary, and neither sender nor receiver is corrupted by adversary.

**FIGURE 1. relation between real and ideal environment.**

**F. UNIVERSALLY COMPOSABLE**

UC framework was proposed by Canetti [53] and combination theorem was put forward. When analyzing security, a certain protocol was firstly analyzed in an independent model. After meeting the security requirements, combination theorem was applied to carry out parallel operation with other protocols in a composite environment, and finally overall protocol remained secure. The proposed UC framework greatly simplifies the calculation of security definition and security proof of composite protocol. The UC framework is very suitable for complex network society. In this paper, we mainly consider the generalized combination, which refers to the combination of cryptographic different protocols. In this paper, we mainly consider the security of
combination protocols between key exchange protocol, bit commitment protocol and oblivious transfer protocol under UC framework.

It mainly aims to analyze security of composite protocols based on indistinguishability between real process and ideal process. In the real environment, there are mainly participants $P_i$, adversary $A$ and environment $Z$ interacting with each other to complete real protocol $P$. Correspondingly, there are mainly $P_i$, simulator $S$ and environment $Z$ interacting with each other to complete ideal function $F$ in the real environment.

1) UC SECURITY [53]

For protocol $P$, if the following conditions are satisfied, protocol $P$ reaches UC security. For the interaction between real adversary $A$ and protocol $P$ in real environment $Z$, there is an ideal simulation adversary $S$ interacting with ideal function $F$. Protocol $P$ reaches UC security when it’s indistinguishable between real state and ideal state interacting with each other.

| TABLE 3. UC-secure OT protocol. |
|----------------------------------|
| **Sender** | **Receiver** |
| $KEM.Gen(a)$ | $KEM.Encaps(a, b)$ |
| $s_i, e_i \leftarrow \chi$ | $s_i, e_1, e_2 \leftarrow \chi$ |
| $b \leftarrow as + e$ | |
| $t \in U(R_q)$ | $b, t \rightarrow u = as_1 + e_1$ |
| | $v = bs_1 + e_2$ |
| | $\sigma \leftarrow db(v)$ |
| | $f = \langle \sigma, y_2 \rangle$ |
| | if $\sigma = 0, A \leftarrow u \oplus b$ |
| | if $\sigma = 1, A \leftarrow (u + t) \oplus b$ |
| $KEM.Decaps(s, (u, f))$ | $A \leftarrow$ |
| $W_0 = A \oplus b$ | |
| $W_1 = A \oplus b - t$ | |
| $K_{sk_0} = \text{rec}(W_0, s, f)$ | $K_{sk} = [\mathbb{V}]_2$ |
| $K_{sk_1} = \text{rec}(W_1, s, f)$ | |
| $t \in \hat{U}(R_q)$ | |
| $Y_0 = \text{Commit}(K_{sk_0}, l)$ | |
| $Y_1 = \text{Commit}(K_{sk_1}, l)$ | $Y = \text{Commit}(K_{sk}, l)$ |
| $Y_0, Y_1, l$ | $\text{Verify}(Y_0, Y_1, Y) = \beta, \beta \in \{0, 1\}$ |

If $\beta = 1$, continue following process
If $\beta = 0$, stop running protocol, output $\bot$ $r \in U(R_q)$ $K_0 = \text{SPHF}(K_{sk_0}, r)$ $K_1 = \text{SPHF}(K_{sk_1}, r)$ $C_0 = \text{Enc}(K_0, M_0)$ $C_1 = \text{Enc}(K_1, M_1)$ $\sigma = \text{Dec}(K_0, C_0)$ $\text{SPHF}\sigma_r$ $\text{SPHF}_r$ $K_{\sigma} = \text{SPHF}(K_{sk}, r)$ $M_{\sigma} = \text{Dec}(K_{\sigma}, C_1)$

IV. UC-SECURE OT PROTOCOL

This protocol includes key exchange phase and encryption phase. Key exchange phase is mainly based on two mathematical functions, random function and reconciliation function. Firstly, sender and receiver get same initial key. Sender applies Commitment algorithm, and receiver verifies the value of commitment. Next, sender and receiver can make sure that this protocol continue or stop.

**Commitment**: Sender makes commitments for two keys $K_{sk_0}$ and $K_{sk_1}$ by applying random value $l$. Then receiver makes commitment for obtained $K_{sk}$ by applying same random value $l$. Commitment scheme consists of KeyGen, Commit and Open algorithms. Considering the security of the post-quantum OT scheme, we mainly apply Commitment scheme based on LWE’s assumption. Sender sends two commitments $Y_0, Y_1$ to receiver. And receiver also makes a commitment $Y$ for initial key $K_{sk}$. Both sender and receiver can open commitment committed by themselves. Receiver checks whether one of the two commitments matches corresponding commitment by himself.

**Verify**: Receiver applies algorithm $\text{Verify}(Y_0, Y_1, Y) = \beta, \beta \in \{0, 1\}$ and verifies whether $Y = Y_0$ or not $Y = Y_1$. When $\beta = 1$, it means $Y = Y_0$ or $Y = Y_1$. Sender and receiver obtain same initial key, so sender can continue running the protocol. When $\beta = 0$, it means that sender and receiver hasn’t obtained same initial key. So sender stop running this protocol.

Then Smooth Projective Hash function is used to get final key. With the help of Enc scheme, sender computes a ciphertext for two messages, and receiver decrypt ciphertext. Considering construction of post-quantum OT, Enc and Dec algorithm can apply SPH system, which is relative to Lattice-SPHF, mainly based on LWE’s assumption [73]. Considering that we propose an UC-secure OT, we give a brief introduction about the process of Enc and Dec algorithm.

In this part, we mainly introduce an approximate-Smooth Projective Hash system. In previous section, we have introduced SPHF, which satisfy several properties. Due to properties of projection and smoothness, SPHF has been applied in Key Exchange Phase. Considering the security of whole protocol, we apply a CCA-secure encryption based on approximate SPH system. We mainly apply a CCA-secure encryption scheme [73]

Lemma 4 [42]: Given parameters $q \geq 2$ and $m \geq 4n \log^2 q$, there is a PPT algorithm $\text{TrapSamp}(1^n, q, m)$, which outputs matrices $B \in \mathbb{Z}_q^{m \times n}, T \in \mathbb{Z}_q^{m \times m}$. And these matrices satisfy two requirements: $\|t_i\| \leq 4\sqrt{m}$, $TB = 0 \pmod{q}$; the distribution of $B$ is statistically indistinguishable with uniform distribution. There is a PPT algorithm $\text{BDDSolve}(T, Z)$, satisfy: If exist a vector $s \in \mathbb{Z}_m^n$, which satisfy $\text{dist}(s, Bs) \leq \sqrt{q}/4$, output $s$. If $\text{dist}(z, Bs) > \sqrt{q}/4$ for any $s \in \mathbb{Z}_m^n$, output $\bot$.

A. CCA-SECURE ENCRYPTION

We mainly apply an CCA-secure encryption for our protocol [73]. Next, we’ll introduce this encryption scheme.

This encryption scheme mainly includes KeyGeneration, Encryption and Decryption algorithms, which is mainly based on LWE’s assumption.
Parameters: Let $n$ be the security parameter, and $l$ be the message length. Denote $q$ as the modulus of the system. Denote $m$ and $n + l + 1$ as the dimension of matrices. Considering Gaussian distribution $\Psi_\beta$.

**KeyGeneration:** Apply TrapSamp algorithm to obtain $(A_{i,b}, S_i, b)$: $\text{TrapSamp}(m \times n + l + 1, q) \rightarrow (A_{i,b}, S_i, b) \in \{1, 2, \ldots, n\}, b \in \{0, 1\}$. Let $pk = (A_{i,0}, A_{i,1}), sk = (S_i, S_i)$. 

**Encryption:** Given message $M \in \mathbb{Z}_q^n$ encrypt $M$ under public key $pk$. In addition to $(pk, sk)$, sender apply signature scheme $\text{SigKeyGen}$ to obtain $(VK, SK)$. Denote matrix $AVK$ as $[A_1, V_{K_1}, \ldots, A_n, V_{K_n}]$. Then sample $s$ from uniform distribution and sample error vector $x$ from Gaussian distribution $\Psi_\beta$. Next compute $y = AVK \cdot (s, 1, M') + x (\text{mod} q)$ and $\sigma = \text{Sign}_{SK}(y)$. Finally outputs ciphertext $C = (VK, y, \sigma)$.

**Decryption:** Firstly parse ciphertext $C$ as $(VK, y, \sigma)$, and verify whether $\sigma$ is a corresponding signature on $y$. If the verification is wrong, outputs ⊥. If the verification is correct, continue next step. Then parse $y$ as $(y_0, y_1, \ldots, y_n, y_1) \in \mathbb{Z}_q^n$. Then continue the loop algorithm’s operation as follows:

```plaintext
for $a = 1$ to $q - 1$, do compute $BDD_Solve(S_{i,b}, ay)$ to obtain $t = (s, a_1, M)$.
  if $a_1 = 1$ then
    if $\|A_{i,b} - ay\| \leq \sqrt{q}/4$ for all $i \in [n]$.
      then output $M/\alpha$ and stop.
  otherwise try next value of $a$.
end
```

**APPENDIX SPH SYSTEM**

Apply this CCA-secure encryption scheme to our OT protocol, we can obtain more secure protocol. Next we’ll introduce an approximate SPH system. [73]

1. Firstly, parse $AVK$ as $(BVK, UVK)$ Denote $\alpha(x) = BVK \cdot x$ as a projection function, which map $(e_1, e_2, \ldots, e_k)$ to $(u_1, u_2, \ldots, u_k) = BVK \cdot (e_1, e_2, \ldots, e_k)$ and satisfy $u_i = BVK \cdot e_i$.

2. Sender inputs hash key $(e_1, e_2, \ldots, e_k)$ and ciphertext $C = (y, M)$, compute $z_i = e_i[y - UVK \cdot (1.M')]$. Then
   
   $$b_i = \begin{cases} 0 & \text{if } z_i < 0 \\ 1 & \text{if } z_i > 0 \end{cases}$$

   Then outputs $H(c, z_i) = b_1 b_2 \ldots b_k$.

3. Receiver inputs a projected key $(u_1, u_2, \ldots, u_k)$, a ciphertext $C = (y, M)$ and a witness $s \in \mathbb{Z}_q^m$ compute
   
   $$b_i = \begin{cases} 0 & \text{if } u_i s < 0 \\ 1 & \text{if } u_i s > 0 \end{cases}$$

   Then outputs $H(C, s) = b_1 b_2 \ldots b_k$ Sender and receiver apply SPH hash function to get final key. Sender applies hash key to obtain two indistinguishable key $K_0$ and $K_1$. Then sender applies projection key to obtain final key $K_\sigma$. Then sender and receiver apply approximate-SPH hash system. Firstly, sender and receiver apply approximate-SPH KeyGeneration to obtain $(pk, sk)$. Sender applies hash key $pk$ to obtain signature $y$ and ciphertext $C$. Then sender sends $y$ and $C$ to receiver. Secondly, receiver verifies signature $y$. If signature can be verified by receiver, continue this protocol. Otherwise, stop this process.

**KEM.Gen(a):**

Sender samples $s, e$ from distribution $\chi$, computes $b = as + e$. Sender samples random value $t$, and sends $b, t$ to receiver.

**KEM.Encaps(a, b):**

Receiver samples $s_1, e_1, e_2$ from distribution $\chi$, computes $u = as_1 + e_1, v = bs_1 + e_2$. Receiver uses random function $\bar{v} = \bar{v}l(v) = 2v - \bar{e} (\bar{v} \in \chi)$. And receiver uses mathematical function $\bar{v}_2 = \lceil \frac{4}{q} \times \bar{v} \rceil \bmod 2$. Finally, receiver computes $A = (a + \sigma t) \oplus b, \sigma \in \{0, 1\}$ and sends $A, f$ to sender.

**KEM.Decaps(s, (u, f))**

Firstly, sender computes $W_0 = A \oplus b - \sigma t, W_1 = A \oplus b - t$. Then sender uses an reconciliation function ‘rec’ for computing $K_{sk_0} = \text{rect}(2W_0, f), K_{sk_1} = \text{rect}(2W_1, f)$. And receiver uses an mathematical function $K_{sk} = \lceil \bar{v}_2 \rceil$ to obtain initial key, which is same as $K_{sk_0}$ or $K_{sk_1}$.

**Commit and Verify**:

Sender computes commitment for two initial key $K_{sk_0}, K_{sk_1}$, Sender choose a random value from uniform distribution and obtains two commitment value by $Y_0 = \text{Commit}(K_{sk_0}, I)$ and $Y_1 = \text{Commit}(K_{sk_1}, I)$. Sender sends $Y_0, Y_1$ and $l$ to receiver. Receiver makes a commitment for $K_{sk}$ and obtains a commitment $Y = \text{Commit}(K_{sk}, I)$. Then receiver applies algorithm Verify(Y_0, Y_1, Y) = $\beta, \beta \in \{0, 1\}$ and verifies whether $Y = Y_0$ or not $Y = Y_1$. When $\beta = 1$, it means $Y = Y_0$ or $Y = Y_1$. Sender and receiver obtain same initial key, so sender can continue running the protocol. When $\beta = 0$, it means that sender and receiver hasn’t obtained same initial key. So sender stop running this protocol.

**Smooth Projective Hash function**

Sender applys Smooth Projective Hash function for better randomness. Firstly, sender samples an $r$ from random distribution and computes $K_\sigma = \text{SPHF}(K_{sk_0}, r)$. Sender sends $SPHF, r$ to receiver. Secondly, receiver uses same $SPHF$ function and same random $r$ for computing hash value. Finally, receiver computes $K_\sigma = \text{SPHF}(K_{sk}, r)$ and check whether obtaining same hash value. If so, sender and receiver will use the final key $K_\sigma$ for next application.

**Enc**:

Sender computes $C_0 = \text{Enc}(K_0, M_0)$ and $C_1 = \text{Enc}(K_1, M_1)$ for two messages. Then sender sends $C_0, C_1$ to receiver.

**Dec**:

Finally, receiver computes $M_\sigma = \text{Dec}(K_\sigma, C_\sigma)$ for obtaining corresponding message.
V. SECURITY PROOF

This UC oblivious transfer protocol mainly includes two phases. The first phase is Key Initialization and Selection Phase, and the second phase is Key Transfer and Enc Phase. Next, I’ll introduce the security of the first phase.

A. KEY INITIALIZATION AND SELECTION PHASE

We mainly consider sender’s security and receiver’s security. Firstly, I’ll introduce sender’s security.

From the view of sender’s security, receiver can only get selected message \( M \), and has no information on \( M_{-a} \) in protocol. If receiver can obtain \( M_{-a} \), meaning that receiver can obtain \( K_{sk_{1-a}} \). We know that \( K_{sk_{1-a}} = rec(2W_{1-a} s, f) \) can be used to compute \( K_{sk_{1-a}} \), and the secret information \( s \) is crucial to obtain initial key. If receiver can obtain initial key \( K_{sk_{1-a}} \), receiver will have ability to obtain \( s \) in RLWE’s assumption. But obtaining \( s \) will contradict with RLWE’s difficulty in lattice theory.

We make use of SPHF functions in forming final key stage. Receiver can also obtain same final key \( K_{1-s} \), which isn’t obtained by \( K_{sk_{1-a}} \). Due to properties of non-collision of Hash functions, we know that the receiver can’t obtain same final key by different from \( K_{sk_{1-a}} \). Due to smoothness property of SPHF function, we can obtain that receiver can’t distinguish the distribution of SPHF’s value from random distribution.

Next, we need to prove that receiver can’t guess \( K_{sk_{1-a}} \).

We’ll apply sequence-of-game approach for security proof. And we mainly constructs a sequence of games, \( Game_0 \) and \( Game_1 \).

\begin{table}[h]
\centering
\caption{Key initialization and selection phase.}
\begin{tabular}{|c|c|}
\hline Sender & Receiver \\
\hline \( KBM.Gen(a) \) & \( KBM.Encaps(a, b) \) \\
\hline \( s, e \leftarrow \chi \) & \( s_1, e_1, e_2 \leftarrow \chi \) \\
\hline \( b = as + e \) & \( v = bs_1 + e_2 \) \\
\hline \( t \in U(\mathbb{R}_q) \) & \( f = \langle v \rangle_2 \) \\
\hline \end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{\( Game_0 \).}
\begin{tabular}{|c|c|}
\hline Sender & Receiver \\
\hline \( s, e \leftarrow \chi \) & \( s_1, e_1, e_2 \leftarrow \chi \) \\
\hline \( b = as + e \) & \( v = bs_1 + e_2 \) \\
\hline \( t \in U(\mathbb{R}_q) \) & \( f = \langle v \rangle_2 \) \\
\hline \end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{\( Game_1 \).}
\begin{tabular}{|c|c|}
\hline Challenger & Adversary \\
\hline \( s, e \leftarrow \chi \) & \( s_1, e_1, e_2 \leftarrow \chi \) \\
\hline \( b = as + e \) & \( v = bs_1 + e_2 \) \\
\hline \( t \in U(\mathbb{R}_q) \) & \( f = \langle v \rangle_2 \) \\
\hline \end{tabular}
\end{table}

In \( Game_1 \), we can obtain the minimum success probability of adversary \( |Pr[S_1] - 1/2| \). We define this probability as \( Adv_1 \). And we can obtain \( Adv_1 \leq |Pr[S_0] - 1/2| \). Considering that \( (a, u) \) and \( (b, v) \) are RLWE’s pairs in \( Game_0 \), (a, u) and (b, v) are random pairs from uniform distribution in \( Game_1 \). Considering Decision-RLWE’s assumption and difference between \( Game_0 \) and \( Game_1 \), Considering that there is a reduction algorithm \( R_1 \). In reduction algorithm, when the input of \( u \) and \( v \) are RLWE’s pair, the output is \( Game_0 \)'s output, and when the input of \( u \) and \( v \) are from uniformly distribution, the output is \( Game_1 \)'s output.
We define $\text{Adv}_{R1}$ as adversary succeeding in reduction algorithm $R1$. The probability $\text{Adv}_{R1}$ of succeeding in distinguishing RLWE’s distribution and uniform distribution.

We can obtain that the probability of succeeding in $R1$ is negligible in reduction algorithm $R1$ due to the hardness of Decision-RLWE’s assumption. So we can obtain that $\text{Adv}_{R1} \leq |\Pr[S_1] - \Pr[S_0]|$. Considering indistinguishability of RLWE’s distribution and uniform distribution, we can obtain $\text{Adv}_{R1}$ is negligible.

We have completed the proof of sender’s security in the above. Secondly, I’ll introduce receiver’s security.

The whole protocol is a four-round protocol, receiver sends $A, f$ to sender in second round. Receiver might reveal some information in this round. So we need prove receiver’s security. In another view, we should prove that adversary can’t obtain useful information about selected bit.

Firstly, we’ll prove that selected bit can’t be revealed by $A$ and $f$. For better proof, we’ll apply sequence-of-game approach. We can obtain that $f = \langle \vec{v} \rangle_2$ and $\vec{v} \leftarrow \text{dbl}(v)$ are relevant with $v$. But $v = \text{bs}_1 + e_2$ are relevant with secret $s_1$. If $f$ reveal information about secret $s_1$, the difficulty of search-RLWE assumption will be solved. But this conclusion contradicts hypothesis about related lattice theory. Next, we’ll apply sequence-of-game approach to prove. We now give a proof of receiver’s security under the search-RLWE and decision-RLWE assumptions by applying sequence-of-game approach.

We mainly constructs a sequence of games, $\text{Game}_0$, $\text{Game}_1$, $\text{Game}_2$.

**Game**: $\text{Game}_0 = \text{Game}_0$, we denote $\text{Game}_0$ for better understanding the process of receiver’s security. $\text{Game}_0$ is the original attack game corresponding to given adversary and challenger in other games. We should notice that $(a, b)$ and $(b, v)$ are RLWE pairs in $\text{Game}_0$. We define the event $S_1$ which successfully output $\sigma^*$ and $\sigma^*$ equals to selected bit $\sigma$ in $\text{Game}_1$. The probability of success of event $S_1$ is defined as $\Pr[S_1]$.

In $\text{Game}_0$, we can obtain that the minimum success probability of adversary is $|\Pr[S_0] - 1/2|$. We define this probability as $\text{Adv}_{0}$. And we can get $\text{Adv}_{0} \leq |\Pr[S_0] - 1/2|$.

**Game**: Considering participants adversary and challenger, challenger sends $b$, $t$ ($t \leftarrow U(\mathbb{R}_q)$) to adversary corresponding to send $b = \text{as} + e$ in $\text{Game}_0$.

In $\text{Game}_1$, we can obtain that the minimum success probability of adversary is $|\Pr[S_1] - 1/2|$. We define this probability as $\text{Adv}_{1}$. And we can get $\text{Adv}_{1} \leq |\Pr[S_1] - 1/2|$.

**Game**: Considering participants adversary and challenger, challenger sends $u, v(u, v \leftarrow U(\mathbb{R}_q))$ to adversary corresponding to send $u = \text{as}_1 + e_1, v = \text{bs}_1 + e_2$ in $\text{Game}_1$.

In $\text{Game}_2$, we can obtain that the minimum success probability of adversary is $|\Pr[S_2] - 1/2|$. We define this probability as $\text{Adv}_{2}$. And we can get $\text{Adv}_{2} \leq |\Pr[S_2] - 1/2|$.

Considering that $(a, b)$ is RLWE pairs in $\text{Game}_0$ and $(a, b)$ is random pairs from uniform distribution in $\text{Game}_1$. Considering Decision-RLWE’s assumption and difference between $\text{Game}_0$ and $\text{Game}_1$, Considering that there is a reduction algorithm $R2$.

In reduction algorithm, when the input of $b$ is RLWE’s pair, the output is $\text{Game}_0$’s output, and when the input of $b$ is from uniformly distribution, the output is $\text{Game}_1$’s output.

We define $\text{Adv}_{R2}$ as adversary succeeding in reduction algorithm $R2$. The probability $\text{Adv}_{R2}$ of succeeding in distinguishing RLWE’s distribution and uniform distribution.

We can obtain that the probability of succeeding in $R2$ is negligible in reduction algorithm $R2$ due to the hardness of Decision-RLWE’s assumption. So we can obtain $\text{Adv}_{R2} \leq |\Pr[S_1] - \Pr[S_0]|$. Considering indistinguishability of RLWE’s distribution and uniform distribution, we can obtain that $\text{Adv}_{R2}$ is negligible.

**Table 7. Main difference between $\text{Game}_0$ and $\text{Game}_1$.**

| $\text{Game}_0$ | $\text{Game}_1$ |
|-----------------|-----------------|
| $s, e \leftarrow \chi$ | $s, e \leftarrow \chi$ |
| $b = \text{as} + e$ | $b = \text{as} + e$ |
| $t \in U(\mathbb{R}_q)$ | $t \in U(\mathbb{R}_q)$ |
| $s_1, e_1, e_2 \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $u = \text{as}_1 + e_1$ | $u = \text{as}_1 + e_1$ |
| $v = \text{bs}_1 + e_2$ | $v = \text{bs}_1 + e_2$ |
| $v \leftarrow \text{dbl}(v)$ | $v \leftarrow \text{dbl}(v)$ |
| $f = \langle \vec{v} \rangle_2$ | $f = \langle \vec{v} \rangle_2$ |
| $\Lambda \leftarrow (u + \sigma t) \oplus b$ | $\Lambda \leftarrow (u + \sigma t) \oplus b$ |

**Table 8. $\text{Game}_1$.**

| $\text{Challenger}$ | $\text{Adversary}$ |
|---------------------|---------------------|
| $s, e \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $b \leftarrow U(\mathbb{R}_q)$ | $b, t$ |
| $t \in U(\mathbb{R}_q)$ | $t \leftarrow U(\mathbb{R}_q)$ |
| $s_1, e_1, e_2 \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $u = \text{as}_1 + e_1$ | $u = \text{as}_1 + e_1$ |
| $v = \text{bs}_1 + e_2$ | $v = \text{bs}_1 + e_2$ |
| $v \leftarrow \text{dbl}(v)$ | $v \leftarrow \text{dbl}(v)$ |
| $f = \langle \vec{v} \rangle_2$ | $f = \langle \vec{v} \rangle_2$ |
| $\Lambda \leftarrow (u + \sigma t) \oplus b$ | $\Lambda \leftarrow (u + \sigma t) \oplus b$ |

**Table 9. Main difference between $\text{Game}_0$ and $\text{Game}_1$.**

| $\text{Game}_0$ | $\text{Game}_1$ |
|-----------------|-----------------|
| $s, e \leftarrow \chi$ | $s, e \leftarrow \chi$ |
| $b = \text{as} + e$ | $b \leftarrow U(\mathbb{R}_q)$ |
| $t \in U(\mathbb{R}_q)$ | $t \in U(\mathbb{R}_q)$ |
| $s_1, e_1, e_2 \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $u = \text{as}_1 + e_1$ | $u = \text{as}_1 + e_1$ |
| $v = \text{bs}_1 + e_2$ | $v = \text{bs}_1 + e_2$ |
| $v \leftarrow \text{dbl}(v)$ | $v \leftarrow \text{dbl}(v)$ |
| $f = \langle \vec{v} \rangle_2$ | $f = \langle \vec{v} \rangle_2$ |
| $\Lambda \leftarrow (u + \sigma t) \oplus b$ | $\Lambda \leftarrow (u + \sigma t) \oplus b$ |
algorithm $R_3$. In reduction algorithm, when the input of $u$ and $v$ are RLWE’s pair, the output is $Game_1$’s output, and when the input of $u$ and $v$ are from uniformly distribution, the output is $Game_2$’s output. We define $Adv_{R3}$ as adversary succeeding in reduction algorithm $R_3$. The probability $Adv_{R3}$ is the probability of succeeding in distinguishing RLWE’s distribution from uniform distribution.

Due to the hardness of Decision-RLWE’s assumption, we can obtain that the probability of succeeding in $R_3$ is negligible in reduction algorithm $R_3$. So we can obtain $Adv_{R3} \leq [Pr[S_2] - Pr[S_1]]$. Considering indistinguishability of RLWE’s distribution from uniform distribution, we can obtain that $Adv_{R3}$ is negligible.

So we have proven that selected bit $\sigma$ can’t be revealed by $A$.

Secondly, we’ll prove that $A$ can’t reveal the information of selected bit.

For better understand, let $A = A_0, \sigma \in \{0, 1\}, A_0 = u \oplus b, A_1 = (u + t) \oplus b$. Considering that $(a, u) (u = a\sigma + e_1)$ is RLWE’s pair, we can obtain that $u$ and $u + t (t \in U(\mathbb{Z}_q))$ is indistinguishable. Due to RLWE’s assumption, we can obtain that $A_0$ and $A_1$ is indistinguishable from the view of sender. We can similarly obtain that $W_0$ and $W_1$ in indistinguishable from the view of sender. So we have proven that selected bit $\sigma$ can’t be revealed by $A$.

We have completed the proof of receiver’s security in the above.

### B. KEY TRANSFER AND ENC PHASE

To better understand key transfer phase, we add the phase which receiver sends $A, f$ to sender. For conveniently understanding the process of getting initial key, we add introduction in process of sending $A$ and $f$ to sender.

We plug $A = (u + \sigma t) \oplus b$ and $W_0 = A \oplus b, W_1 = A \oplus b - t$ into the formula for $K_{sk_0} = \text{rec}(2W_0s, f), K_{sk_1} = \text{rec}(2W_1s, f)$. We mainly introduce two situation in indistinguishability. In one case, when $\sigma = 0$, we can obtain $K_{sk_0} = \text{rec}(2W_0s, f) = \text{rec}(2us, f)$ and $K_{sk_1} = \text{rec}(2W_1s, f) = \text{rec}(2(u - t)s, f)$. Considering that $u$ and $u - t$ is indistinguishable, and properties of reconciliation function, we can obtain that $K_{sk_0}$ and $K_{sk_1}$ are indistinguishable.

In another case, when $\sigma = 1$, we can obtain $K_{sk_0} = \text{rec}(2W_0s, f) = \text{rec}(2(u + t)s, f)$ and $K_{sk_1} = \text{rec}(2W_1s, f) = \text{rec}(2us, f)$. Considering that $u + t$ and $u$ is indistinguishable, and properties of reconciliation function, we can obtain that $K_{sk_0}$ and $K_{sk_1}$ are indistinguishable.

Sender sends $b, t$ to receiver, and receiver sends $A, f$ to sender. Then both sender and receiver can obtain same initial key. Considering that RLWE’s pairs are indistinguishable with random values, combined reconciliation function with modular rounding function, we can obtain initial key. This progress will not reveal any information. Receiver sends $A, f$ to sender, and sender applys $K_{sk_0} = \text{rec}(2W_0s, f), K_{sk_1} = \text{rec}(2W_1s, f)$ to get initial key. Receiver doesn’t know secret s from sender, so receiver will not know the value of $K_{sk_0}$. But receiver can obtain initial key $K_{sk}$ with the help of modular rounding function. Receiver makes use of modular rounding function based on $\bar{v}$. Considering that $\bar{v} = 2v - \bar{e}$ and $\bar{e}$ is uniformly distributed, sender can’t reveal any information about $\bar{v}$.

From the point of view of receiver, receiver can give necessary information for sender to obtain initial key without

| Table 10. $Game_2$. |
|---------------------|
| Challenger          | Adversary |
| $s, e \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $b \leftarrow U(\mathbb{Z}_q)$ | $b \leftarrow U(\mathbb{Z}_q)$ |
| $t \in U(\mathbb{Z}_q)$ | $t \in U(\mathbb{Z}_q)$ |
| $s_1, e_1, e_2 \leftarrow \chi$ | $s_1, e_1, e_2 \leftarrow \chi$ |
| $u = a\sigma_1 + e_1$ | $u \leftarrow U(\mathbb{Z}_q)$ |
| $v = b_{s_2} + e_2$ | $v \leftarrow U(\mathbb{Z}_q)$ |
| $\bar{v} \leftarrow \text{db}(v)$ | $\bar{v} \leftarrow \text{db}(v)$ |
| $f = \langle v \rangle_2$ | $f = \langle v \rangle_2$ |
| $A \leftarrow (u + \sigma t) \oplus b$ | $A \leftarrow (u + \sigma t) \oplus b$ |

| Table 11. Main difference between $Game_1$ and $Game_2$. |
|---------------------------------------------|
| $Game_1$                                   | $Game_2$                                   |
| $s, e \leftarrow \chi$                     | $s, e \leftarrow \chi$                     |
| $b \leftarrow U(\mathbb{Z}_q)$             | $b \leftarrow U(\mathbb{Z}_q)$             |
| $s_1, e_1, e_2 \leftarrow \chi$            | $s_1, e_1, e_2 \leftarrow \chi$            |
| $u \leftarrow U(\mathbb{Z}_q)$             | $u \leftarrow U(\mathbb{Z}_q)$             |
| $v \leftarrow U(\mathbb{Z}_q)$             | $v \leftarrow U(\mathbb{Z}_q)$             |
| $\bar{v} \leftarrow \text{db}(v)$         | $\bar{v} \leftarrow \text{db}(v)$         |
| $f = \langle v \rangle_2$                   | $f = \langle v \rangle_2$                   |
| $A \leftarrow (u + \sigma t) \oplus b$      | $A \leftarrow (u + \sigma t) \oplus b$      |

| Table 12. Key transfer and enc phase. |
|--------------------------------------|
| **Sender**                           | **Receiver**                               |
| $A$ if $\sigma = 0, A \leftarrow u \oplus b$ | $K_{sk} = \langle k \rangle_2$ |
| $A$ if $\sigma = 1, A \leftarrow (u + t) \oplus b$ | $Y_0 = \text{Commit}(K_{sk_0}, t)$ |
| $K_{sk} = \text{SPHF}(K_{sk_0}, r)$ | $Y_1 = \text{Commit}(K_{sk_1}, t)$ |
| $Y = \text{Commit}(K_{sk}, t)$ | $Y = \text{Commit}(K_{sk}, t)$ |
| $\text{Verify}(Y_0, Y_1, Y) = \beta, \beta \in \{0, 1\}$ | $\text{Verify}(Y_0, Y_1, Y) = \beta, \beta \in \{0, 1\}$ |
leaking information. This also reflects zero knowledge property. If sender cheat and give deceptive b, t to receiver, receiver will give corresponding A and f to sender, this will result in termination and can’t obtain same initial key. This also reflects soundness property. If sender and receiver honestly run protocol, both of them will obtain same final key. This also reflects completeness property.

Considering the indistinguishability between RLWE’s pairs’ distribution and uniform distribution, we combine this indistinguishability with special properties of SPHF function to obtain two indistinguishable keys. Due to the projection and smoothness properties of Smooth Projective Hash Function, we can obtain that $K_0 = H(K_{sk_0}, r)$ and $K_1 = H(K_{sk_1}, r)$ are indistinguishable.

Now let’s think about bit Enc phase. Sender encrypts $M_0, M_1$ to receiver. Sender sends $C_0 = Enc(K_0, M_0)$ and $C_1 = Enc(K_1, M_1)$ to receiver, which based on two different keys $K_0, K_1$. Receiver can use Dec algorithms to get $M_\sigma = Dec(K_\sigma, C_\sigma)$, which receiver only decrypts $C_\sigma$.

If sender and receiver honestly run whole protocol, receiver will obtain correct $M_\sigma$. In Enc phase, receiver has no information about $\sigma$. In Dec phase, receiver only obtain the value of $M_\sigma$, and has no information about $M_{1-\sigma}$.

C. SIMULATION

Protocol’s UC security is on the condition that malicious adversary attacks is securely authenticated in static corruption and channels between participants. SPHF hash function can be simulated by standard oracle, which used in simulation process proof.

We make use of an ideal function for proving security of protocols. In real world, participant $P_1$ which is corrupted by adversary $A$, interact with other parties based on designed protocols $P$. In ideal world, ideal adversary interacts with other parties based on ideal function. We mainly consider simulation between oblivious transfer protocols and ideal function $F_{OT}^\star$.

Considering interaction between sender $P_1$ and receiver $P_2$ in real and ideal world, combine with adversary $A$ in real world with simulator $S$ in ideal world. Invoking environment $Z$ used to distinguish ideal world from real world, define view in ideal and real world as $IDEAL_{F_{OT},S,Z}$ and $EXEC_{P,A,Z}$.

Let $F_{OT}$ be a ideal function used to implement OT, $P$ is an OT protocol, $A$ is any adversary in real world and $S$ is simulator in ideal world. If for any environments $Z$, obtain $IDEAL_{F_{OT},S,Z} \approx EXEC_{P,A,Z}$ (meaning computationally indistinguishable), protocol $P$ will obtain UC’s security.

We consider four cases. Firstly, adversary corrupts the sender. Secondly, adversary corrupts the receiver. Thirdly, adversary corrupts the sender and the receiver. Fourthly, neither the sender nor the receiver is corrupted by adversary.

Considering designed protocols’ security in universal composable model, we need construct a simulator in ideal environment.

1) WHEN SENDER IS CORRUPTED

When sender $P_1$ is corrupted by adversary $A$, define this corrupted sender as $P_1^\star$. We have a task that interaction between $P_1^\star$ and $P_2$ in real world is indistinguishable with interaction between $S$ and $P_2$ in ideal world for any environment $Z$.

Firstly, simulator $S$ extracts $P_1^\star$’s input ($M_0, M_1$), sends ($M_0, M_1$) to ideal function $F_{OT}$.

Simulator $S$ knows $P_1^\star$’s queries about hash function SPHF. Considering that SPHF has projection property, simulator $S$ can make use of stored key to decrypt ciphertext.

Secondly, we construct a simulator $S$ as follow:

1. Before interaction between $P_1^\star$ and $S$, simulator $S$ can reply standard oracle’s inquiry.

2. $P_1^\star$ sends $b$ and $t$ to simulator $S$.

3. Simulator $S$ chooses $s_3, e_3$ from uniform distribution, computes $u = as_3 + e_3$, $v = bs_3 + e_2$, $\bar{v} \leftarrow dbl(v)$, $f = (\bar{v})_2$ and sends $A, f$ to $P_1^\star$. Simulator $S$ makes two commitment for $K_{sk_i}, i \in \{0, 1\}$. And $S$ sends two commitment value $Y_i$ to $P_2$: $P_1^\star$ answers $K_i = SPHF(K_{sk_i}, r)$ to obtain $K_i, i \in \{0, 1\}$. Simulator $S$ stores ($K_{sk_i}, K_i$) to decrypt $P_1^\star$’s ciphertext.

4. $P_1^\star$ outputs ($C_0, C_1$) to simulator $S$, then $S$ use Dec($K_i, C_i$) to obtain $M_i$, otherwise outputs nothing and halts.

5. Simulator $S$ outputs ($M_0, M_1$) to ideal function $F_{OT}$.

From the view of any environment $Z$, combined with simulator $S$, we need to prove the indistinguishability between $IDEAL_{F_{OT},S,Z}$ and $EXEC_{P,A,Z}$.

$P_1^\star$ sends ($b, t$) to simulator $S$, then simulator $S$ simulates receiver’s input ($A, f$) to $P_1^\star$. Considering receiver’s security in above part, ($A, f$) generated from simulator $S$ is indistinguishable with ($A, f$) generated from receiver $P_2$.

Due to RLWE’s pairs’ characteristic of pseudo-uniform random distribution, obtain $IDEAL_{F_{OT},S,Z} \approx EXEC_{P,A,Z}$ (computationally indistinguishable).

2) WHEN RECEIVER IS CORRUPTED

When receiver $P_2$ is corrupted by adversary $A$, define this corrupted receiver as $P_2^\star$. We have a task that interaction between $P_1$ and $P_2^\star$ in real world is indistinguishable with interaction between $P_1$ and $S$ in ideal world for any environment $Z$.

Firstly, simulator $S$ interacts with $P_2^\star$, $S$ sends $b = as + e$ and $t$ selected from distribution to $P_2^\star$ for extracting $P_2^\star$’s input selected bit $i$. Once simulator $S$ is activated in ideal world, simulator $S$ will send $i$ to ideal function $F_{OT}$ for obtaining $M_i$. Secondly, simulator $S$ who act as sender $P_1$ sends ($C_0, C_1$) to $P_2^\star$.

Considering that simulator $S$ knows $P_3^\star$’s queries about reconciliation $F_{OT}$ and hash function SPHF, simulator can extract corresponding selected bit $i$.

Thirdly, we construct a simulator $S$ as follow:

1. Before interaction between $P_2^\star$ and $S$, simulator $S$ can reply standard oracle’s inquiry.

2. Simulator $S$ computes $b = as + e(s, e$ selected from uniform distribution).
3. Simulator $S$ receives $A$ and $f$ from corrupted receiver $P_2^*$. After receiving $(A, f)$ from $P_2^*$, simulator $S$ invokes reconciliation function $K_{sk_i} = rec(2W(s, f))$ according to designed protocol $P$. $S$ verifies the commitment from sender $P_1$. When algorithm Verify outputs $\beta = 1$, protocol continues. Simulator $S$ checks if $K_{sk_i} = rec(2W(s, f))$ has been queried by hash function SPHF. Then simulator $S$ can obtain corresponding selected bit $i \in \{0, 1\}$.

5. Simulator $S$ sends $(C_0, C_1)$ to $P_2^*$ based on algorithm $C_0 = Enc(K_{0}, M_0)$. Corresponding $(C_0, C_1)$ from simulator $S$, and $(C_0, C_1)$ from sender $P_1$, ciphertext pairs $(C_0, C_1)$ which are simulator $S$ and sender $P_1$ are indistinguishable. $P_2^*$ has access to query random oracle to obtain final key $K_i, i \in \{0, 1\}$. This means that $P_2^*$ also invokes reconciliation function to obtain initial key’s value of $K_{sk_i}, i \in \{0, 1\}$ ($K_{sk_i} = rec(2W(s, f))$).

But this result contradicts s-RLWE’s assumption. Sender $P_1$ has a secret ‘$s$’ which is used in reconciliation function $rec()$. And $P_2^*$ has no information about secret ‘$s$’ from sender $P_1$. So $P_2^*$ can’t obtain corresponding key.

Considering that RLWE’s distribution is indistinguishable with uniform random distribution, it’s indistinguishable between $(b, t)$ from sender $P_1$ and $(b, t)$ from simulator $S$.

We can obtain $IDEAL_{for, S, Z} \approx EXEC_{P, A, Z}$ (computationally indistinguishable).

3) WHEN BOTH SENDER AND RECEIVER ARE CORRUPTED
When both parties are corrupted, considering the above two situations, simulator $S$ only needs to simulate copies information generated from adversary’s attack. Generally speaking, the combination of the above two corruptions is relatively simple to understand.

4) WHEN NEITHER SENDER NOR RECEIVER IS CORRUPTED
Communication channel between sender and receiver is authenticated, public, but not secret. Adversary $A$ can observe the information interacted between two parties in communication channel. We can construct a simulator $S$ which can simulate $(b, t)$, $(A, f)$, $(Y_0, Y_1, t)$, $(SPHF, r)$ and $(C_0, C_1)$.

We construct a simulator $S$ as follow:

1. $S$ samples $s, e, s_1, e_1, e_2, t, r$ from uniform distribution and computes $b = as + e$. $S$ sends $(b, t)$ to $P_2$.
2. $S$ computes $u = as_1 + e_1, v = bs_1 + e_2, \tilde{v} = dbl(v), f = (\tilde{v})_2, A = (u + it) \oplus b, S$ sends $(A, f)$ to $P_1$.
3. $S$ computes $K_{sk_i} = rec(2A \oplus b - it)s, f), i \in \{0, 1\}$. Sender makes commitments for $K_{sk_i}$, and receiver verify value of commitment. When algorithm Verify outputs $\beta = 1$, $S$ continues computes $K_i = SPHF(K_{sk_i}, r)$. Simulator $S$ sends $(SPHF, r)$ to $P_2$.
4. $S$ computes $C_i = Enc(K_i, M_i)$ to obtain $(C_0, C_1)$. And $S$ sends $(C_0, C_1)$ to $P_2$.

Due to indistinguishability between RLWE’s pairs’ distribution and uniform distribution, combine with smoothness property of SPHF hash function and binding property of Bit Commit, $(b, t)$, $(A, f)$, $(C_0, C_1)$ from real world is indistinguishable with ideal world by simulation. We can obtain $IDEAL_{for, S, Z} \approx EXEC_{P, A, Z}$ (computationally indistinguishable).

VI. CONCLUSION
Considering that post-quantum oblivious transfer in universal composable model, we mainly apply key exchange to obtain post-quantum OT. We also apply Smooth Projective Hash Function’s property for designed protocol’s security and can be effectively modeled standard oracle. Combined with Bit Commitment, obtain commitment which can only be verified by honest receiver. Considering SPH system’s application, Enc and Dec algorithm can be resistant to quantum attacks, which is more secure in Post-quantum Era. Combined with sequence of games and full-simulation, we conduct a series of security proof. In brief, Sender’s security can be obtained by S-RLWE’s assumption, which can make sure the privacy of message. Receiver’s security can be obtained by D-RLWE’s assumption, which can make sure the privacy of selected bit. We can also think about oblivious transfer based on NTRUEncrypt.

REFERENCES
[1] M. O. Rabin, “How to exchange secrets with oblivious transfer,” JACR Cryptol. ePrint Arch., vol. 2005, p. 187, Jun. 2005.
[2] S. Even, O. Goldreich, and A. Lempel, “A randomized protocol for signing contracts,” Commun. ACM, vol. 28, no. 6, pp. 637–647, Jun. 1985.
[3] J. Kilian, “Founding cryptography on oblivious transfer,” in Proc. 20th Annu. ACM Symp. Theory Comput. (STOC), 1988, pp. 20–31.
[4] M. Bellare and S. Micali, “Non-interactive oblivious transfer and applications,” in Proc. Conf. Theory Appl. Cryptol. New York, NY: Springer, 1989, pp. 547–557.
[5] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skibizewska, “Practical quantum oblivious transfer,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 1991, pp. 351–366.
[6] D. Beaver, “Precomputing oblivious transfer,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 1995, pp. 97–109.
[7] C. Crépeau, D. van de Graaf, and A. Tapp, “Committed oblivious transfer and private multi-party computation,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 1995, pp. 110–123.
[8] M. Naor and B. Pinkas, “Oblivious transfer with adaptive queries,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 1999, pp. 573–590.
[9] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan, “The relationship between public key encryption and oblivious transfer,” in Proc. 41st Annu. Symp. Found. Comput. Sci., Nov. 2000, pp. 325–335.
[10] M. Naor and B. Pinkas, “Efficient oblivious transfer protocols,” in Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms Soc. Ind. Appl. Math., 2001, pp. 448–457.
[11] W-G. Tseng, “Efficient 1-out-n oblivious transfer schemes,” in Proc. Int. Workshop Public Key Cryptogr. Berlin, Germany: Springer, 2002, pp. 159–171.
[12] Y. Mu, J. Zhang, and V. Varadharajan, “$m$ out of $n$ oblivious transfer,” in Proc. Australas. Conf. Inf. Secur. Privacy. Berlin, Germany: Springer, 2002, pp. 395–405.
[13] D. Beaver, “Correlated pseudorandomness and the complexity of private computations,” in Proc. 28th Annu. ACM Symp. Theory Comput. (STOC), 1996, pp. 479–488.
[14] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, “Extending oblivious transfer efficiently,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 2003, pp. 145–161.
[15] H. Lipmaa, “Verifiable homomorphic oblivious transfer and private equality test,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. Berlin, Germany: Springer, 2003, pp. 416–433.
[16] C. Crépeau, K. Morozov, and S. Wolf, “Efficient unconditional oblivious transfer from almost any noisy channel,” in Proc. Int. Conf. Secur. Commun. Netw. Berlin, Germany: Springer, 2004, pp. 47–59.

[17] W. Ogata and R. Sasakiara, “K out of n oblivious transfer without random oracles,” IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. 87, no. 1, pp. 147–151, 2004.

[18] J. A. Garay, P. MacKenzie, and K. Yang, “Efficient and universally composable oblivious transfer and applications,” in Proc. Theory Cryptoc. Conf. Berlin, Germany: Springer, 2004, pp. 297–316.

[19] H. F. Huang, C. C. Chang, and J. S. Yeh, “Enhancement of non-interactive oblivious transfer scheme,” in Proc. 4th Int. Conf. Inf. Manage. Sci., 2005, pp. 196–199.

[20] C. K. Chu and W. G. Tzeng, “Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries,” International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2005, pp. 172–183.

[21] D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen, “On robust oracles,’’ in Proc. Theory Cryptogr. Conf. Berlin, Germany: Springer, 2004, pp. 96–113.

[22] R. Cramer and V. Shoup, “Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption,” in Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2005, pp. 45–64.

[23] Y. T. Kalai, “Smooth projective hashing and two-message oblivious transfer,” in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2005, pp. 78–95.

[24] S. Wolf and J. Wullschleger, “Oblivious transfer is symmetric,” in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2006, pp. 222–232.

[25] M. Fischlin, “Universally composable oblivious transfer in the multi-party setting,” in Proc. Cryptographers’ Track RSA Conf. Berlin, Germany: Springer, 2006, pp. 332–349.

[26] H.-F. Huang and C.-C. Chang, “A new t-out-of-n oblivious transfer with low bandwidth,” Appl. Math. Sci., vol. 1, no. 7, pp. 311–320, 2007.

[27] M. Green and S. Hohenberger, “Blind identity-based encryption and simulatable oblivious transfer,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. Berlin, Germany: Springer, 2007, pp. 265–282.

[28] J. Camenisch, G. Neven, and A. Shelat, “Simulatable adaptive oblivious transfer,” in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2007, pp. 573–590.

[29] R. Meier, B. Przydatek, and J. Wullschleger, “Robust combiners for oblivious transfer,” in Proc. Theory Cryptoc. Conf. Berlin, Germany: Springer, 2007, pp. 404–418.

[30] M. Green and S. Hohenberger, “Universally composable adaptive oblivious transfer,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. Berlin, Germany: Springer, 2008, pp. 179–197.

[31] A. Y. Lindell, “Efficient fully-simulatable oblivious transfer,” in Proc. Cryptographers’ Track RSA Conf. Berlin, Germany: Springer, 2008, pp. 52–70.

[32] I. Haitner, “Semi-honest to malicious oblivious transfer—The black-box way,” in Proc. Theory Cryptoc. Conf. Berlin, Germany: Springer, 2008, pp. 412–426.

[33] B. Shankar, K. Sritharan, and C. P. Rangan, “Alternative protocols for generalized oblivious transfer,” in Proc. Int. Conf. Distr. Comput. Netw. Berlin, Germany: Springer, 2008, pp. 304–309.

[34] R. Dowsley, J. van de Graaf, J. Müller-Quade, and A. C. A. Nascimento, “Oblivious transfer based on the McEliece assumptions,” in Proc. Int. Conf. Theor. Secur. Berlin, Germany: Springer, 2008, pp. 107–117.

[35] A. Parakh, “Oblivious transfer based on key exchange,” Cryptologia, vol. 32, no. 1, pp. 37–44, Jan. 2008.

[36] C. Peikert, V. Vaikuntanathan, and B. Waters, “A framework for efficient and composable oblivious transfer,” in Proc. Annu. Int. Conf. Cryptol. Berlin, Germany: Springer, 2008, pp. 554–571.

[37] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, “More efficient oblivious transfer and extensions for faster secure computation,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), 2013, pp. 535–548.

[38] R. Canetti, “Universally composable security: A new paradigm for cryptographic protocols,” in Proc. 42nd IEEE Symp. Found. Comput. Sci., Oct. 2001, pp. 136–145.

[39] V. Guleria and R. Dutta, “Efficient adaptive oblivious transfer in UC framework,” in Proc. Int. Conf. Inf. Secur. Pract. Exper. Cham, Switzerland: Springer, 2014, pp. 271–286.

[40] B. David, R. Dowsley, and A. C. A. Nascimento, “Universally composable oblivious transfer based on a variant of LPN,” in Proc. Int. Conf. Cryptol. Netw. Secur. Cham, Switzerland: Springer, 2014, pp. 143–158.

[41] O. Blazy and C. Chevalier, “Generic construction of UC-secure oblivious transfer,” in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. Cham, Switzerland: Springer, 2015, pp. 65–86.

[42] C. Zhao, H. Jiang, X. Wei, Q. Xu, and M. Zhao, “Cut-and-choose bilateral oblivious transfer and its application,” in Proc. IEEE Trustcom/BigDataSE/ISPA, vol. 1, Aug. 2015, pp. 384–391.

[43] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, “More efficient oblivious transfer extensions with security for malicious adversaries,” in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2015, pp. 673–701.

[44] H. Jannati and B. Bahram, “An oblivious transfer protocol based on ElGamal encryption for preserving location privacy,” Wireless Pers. Commun., vol. 97, no. 2, pp. 3113–3123, Nov. 2017.

[45] Z. Li, C. Xiang, and C. Wang, “Oblivious transfer via lossy encryption from lattice-based cryptography,” Wireless Commun. Mobile Comput., vol. 18, no. 11, pp. 1–11, Sep. 2018.

[46] B. Li and D. Micciancio, “Equational security proofs of oblivious transfer based on the RLWE assumption,” IEEE Trans. Inf. Forensics Security, vol. 75, pp. 90–100, May 2019.
[66] L. Bu, R. Agrawal, H. Cheng, and M. A. Kinsy, “Post-quantum cryptographic hardware primitives,” 2019, arXiv:1903.03735. [Online]. Available: http://arxiv.org/abs/1903.03735

[67] N. Döttling, S. Garg, M. Hajiabadi, D. Masny, and D. Wichs, “Two-round oblivious transfer from CDH or LPN.” IACR Cryptol. ePrint Arch., Tech. Rep., 2019, p. 414.

[68] H. C. Ding, H. Jiang, J. Cai, C. G. Wang, J. Zou, and Q. L. Xu, “Research on key exchange protocol based on LWE,” in Proc. Comput. Intell. Secur., 2019, pp. 236–240.

[69] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” J. ACM, vol. 56, no. 6, pp. 1–40, Sep. 2009.

[70] W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Trans. Inf. Theory, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[71] C. Peikert, “Lattice cryptography for the Internet,” IACR Cryptol. ePrint Arch., Tech. Rep., 2019, pp. 197–219.

[72] R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte, “Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications,” in Proc. Annu. Int. Cryptol. Conf. Cham, Switzerland: Springer, 2019, pp. 147–175.

[73] J. Katz and V. Vaikuntanathan, “Smooth projective hashing and password-based authenticated key exchange from lattices,” in Proc. 15th Int. Conf. Theory Appl. Cryptol. Inf. Secur., Tokyo, Japan: Springer-Verlag, Dec. 2009, pp. 636–652.

DING HANGCHAO received the master’s degree from the School of Mathematics, Shandong University, Jinan, China, in 2016, where she is currently pursuing the Ph.D. degree in cyberspace security. Her research interests include information security and cryptography, especially post-quantum cryptography based on lattices. She is a member of CACR.

HAN JIANG received the master’s and Ph.D. degrees from the School of Computer Science and Technology, Shandong University, Jinan, China, in 2005 and 2008, respectively. He is currently an Associate Professor with Shandong University. His main interests include cryptography and information security, especially secure multi-party computation. He is a member of CACR.

DIQIULIANG XU received the master’s and Ph.D. degrees from Shandong University, Jinan, China, in 1985 and 1999, respectively. He is currently a Professor and a Ph.D. Supervisor with Shandong University, where he has been, since 1985. He is also a Syndic of the Chinese Association for Cryptologic Research. His main interests include public key cryptography and multi-party secure computation. He holds several Science Foundations and Key Program of China.