Hindawi
IET Information Security
Volume 2024, Article ID 5513292, 16 pages
https://doi.org/10.1049/2024/5513292

Research Article

Dual-Mode Encryption for UC-Secure String OT from Learning with Errors

Momeng Liu,[1,2] **Yupu Hu**,[3] **Qiqi Lai**,[2,4] **Shanshan Zhang**,[3,5] **Huiwen Jia**,[6] **Wen Gao**,[7] **and Baocang Wang**[3]

[1]*Shaanxi Key Laboratory of Clothing Intelligence, School of Computer Science, Xi'an Polytechnic University, Xi'an, China*
[2]*Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, China*
[3]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China*
[4]*School of Computer Science, Shaanxi Normal University, Xi'an, China*
[5]*School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji, China*
[6]*Key Laboratory of Information Security, School of Mathematics and Information Science, Guangzhou University, Guangzhou, China*
[7]*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, China*

Correspondence should be addressed to Qiqi Lai; laiqq@snnu.edu.cn

Received 5 June 2023; Revised 14 August 2023; Accepted 29 January 2024; Published 15 February 2024

Academic Editor: Naghmeh Moradpoor

Universal composability (UC) is a primary security flavor for designing oblivious transfer (OT) due to its advantage of arbitrary composition. However, the study of UC-secure OT over lattices is still far behind compared with constructions over prequantum assumptions. Relying on the learning with errors (LWE) assumption, Quach proposes a dual-mode encryption scheme (SCN'20) for deriving a two-round OT whose security is provably UC-secure in the common reference string (CRS) model. Due to its use of a randomized rounding function proposed by Benhamouda et al. (PKC'18), this OT can only be limited to transmitting single-bit messages. Therefore, conducting trivial repetitions of Quach's OT when transmitting multibit strings would be very costly. In this work, we put forward a modified dual-mode encryption cryptosystem under the decisional LWE assumption, from which we can derive a UC-secure string OT with both full-fledged dual-mode security and better efficiency on transmitting strings. The key technique we adopt is a key reconciliation scheme proposed by Jiang et al. (PKC'20), which is utilized to extend the single-bit symmetric encryption key (produced by the aforementioned rounding function) to a multibit case. Through a comprehensive performance analysis, we demonstrate that our proposal can indeed strike a balance between security and efficiency.

1. Introduction

The two-party computation primitive *oblivious transfer* (OT) was first introduced by Rabin [1] and acted as a fundamental cryptographic building block widely used in secure multiparty computation [2, 3]. In this scenario, the sender $\mathcal{S}$ takes two messages $(\mu_0, \mu_1) \in \{0, 1\}^{\ell}$ (where $\ell \geq 1$) as input and the receiver $\mathcal{R}$ takes a bit $b \in \{0, 1\}$ as his message choice, with requiring that $\mathcal{R}$ can only obtain the output μ_b in the end and remain oblivious to μ_{1-b}, while $\mathcal{S}$ is totally unaware of $\mathcal{R}$'s choice b.

Essentially, OT can be realized in a *two-round* way. $\mathcal{R}$ first generates and sends to $\mathcal{S}$ a public key embedded with a message choice b. $\mathcal{S}$ will use this public key to compute the other public key for, respectively, encrypting μ_0 and μ_1, and send back to $\mathcal{R}$ these two encryptions, where only μ_b can be exactly recovered by secret decryption key.

With security concerns, *universal composability* (UC) [4] is a powerful notion among different simulation-based security flavors, which offers strong security guarantees and efficiency benefits whenever the protocol is executed concurrently or by arbitrary compositions within some advanced protocols, especially in multiparty computation or the complex Internet environment.

At CRYPTO'08, a dual-mode encryption framework for UC-secure OT is introduced by Peikert et al. [5]. To our best

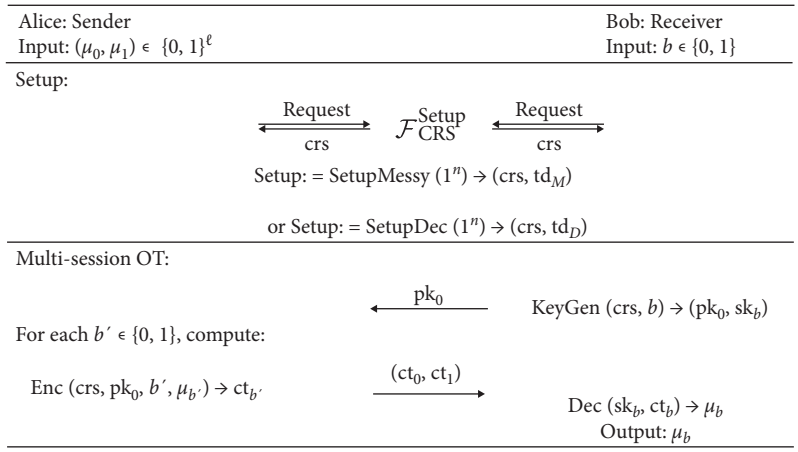

Figure 1: UC-secure OT from LWE.

knowledge, this is the optimal OT framework up to now, which not merely satisfies the succinct two-round paradigm with high efficiency but also achieves UC security under the *common reference string* (CRS) model against static corruptions (i.e., the corruption case is determined before the protocol execution without any modification during the course of protocol execution). They claim that this generic construction can provide statistical security for one specific party in each mode when generally realized under the decisional Diffie–Hellman assumption and the quadratic residuosity assumption. When it comes to the *learning with errors* (LWE) assumption, the receiver can only achieve computational security in either mode, and each CRS can be reused in limited sessions.

Targeting to solve this problem, an upgraded dual-mode encryption from LWE is proposed by Quach at SCN'20 [6], which rises the receiver's security to a statistical level and the reusability of each CRS to an unbounded case. In a nutshell, they utilize the noise flooding technique, requiring a superpolynomial LWE modulus q to promote the security of the receiver and the reusability of each CRS. However, such a use of superpolynomial modulus would directly contradict to a polynomial time $O(q)$ simulator for arguing sender's security in [5]. For addressing this issue, the work of [6] adopts a randomized rounding function R (with one-bit output) introduced by Benhamouda et al. [7] to make the public key messiness efficiently testable (applying lattice trapdoor techniques) and independent of the LWE modulus size.

However, since only one single-bit output from R is taken as an almost-uniform symmetric key to hide messages in the dual-mode encryption of [6], this further limits the derived UC-secure OT to transmit multibit strings. In addition, as mentioned in [7], the extension of R into a multibit output version is still an open question.

One may wonder that without costly trivial repetitions of this single-bit OT [6], *does a variant of dual-mode encryption over lattices for deriving UC-secure string OT exist, along with full-fledged dual-mode properties and unbounded reusability of CRS?*

Fortunately, our dual-mode encryption cryptosystem (see Section 3.2) provides an affirmative answer to this question.

1.1. Our Result. Based on the framework of [5], we propose an improved dual-mode encryption scheme [6] where it can directly derive a UC-secure OT (see Figure 1) for transmitting strings, as shown in Theorem 1.

Theorem 1 (informal). *Relying on the hardness of LWE with a subpolynomial modulus, a two-round UC-secure OT against static corruptions in the common reference string (CRS) model exists and satisfies the following properties:*

(1) *Each CRS can be instantiated in either messy or decryption mode, where the two modes are computationally indistinguishable.*
(2) *In messy mode, it can only provide the sender statistical security and the receiver computational security. In decryption mode, it can only provide the sender computational security and the receiver statistical security instead.*
(3) *Each CRS can be reused unbounded times for amortization between a fixed pair of participants.*
(4) *This UC-secure OT can transmit multibit strings while avoiding costly trivial repetitions of single-bit OT.*

1.2. Technical Overview. Our work can be viewed as an improvement of [6], and both works rely on the framework of [5]. For clarity, we first review the main technique adopted by Quach [6].

1.2.1. Technical Review of [6]. The work of [6] utilizes the noise flooding technique (requiring a superpolynomial size of LWE modulus) to upgrade the receiver security to a statistical level in decryption mode. However, it results in an inefficient simulator for arguing the sender's statistical security in messy mode. In particular, such a polynomial-time simulator for the sender security has to be completed in time $O(q)$, which directly conflicts with the use of a superpolynomial LWE modulus in noise flooding technique. Therefore, a failure happens in a polynomial-time simulator for arguing receiver's statistical security.

Table 1: Analysis on security.

OT protocol	[5]		[6]		This work	
	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{R}$
Security						
Messy	Statistical	Computational	Statistical	Computational	Statistical	Computational
Dec	Computational	Computational	Computational	Statistical	Computational	Statistical
Dual-mode properties	✗		✓		✓	

For addressing this issue, they follow the pattern of [8] and take the hash value output of an *approximate smooth projective hash* (ASPH) scheme [7] as a symmetric session key to encrypt the message. In a nutshell, an ASPH scheme operates on a set X and an NP-language $L \subseteq X$ by assuming the existence of a hard subset membership problem, i.e., it is hard to distinguish whether a random element is chosen from L or $X \setminus L$. For any $x \in L$, there exists a witness w such that the pair (x, w) satisfies a certain NP-relation. In addition, an ASPH scheme also involves a *hashing key* hk and a *projection key* hp. The *projection* property demands that the hash value, $\mathsf{H}(\mathsf{hk}, x)$, is determined by computing the projected hash value, $\mathsf{pH}(\mathsf{hp}, w)$, if $x \in L$. The *smoothness* property requires that for any $x \in X \setminus L$, $\mathsf{H}(\mathsf{hk}, x)$ is uniformly distributed even given hp and x.

In particular, the work of [6] utilizes a bit-ASPH [7], whose hash value is one single-bit output from a randomized rounding function $R: \mathbb{Z}_q \longrightarrow \{0, 1\}$ (see Section 2.3). Its OT execution mainly works as follows: Bob (the role of the receiver) first generates and sends to Alice (the role of the sender) his public key $(\mathbf{A}, \mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f})$, where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \longleftarrow \chi^m$, $\mathbf{f} \xleftarrow{\$} [-B', B']^m$. For all $i \in [N]$, Alice generates a *hashing key* $\mathsf{hk}_i = \mathbf{r}_i \longleftarrow D_{\mathbb{Z}, s}^m$ and a *projection key* $\mathsf{hp}_i = \mathbf{p}_i^T = \mathbf{r}_i^T \mathbf{A}$, and then computes the *hash value* $\mathsf{H}_i = R(\mathbf{r}_i^T \cdot \mathbf{c})$ to encrypt a one-bit message $\mu \in \{0, 1\}$ as $\beta_i = R(\mathbf{r}_i^T \cdot \mathbf{c}) \oplus \mu$. Alice sends (hp_i, β_i) to Bob. Then Bob computes the *projected hash value* $\mathsf{pH}_i = R(\mathbf{p}_i^T \cdot \mathbf{s})$ and $R(\mathbf{p}_i^T \cdot \mathbf{s}) \oplus \beta_i$ for decryption. If $\mathbf{c}$ is close to $\Lambda(\mathbf{A})$ (i.e., the q-arry lattice generated by $\mathbf{A}$), by the *approximate correctness* of R, we have $\mathsf{H}_i = R(\mathbf{r}_i^T \cdot \mathbf{c}) = R(\mathbf{p}_i^T \cdot \mathbf{s}) = \mathsf{pH}_i$ with high probability. Therefore, we have $\mu = R(\mathbf{p}_i^T \cdot \mathbf{s}) \oplus \beta_i$ with majority in all $i \in [N]$. Otherwise, by the *statistical smoothness* of R, the public key $(\mathbf{A}, \mathbf{c})$ is *messy* (see Section 3.1), and the distribution of β_i is statistically close to uniform. The approximate correctness of R guarantees that Bob can recover μ_b on the decryptable branch $b \in \{0, 1\}$, while the statistical smoothness of R provides the message-lossy property for μ_{1-b}, i.e., Bob is oblivious to μ_{1-b}.

In addition, this rounding function R offers a crucial property for arguing the simulation-based security of the sender. That is, given an appropriate trapdoor, public key messiness can be testable efficiently and independently of modulus q. It helps to complete the UC security proof for the derived OT in [6] and achieve all the properties of that well-defined dual-mode encryption (see Section 2.1) over lattices instead of a weaker instantiation proposed by Peikert et al. [5].

However, the work of [6] can only encrypt single-bit messages by the employment of that rounding function R, and a version of R with $\Theta(n)$-bit output is unresolved yet. In this work, we adopt a key reconciliation scheme introduced by Jiang et al. [9] to extend the single-bit symmetric key output by the bit-ASPH scheme [7] for a UC-secure string OT.

1.2.2. Extension of Symmetric Key. In essential, the work of [6] utilizes a KV09-type [10] ASPH scheme [7] to generate the symmetric keys $\mathsf{H}_i = R(\mathbf{r}_i^T \cdot \mathbf{c})$ and $\mathsf{pH}_i = R(\mathbf{p}_i^T \cdot \mathbf{s})$ for hiding and recovering messages, respectively. Recently, the work of [9] proposes two types (i.e., type-A and type-B) of ASPH over lattices (both are KV09-type) for building a password-based authenticated key exchange (PAKE) framework. They introduce a novel key reconciliation scheme to concatenate after the execution of type-B ASPH in the PAKE framework for agreeing on a shared secret key between two participants, i.e., extracting a random multibit shared key from two close hash value outputs of the type-B ASPH. For clarity, we denote this key reconciliation scheme as $\mathfrak{£} = (\mathfrak{£}_{\text{alice}}, \mathfrak{£}_{\text{bob}})$, which consists of two algorithms (i.e., $\mathfrak{£}_{\text{alice}}(d) \longrightarrow (\sigma, \xi_{\text{alice}})$ and $\mathfrak{£}_{\text{bob}}(\sigma, d') \longrightarrow \xi_{\text{bob}}$) and is executed between Alice and Bob as a one-message key reconciliation protocol (i.e., from Alice to Bob). Assume that $d \longleftarrow \mathbb{Z}_q$ (Alice's secret) and $d' \longleftarrow \mathbb{Z}_q$ (Bob's secret) satisfy the condition $|(d - d')_q| \leq \delta$ (where $(x)_q$ represents the residue of $x \in \mathbb{Z}_q$ over $[-q/2, q/2)$) for some integer $\delta \leq q/32$. After the execution of this protocol, both participants can agree on a common secret ξ, i.e., $\xi = \xi_{\text{alice}} = \xi_{\text{bob}}$ as the subsequent symmetric session key for encryption. Because the two hash values output by the type-B ASPH are actually $d = \mathbf{r}_i^T \cdot \mathbf{c}$ and $d' = \mathbf{p}_i^T \cdot \mathbf{s}$, which will be taken as input into $\mathfrak{£}$ sequentially. By observation, both d and d' are exactly taken as input into the rounding function R as well [7]. Therefore, we can utilize this key reconciliation mechanism $\mathfrak{£} = (\mathfrak{£}_{\text{alice}}, \mathfrak{£}_{\text{bob}})$ to extend the single-bit symmetric key output by R and encrypt a multibit message $\mu = (\mu_R, \mu_{\mathfrak{£}}) \in \{0, 1\}^{\ell}$ as follows:

$$\begin{bmatrix} \beta_i \\ \beta_{\mathfrak{£}} \end{bmatrix} \longleftarrow \begin{bmatrix} R(\mathbf{r}_i^T \cdot \mathbf{c}) \oplus \mu_R \\ \xi_{\text{alice}} \oplus \mu_{\mathfrak{£}} \end{bmatrix}, \tag{1}$$

where μ_R is the first bit of μ, and $\mu_{\mathfrak{£}}$ is the residual bits of μ. The correctness of decryption can be guaranteed by $\mathsf{H}_i = R(\mathbf{r}_i^T \cdot \mathbf{c}) = R(\mathbf{p}_i^T \cdot \mathbf{s}) = \mathsf{pH}_i$ with very large probability and $\xi_{\text{alice}} = \xi_{\text{bob}}$ with $|(d - d')_q| \leq \delta$ for $\delta \leq q/32$.

The approach we proposed above not only guarantees an efficient simulator for arguing sender's statistical security in the UC model by retaining the use of R but also solves the open problem for obliviously transferring multibit strings existing in [6]. Moreover, the adoption of $\pounds = (\pounds_{\text{alice}}, \pounds_{\text{bob}})$ is still compatible to public key messiness properties (see Lemma 9) when $\mathbf{c}$ is far away from $\Lambda(\mathbf{A})$. Therefore, our dual-mode encryption cryptosystem is a full-fledged instantiation over the lattice, which can exactly realize the well-defined primitive notion (see Definition 1).

1.3. Performance Analysis. We compare the security of our dual-mode encryption cryptosystem with another two related works (i.e., [5, 6]) in Table 1 to show that this work can fully achieve the dual-mode properties, as Definition 1 required.

Note that a multisession UC-secure OT (see Figure 1) can be derived from our proposed dual-mode cryptosystem, where crs can be reused unbounded times and multibit string transmitting is available in each single session. However, in the work of [5], crs can be simply reused in limited sessions. Moreover, in the work of [6], only single-bit message transmission is allowed in each single session instead of transferring multibit strings.

For a clear efficiency comparison on those three works, we illustrate some notations for clarity in Table 2. We let ℓ denote the bit-length of an encrypted message in each session and ℓ' denote the number of permitted sessions for a common crs. Then we mainly inspect the cost on vector sampling and the amortization performance during a multisession string OT execution (i.e., $\ell \geq 1$ and $\ell' \geq 1$). In particular, we analyze the cost on generating crs, pk and ct in each mode, respectively. The cost on generating crs is due to producing $\mathbf{A}$ and $\mathbf{v}$, the cost on generating pk is due to producing sk and error vector, and the cost on generating ct is due to randomness sampling.

Here, we let $a \cdot \widehat{g}$ denote the cost on running a times Gaussian sampling from $\mathbb{Z}^m$, and $b \cdot \widehat{u}$ denote the cost on running b times uniform sampling from $\mathbb{Z}_q^m$. For convenience, we treat the cost on sampling an n-dimensional vector as the same as that of sampling an m-dimensional vector according to some certain distribution. Since public matrix $\mathbf{A}$ in messy mode is produced by $\mathsf{TrapGen}(1^n, 1^m, q) \longrightarrow (\mathbf{A}, \mathbf{T})$ (see Lemma 4), we denote the cost on generating such a matrix $\mathbf{A}$ as $\widehat{trap_A}$. In addition, [6] and our work both use a heuristic randomized rounding function R (see Lemma 6), and we denote $\widehat{R}$ as the cost on sampling required randomness during each execution of R. Moreover, our scheme utilizes the key reconciliation mechanism $\pounds$ (see Section 2.4); we denote the cost on sampling a binary form integer $f = a_{t-1} \cdots a_{g+1} a_g \cdots a_1 a_0$ as $\widehat{f}$. Therefore, we can observe the comparison result from Table 3.

For transmitting strings by a multisession UC-secure OT between two fixed participants (e.g., Alice and Bob), the work of [5] can only reuse a common $\mathbf{A}$ during different (bounded) ℓ'-OT sessions (i.e., requiring ℓ' multiples of the cost on generating independent $\mathbf{v}$), and each session can obliviously transfer multibit messages (i.e., $\ell \geq 1$). Although the work of [6] can reuse a common crs during different (unbounded) ℓ'-OT sessions, due to the use of R, each session can only obliviously transfer single-bit messages (i.e., $\ell = 1$) and need $N = O(n)$ times independent randomness sampling for decryption correctness. Our work can also reuse a common crs during different (unbounded) ℓ'-OT sessions, but each session can obliviously transfer multibit strings (i.e., $\ell \geq 1$) with the additional price of sampling binary integer f. Therefore, the total costs on randomness sampling for encrypting $\{(\mu_0, \mu_1)_j \in \{0,1\}^{\ell}\}_{j \leq \ell'}$ in the above three works are $\ell' \cdot (2\widehat{u})$, $\ell' \cdot (2N(\widehat{u} + \widehat{R}))$, and $\ell' \cdot (2N(\widehat{u} + \widehat{R} + \widehat{f}))$, respectively.

Table 2: Notations for efficiency analysis.

Parameter	Description
ℓ	The bit-length of an encrypted message μ in each session
ℓ'	The number of permitted sessions for a common crs
n	Implicit security parameter $n \geq 1$
m	Lattice dimension $m \geq 2(n+1)\log q$
$\widehat{g}$	The cost on running one Gaussian sampling from $\mathbb{Z}^m$
$\widehat{u}$	The cost on running one uniform sampling from $\mathbb{Z}_q^m$
$\widehat{trap_A}$	The cost on generating a matrix $\mathbf{A}$ by $\mathsf{TrapGen}(1^n, 1^m, q) \longrightarrow (\mathbf{A}, \mathbf{T})$
$\widehat{R}$	The cost on sampling required randomness during each execution of R
$\widehat{f}$	The cost on sampling a binary form integer $f = a_{t-1} \cdots a_{g+1} a_g \cdots a_1 a_0$

Moreover, the communication cost in one OT execution is mainly on transmitting $(\mathsf{pk}, \{\mathsf{ct}_0, \mathsf{ct}_1\})$. Since the main difference of communication cost is on the ciphertext size, we conclude the bit-length of one single ciphertext (i.e., size) of these three works in Table 3. We observe that the work of [5] only needs to transmit $(n + \ell)\log q$ bits for the encryption of an ℓ-bit message, which is more efficient than our work for transferring strings. However, our work can achieve higher security and allow string OT via transmitting $N(n \log q + 1) + \log N + \log q + (\ell - 1)$ bits, instead of the work of [6] needs $N(n \log q + 1)$ bits for encrypting one single bit.

To sum up, if asking for higher efficiency but permitting lower security, the work of [5] would be recommended to use, since its costs on the randomness sampling and the ciphertext size are both less than the other two works. If it requires transferring multibit messages (i.e., $\ell > 1$) with full-fledged dual-mode security, we only need to run one session of our string OT, while the work of [6] has to run $\ell' = \ell$ single-bit OT sessions with huge overhead.

1.4. Other Related Work. The work of [11] builds a two-message OT protocol from LWE, which achieves statistical

Table 3: Analysis on efficiency.

OT protocol		[5]		[6]		This work	
		Messy	Dec	Messy	Dec	Messy	Dec
crs	$\mathbf{A}$	$\widehat{trap_A}$	$n \cdot \widehat{u}$	$\widehat{trap_A}$	$n \cdot \widehat{u}$	$\widehat{trap_A}$	$n \cdot \widehat{u}$
	$\mathbf{v}$	$\ell' \cdot (2\widehat{u})$	$\ell' \cdot (3\widehat{u} + 2\widehat{g})$	$\widehat{u}$	$\widehat{u} + \widehat{g}$	$\widehat{u}$	$\widehat{u} + \widehat{g}$
pk	sk	$\widehat{u}$	$\widehat{u}$	$\widehat{u}$	$\widehat{u}$	$\widehat{u}$	$\widehat{u}$
	Error	$\widehat{g}$	$\widehat{g}$	$\widehat{u} + \widehat{g}$	$\widehat{u} + \widehat{g}$	$\widehat{u} + \widehat{g}$	$\widehat{u} + \widehat{g}$
ct	Randomness	$\ell' \cdot (2\widehat{u})$	$\ell' \cdot (2\widehat{u})$	$\ell' \cdot (2N(\widehat{u} + \widehat{R}))$	$\ell' \cdot (2N(\widehat{u} + \widehat{R}))$	$\ell' \cdot (2N(\widehat{u} + \widehat{R} + \widehat{f}))$	$\ell' \cdot (2N(\widehat{u} + \widehat{R} + \widehat{f}))$
	Size	$(n + \ell')\log q$	$(n + \ell')\log q$	$N(n\log q + 1)$	$N(n\log q + 1)$	$N(n\log q + 1) + \log N + \log q + (\ell' - 1)$	$N(n\log q + 1) + \log N + \log q + (\ell' - 1)$

sender security and computational receiver security against malicious adversaries. For obliviously transferring multibit strings, although ours is less efficient (due to a superpolynomial modulus q) than their work, our scheme can obtain a stronger UC security at the expense of relying upon a trusted CRS.

In addition, the work of [12] proposes a generic construction to upgrade a two-round elementary OT to a UC-secure version in the malicious setting, where the CRS is reusable for unbounded times. By taking [5] or [11] as the elementary OT, we can obtain an LWE-based instantiation with a polynomial-size modulus. However, their work can only offer both participants computational security instead, and our proposal is more efficient by avoiding any non-black-box techniques.

Recently, the work of [13, 14] first introduced an LWE-based dual-mode non-interactive zero-knowledge proof (NIZK). We can take [5] as a semimalicious secure dual-mode OT into the framework of [13, 14] to derive a dual-mode OT with fully malicious security. Since [5] only achieves computational receiver security from LWE, if we fix this flaw with the noise flooding technique, the resulting issue would be the same as the problem in our scheme caused by the subexponential LWE modulus. Since the reductions for the soundness of [13, 14] are in a black-box way, it inherently implies the non-adaptively sound NIZKs of [13, 14] in statistical zero-knowledge mode. This can be patched up by complexity leveraging, but it would consequently lean upon the subexponential LWE hardness. Moreover, compiling the OT of [5] into the generic NIZKs framework of [13, 14] would result in practically inefficient proofs.

2. Preliminaries

2.1. Notations. Here, we take n as an implicit security parameter. We let $\mathsf{poly}(n)$ denote any function $f(n) = O(n^c)$ for some constant c, and $\mathsf{negl}(n)$ denote an unspecified function $f(n)$ such that $f(n) = n^{-\omega(1)}$. If a probability is $1 - \mathsf{negl}(n)$, we call it *overwhelming*.

We denote column vectors by bold lower cases and matrices by bold upper cases, e.g., $\mathbf{a}$ and $\mathbf{A}$. Their transposition operations are denoted by $\mathbf{a}^T$ and $\mathbf{A}^T$. We let $x \bmod q$ represent the residue of $x \in \mathbb{Z}_q$ over $[0, \ldots, q)$, and $(x)_q$ represent the residue of $x \in \mathbb{Z}_q$ over $[-q/2, q/2)$. The largest integer smaller than x and the smallest integer greater than x are, respectively, written by $\lfloor x \rfloor$ and $\lceil x \rceil$. We let $x \oplus y$ represent the xor operation between two bit strings $x, y \in \{0, 1\}^k$. All the distances $d(\cdot, \cdot)$ and norms $\|\cdot\|$ are in the ℓ_2 norm unless otherwise specified. Let $\|\cdot\|_\infty$ denote the infinity norm. For any positive integers $N \geq 1$, we let $[N]$ denote a set of integers $\{1, \ldots, N\}$.

We let $\mathcal{U}(E)$ represent the uniform distribution over a set E and $x \xleftarrow{\$} E$ represent the uniform sampling $x \leftarrow \mathcal{U}(E)$. We say a distribution ψ is *B-bounded* if the probability of sampling from ψ with the norm at most $B \in \mathbb{R}$ is overwhelming. The *statistical distance* between two distributions D_1 and D_2 is defined as $\Delta(D_1, D_2) = \frac{1}{2}\sum_x |\Pr_{D_1}(x) - \Pr_{D_2}(x)|$, where $\Pr_D(\cdot)$ is the probability mass function of D. We say that D_1 and D_2 are *statistically indistinguishable* if $\Delta(D_1, D_2) \leq \mathsf{negl}(n)$, denoted by $D_1 \approx_s D_2$. If for any probabilistic polynomial time distinguisher $\mathcal{A} \longrightarrow \{0, 1\}$ such that $|\Pr[\mathcal{A}(D_1) = 1] - \Pr[\mathcal{A}(D_2) = 1]| \leq \mathsf{negl}(n)$, we say that D_1 and D_2 are *computationally indistinguishable*, denoted by $D_1 \approx_c D_2$.

2.2. Dual-Mode Encryption. We first recall the notion of dual-mode encryption [5, 6]. For clarity, we adopt their notations for illustration.

Definition 1 (dual-mode encryption). A *dual-mode encryption* scheme with message space $\{0, 1\}^\ell$ consists of a bundle of probabilistic polynomial-time algorithms ($\mathsf{SetupMessy}$, $\mathsf{SetupDec}$, KeyGen, Enc, Dec, $\mathsf{FindMessy}$, $\mathsf{TrapKeyGen}$) defined as follows:

(1) $\mathsf{SetupMessy}(1^n) \longrightarrow (\mathsf{crs}, \mathsf{td}_M)$: Given as input the security parameter n, the setup algorithm outputs a common reference string crs along with a trapdoor td_M in messy mode.

(2) $\mathsf{SetupDec}(1^n) \longrightarrow (\mathsf{crs}, \mathsf{td}_D)$: Given as input the security parameter n, the setup algorithm outputs a common reference string crs along with a trapdoor td_D in decryption mode.

(3) $\mathsf{KeyGen}(\mathsf{crs}, b) \longrightarrow (\mathsf{pk}, \mathsf{sk}_b)$: Given as input a common reference string crs and a branch $b \in \{0, 1\}$, the key generation algorithm outputs a public encryption key pk and a secret decryption key sk_b for message encrypted on branch b.

(4) $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, b', \mu) \longrightarrow \mathsf{ct}$: Given as input a common reference string crs, a public key pk, a branch $b' \in \{0, 1\}$ and a message $\mu \in \{0, 1\}^\ell$, the encryption algorithm outputs a ciphertext ct on branch b'.

(5) $\mathsf{Dec}(\mathsf{sk}_b, \mathsf{ct}) \longrightarrow \mu$: Given as input a secret key sk_b and a ciphertext ct, the decryption algorithm outputs a message $\mu \in \{0, 1\}^\ell$.

(6) $\mathsf{FindMessy}(\mathsf{crs}, \mathsf{td}_M, \mathsf{pk}) \longrightarrow \tilde{b}$: Given as input a common reference string crs, a trapdoor in messy mode td_M and a (possibly malformed) public key pk, the algorithm outputs a branch $\tilde{b} \in \{0, 1\}$ corresponding to a messy branch of pk.

(7) $\mathsf{TrapKeyGen}(\mathsf{crs}, \mathsf{td}_D) \longrightarrow (\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$: Given as input a common reference string crs and a trapdoor in decryption mode td_D, the algorithm outputs keys $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$, where pk is a public encryption key, and sk_0 and sk_1 are corresponding secret decryption keys for branches 0 and 1, respectively.

The above dual-mode encryption is demanded to hold the following properties:

(1) Completeness on decryptable branch: For every $\mu \in \{0, 1\}^\ell$ and $b \in \{0, 1\}$, whether $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupMessy}(1^n)$ or $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupDec}(1^n)$ is executed in setup phrase, decryption is correct on branch b with overwhelming probability over the randomness of the entire experiment, i.e.,

$$\Pr[\mathsf{Dec}(\mathsf{sk}_b, \mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, b, \mu)) = \mu] \geq 1 - \mathsf{negl}(n), \tag{2}$$

where $(\mathsf{pk}, \mathsf{sk}_b) \leftarrow\!\!- \mathsf{KeyGen}(\mathsf{crs}, b)$.

(2) Indistinguishability of modes: For every $(\mathsf{crs}_M, \mathsf{td}_M) \leftarrow \mathsf{SetupMessy}(1^n)$ and $(\mathsf{crs}_D, \mathsf{td}_D) \leftarrow \mathsf{SetupDec}(1^n)$, both two crs outputs from two distinct setup algorithms are computationally indistinguishable, i.e.,

$$\mathsf{crs}_M \approx_c \mathsf{crs}_D. \tag{3}$$

(3) Security in messy mode: For all $(\mathsf{crs}, \mathsf{td}_M) \leftarrow\!\!- \mathsf{SetupMessy}(1^n)$ and (possibly malformed) pk, $\bar{b} \leftarrow\!\!- \mathsf{FindMessy}(\mathsf{crs}, \mathsf{td}_M, \mathsf{pk})$ implies that $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, \bar{b}, \cdot)$ is message-lossy (i.e., messy). That is, for all messages $\mu_0, \mu_1 \in \{0,\ 1\}^{\ell}$,

$$\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, \bar{b}, \mu_0) \approx_s \mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, \bar{b}, \mu_1). \tag{4}$$

(4) Security in decryption mode: For all $(\mathsf{crs}, \mathsf{td}_D) \leftarrow\!\!- \mathsf{SetupDec}(1^n)$, it holds that for every $b \in \{0, 1\}$,

$$(\mathsf{crs}, \mathsf{pk}, \mathsf{sk}_b) \approx_s (\mathsf{crs}, \mathsf{KeyGen}(\mathsf{crs}, b)), \tag{5}$$

where $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow\!\!- \mathsf{TrapKeyGen}(\mathsf{td}_D)$ for the left-hand side above.

The work of [5] showed that once a well-constructed dual-mode encryption scheme is completed as the above notion, a UC-secure OT can be directly obtained. Here, we suppose all readers know the UC security model well and omit to introduce its corresponding background here. We recommend to go to [5] for more details.

Theorem 2 (UC-secure OT from dual-mode encryption [5, 6]). *If a dual-mode encryption scheme* ($\mathsf{SetupMessy}$, $\mathsf{SetupDec}$, KeyGen, Enc, Dec, $\mathsf{FindMessy}$, $\mathsf{TrapKeyGen}$) *is well-defined as above, we can obtain a protocol to UC-realize the multisession OT functionality* $\widehat{\mathcal{F}}_{\mathrm{OT}}$ *in the* $\mathcal{F}_{\mathrm{CRS}}$*-hybrid model under static corruptions.*

We can execute this UC-secure OT protocol in either of two modes. Each time, it is run over a distinct functionality $\mathcal{F}_{\mathrm{CRS}}$ that produces crs according to the corresponding setup algorithm. The messy mode only provides statistical security for the sender. The decryption mode only provides statistical security for the receiver. The other party in each mode can only achieve computational security.

2.3. Lattices Background

2.3.1. Lattices and Gaussians. We first recall some basic knowledge regarding lattices.

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ consist of n linearly independent m-dimensional column vectors $\mathbf{b}_i \in \mathbb{R}^m$ for all $i \in [n]$. The m-dimensional lattice Λ generated by $\mathbf{B}$ is defined as $\Lambda(\mathbf{B}) = \{\mathbf{Bc} = \sum_{i\in[n]} c_i \cdot \mathbf{b}_i : \mathbf{c} \in \mathbb{Z}^n\}$. The *dual lattice* of Λ is defined as $\Lambda^* = \{\mathbf{y} \in \mathsf{Span}_{\mathbb{R}}(\Lambda) | \forall \mathbf{x} \in \Lambda, <\mathbf{x}, \mathbf{y}> \in \mathbb{Z}\}$. Let $\lambda_1^{\infty}(\Lambda) = \min_{\mathbf{x}\in\Lambda\setminus\{\mathbf{0}\}} \|\mathbf{x}\|_{\infty}$ define the *minimum distance* of a lattice in infinity norm. If the column vectors of a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ are linearly independent, we say that $\mathbf{A}$ is *full-rank*. Now we introduce two *q-ary lattices* defined by $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$:

$$\Lambda(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^m | \mathbf{x} = \mathbf{As} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^n\right\}, \tag{6}$$

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{r} \in \mathbb{Z}^m | \mathbf{r}^T\mathbf{A} = \mathbf{0}^T \bmod q\}. \tag{7}$$

These two lattices are dual to each other up to a scaling factor q such that $\Lambda(\mathbf{A}) = q \cdot \Lambda^{\perp}(\mathbf{A})^*$ and $\Lambda^{\perp}(\mathbf{A}) = q \cdot \Lambda(\mathbf{A})^*$.

We define the *Gaussian weight function* on $\mathbb{R}^m$ with parameter $\tau > 0$ as follows:

$$\forall \mathbf{x} \in \mathbb{R}^m,\ \rho_{\tau}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/\tau^2). \tag{8}$$

The *discrete Gaussian distribution* over $\mathbb{Z}$ with parameter $\tau > 0$ is defined as follows:

$$\forall x \in \mathbb{Z},\ D_{\mathbb{Z},\tau}(x) = \frac{\rho_{\tau}(x)}{\sum_{y\in\mathbb{Z}}\rho_{\tau}(y)}. \tag{9}$$

Moreover, we recall an important lattice parameter, i.e., the *smoothing parameter* [15]. For an m-dimensional lattice Λ and a positive real $\epsilon > 0$, the smoothing parameter $\eta_{\epsilon}(\Lambda)$ is defined as the smallest $\tau > 0$ such that $\rho_{1/\tau}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$.

Now we introduce some useful lemmas regarding the above q-ary lattices defined by $\mathbf{A}$ and the corresponding lattice quantity η_{ϵ}.

Lemma 1 (see [16] Lemma 5.2). *Suppose a matrix* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ *whose row vectors can generate* $\mathbb{Z}_q^n$ *(a.k.a.* $\mathbf{A}$ *is full-rank),* $\epsilon \in (0, \frac{1}{2})$ *and* $\tau \geq \eta_{\epsilon}(\Lambda^{\perp}(\mathbf{A}))$*. For any* $\mathbf{e} \leftarrow\!\!- D_{\mathbb{Z},\tau}^m$*, the distribution of* $\mathbf{u} = \mathbf{e}^T\mathbf{A} \bmod q$ *is close to the uniform distribution over* $\mathbb{Z}_q^n$ *within statistical distance* 2ϵ*.*

Lemma 2 (see [16] Lemmas 5.1 and 5.3). *Let n, m, and q be positive integers with q prime and* $m \geq 2n \log q$*. For all but an at most* q^{-n} *fraction of* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$*, the rows of* $\mathbf{A}$ *can generate* $\mathbb{Z}_q^n$*. For all but an at most* q^{-n} *fraction of* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$*, we have a large minimum distance* $\lambda_1^{\infty}(\Lambda(\mathbf{A})) \geq q/4$*. That is*

$$\Pr_{\mathbf{A} \xleftarrow{s} \mathbb{Z}_q^{m\times n}} [\mathbf{A} \text{ is full-rank} \wedge \lambda_1^{\infty}(\Lambda(\mathbf{A})) \geq q/4] \geq 1 - 2q^{-n}. \tag{10}$$

Lemma 3 (see [15–17]). *For any m-dimensional lattice* Λ *and positive real* $\epsilon > 0$*, we have the following:*

$$\eta_\epsilon(\Lambda) \le \frac{\sqrt{\log(2m/(1+1/\epsilon))/\pi}}{\lambda_1^\infty(\Lambda^*)}. \tag{11}$$

Let n, m, and q be positive integers with q prime and $m \ge 2n\log q$. *For any function* $\omega(\sqrt{\log m})$, *there is a negligible function* $\epsilon(m)$ *such that*

$$\eta_\epsilon(\Lambda^\perp(\mathbf{A})) \le \omega\left(\sqrt{\log m}\right), \tag{12}$$

with overwhelming probability over the choice of $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$.

2.3.2. LWE. We introduce the definition of (decisional) LWE problem.

Definition 2 ((decisional) LWE [18]). Let $n \ge 1$ and $q = q(n) \ge 2$ be positive integers, and χ denote an error distribution over $\mathbb{Z}$. The (decisional) LWE problem $\mathsf{LWE}_{q,\chi,n}$ states that for all $m = \mathsf{poly}(n)$ and some secret vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, the distribution $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ is computationally indistinguishable from the distribution of $(\mathbf{A}, \mathbf{b})$, i.e.,

$$(\mathbf{A}, \mathbf{As}+\mathbf{e}) \approx_c (\mathbf{A}, \mathbf{b}), \tag{13}$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{e} \longleftarrow \chi^m$, and $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^m$.

$\mathsf{LWE}_{q,\chi,n}$ hardness [18]. For all $B \ge \tilde{\Omega}(\sqrt{n})$, a B-bounded distribution $\chi = \chi(n)$ exists such that within approximation factor $\gamma = \tilde{O}(\sqrt{n}q/B)$, breaking the average case problem $\mathsf{LWE}_{q,\chi,n}$ is at least as hard as solving the worst case problems GapSVP_γ and SIVP_γ using a quantum algorithm.

2.3.3. Lattices Trapdoors. Now, we introduce a lemma regarding the lattice trapdoor technique, which is used to identify messy public keys for arguing the sender's statistical security in messy mode.

Lemma 4 (see [19] Theorem 5.1). *Given some integers* $n \ge 1$, $q \ge 2$, *and* $m \ge \Omega(n\log q)$ *as input, there exists an efficient randomized algorithm*

$\mathsf{TrapGen}(1^n, 1^m, q)$ *which outputs* $\mathbf{A} \in \mathbb{Z}^{m\times n}$ *along with a trapdoor* $\mathbf{T}$ *such that*

(1) *The distribution of* $\mathbf{A}$ *is statistically close to* $\mathcal{U}(\mathbb{Z}_q^{m\times n})$.
(2) *For any* $\mathbf{s} \in \mathbb{Z}_q^m$ *and* $\mathbf{e} \in \mathbb{Z}_q^m$ *such that* $\|\mathbf{e}\| < q/6\sqrt{m}$, *given* $\mathbf{c} = \mathbf{As}+\mathbf{e}$ *and the above* $(\mathbf{A}, \mathbf{T})$ *as input, an efficient deterministic trapdoor inversion algorithm which can output* $(\mathbf{s}, \mathbf{e})$ *exists, i.e.,* $\mathsf{Invert}(\mathbf{T}, \mathbf{A}, \mathbf{c}) \longrightarrow (\mathbf{s}, \mathbf{e})$.

2.3.4. Noise Flooding. The following lemma is used for arguing the receiver's statistical security in decryption mode.

Lemma 5 (see [9, 20]). *Suppose* $B = B(n)$ *and* $B' = B'(n) \in \mathbb{Z}$ *are two positive integers. Let* $e_1 \in [-B, B]$ *be a fixed integer and* $e_2 \xleftarrow{\$} [-B', B']$. *The distribution of* e_2 *is statistically close to the distribution of* $e_2 + e_1$ *as long as* $B/B' = \mathsf{negl}(n)$, *i.e.,*

$$\mathcal{U}([-B', B']) \approx_s \mathcal{U}([-B', B']) + e_1. \tag{14}$$

2.4. Statistically Smooth Rounding Function over Lattices. We still employ the statistically smooth rounding function [7] in our dual-mode encryption construction. It can provide a crucial property that identifying messy public key is simply running the trapdoor inversion algorithm once (independent of the superpolynomial LWE modulus q), which further helps to build an efficient simulator for arguing the sender's statistical security in the UC model.

Lemma 6 (see [6, 7]). *A randomized rounding function* $R: \mathbb{Z}_q \longrightarrow \{0, 1\}$ *is well-defined such that*

$$\Pr[R(x) = 1] = 1/2 + \frac{\cos(2\pi x/q)}{2}. \tag{15}$$

Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ *with* $m = \Theta(n\log q)$, $\mathbf{p} \in \mathbb{Z}_q^n$, *and* $\tau \ge \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ *for some* $\epsilon = \mathsf{negl}(n)$. *Then, the above randomized rounding function R satisfies the following properties:*

(1) *Statistical smoothness: If* $\mathbf{A}$ *is full-rank and for all* $\mathbf{c} \in \mathbb{Z}_q^m$ *with* $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge q\sqrt{m}/\tau$, *we have the following:*

$$\left| \Pr_{R, \mathbf{r} \longleftarrow D_{\mathbb{Z},\tau}^m} [R(\langle \mathbf{r}, \mathbf{c}\rangle) = 1 | \mathbf{r}^T\mathbf{A} = \mathbf{p}^T] - 1/2 \right| \le \mathsf{negl}(n), \tag{16}$$

where the probability is taken over $\mathbf{r} \longleftarrow D_{\mathbb{Z},\tau}^m$ *and the randomness of R.*

(2) *Approximate correctness: For all* $\mathbf{c} = \mathbf{As}+\mathbf{e} \in \mathbb{Z}_q^m$, *where* $\mathbf{s} \in \mathbb{Z}_q^n$ *and* $\mathbf{e} \in \mathbb{Z}_q^m$ *such that* $d(\mathbf{c}, \Lambda(\mathbf{A})) \le B$ *(i.e.,* $\|\mathbf{e}\| \le B$*) and* $B \cdot \tau \cdot \sqrt{m} = o(q)$, *then for all large enough n, we have the following:*

$$\Pr_{R, \mathbf{r} \longleftarrow D_{\mathbb{Z},\tau}^m} [R(\mathbf{r}^T\mathbf{As}) = R(\mathbf{r}^T\mathbf{c})] \ge 2/3. \tag{17}$$

2.5. Key Reconciliation over Lattices. Now, we recall the key reconciliation scheme introduced in [9], which can extract a random multibit shared key from two close secrets over $\mathbb{Z}_q$. We denote this scheme as $\mathfrak{L} = (\mathfrak{L}_{\text{alice}}, \mathfrak{L}_{\text{bob}})$, which consists of two algorithms and can be viewed as a one-message key reconciliation protocol sequentially executed from Alice to Bob. Assume $d \longleftarrow \mathbb{Z}_q$ (Alice's secret) and $d' \longleftarrow \mathbb{Z}_q$ (Bob's secret) with $|(d-d')_q| \le \delta$ for some integer $\delta \le q/32$. At the end of the execution, Alice and Bob could reach a consensus on a common secret ξ, i.e., $\xi = \xi_{\text{alice}} = \xi_{\text{bob}}$. Let $t = \lfloor \log q \rfloor$ and $g = \lceil \log\delta \rceil$. The scheme $\mathfrak{L}$ works as follows:

Alice's execution (a.k.a. $\mathfrak{L}_{\text{alice}}(d) \longrightarrow (\sigma, \xi_{\text{alice}})$):

Table 4: Parameters setting.

Parameter	Description	Setting
n	Implicit security parameter	$n \geq 1$
q	Superpolynomial prime modulus	$q = n^{\omega(1)} \geq 2$
m	Lattice dimension	$m \geq 2(n+1)\log q$
N	Number of Sampling	$N = N(n) = n$
τ	Gaussian parameter	$\tau \geq 4\sqrt{m}$
χ	B-bounded distribution	$\chi = \chi(n)$
B	Bound defined in Definition 2	$B = \tilde{\Omega}(\sqrt{n})$
B'	Bound defined in Lemma 5	$B' \in \mathbb{Z}$ s.t. $(B+B') \cdot \tau\sqrt{m} = o(q)$ and $B/B' = \mathsf{negl}(n)$
δ	Upper bound of the gap between two close inputs of $\pounds$	$\delta \leq q/32$
t	Length of f in binary form	$t = \lfloor \log q \rfloor$
g	Index of the bit set as 1 in generating binary f	$g = \lceil \log \delta \rceil$
ℓ	Bit-length of binary message μ	$\ell = t - g - 1$

(1) Alice sets an integer $f = a_{t-1} \cdots a_{g+1} a_g \cdots a_1 a_0$ in a binary form, where she defines $a_g = 1$ and $a_{g+1} = 0$, and takes $a_j \longleftarrow \{0, 1\}$ for $0 \leq j \leq t-1$ but $j \neq g, g+1$.

(2) Alice sets the common secret as $\xi_{\text{alice}} = (a_{t-1}, \ldots, a_{g+2})^T$ and sends $\sigma = (f + d) \bmod q$ to Bob.

Bob's execution (a.k.a. $\pounds_{\text{bob}}(\sigma, d') \longrightarrow \xi_{\text{bob}}$):

After receiving σ, Bob takes as input σ and d', and sets the common secret $\xi_{\text{bob}} = \lfloor \frac{(\sigma - d') \bmod q}{2^{g+2}} \rfloor$ in its binary form.

Lemma 7 (see [9]). *We assume $d, d' \in \mathbb{Z}_q$ with $|(d - d')_q| \leq \delta$, then Alice and Bob can agree on a common secret (i.e., $\xi = \xi_{\text{alice}} = \xi_{\text{bob}}$) after the execution of $\pounds = (\pounds_{\text{alice}}, \pounds_{\text{bob}})$. Furthermore, if $d \xleftarrow{\$} \mathbb{Z}_q$, the common key ξ is confidential (even given σ) and uniformly distributed over $\{0, 1\}^{(t-g-2)}$. The entropy $H(\xi) = H(\xi|\sigma)$ is at least as large as $\log \frac{q}{16\sigma}$.*

Remark 1. Note that f is independent of d, then d is the one-time pad for f by $\sigma = f + d \bmod q$. Hence, f is independent of σ. Furthermore, ξ is determined by the first $t - g - 2$ randomly chosen bits of f, then ξ is independent of σ and uniformly random. Therefore, we can use ξ as the one-time pad key to encrypt multiple bits in our dual-mode encryption scheme.

3. LWE-Based Dual-Mode Encryption for UC-Secure String OT

In this section, an LWE-based dual-mode encryption (see Section 3.2) is proposed for deriving a UC-secure string OT (as shown in Figure 1), which is more efficient than costly running multiple independent executions of single-bit OT [6] (see Table 3) for transmitting multibit messages. We first introduce its underlying LWE-based messy public-key encryption in Section 3.1, i.e., an extension scheme of the counterpart of [6].

3.1. Extended Messy Public-Key Encryption. For a lattice-based dual-mode encryption cryptosystem over multibit messages, we need an LWE-based messy public-key encryption as its underlying encryption algorithm, which is obtained by extending the messy public-key encryption of [6] to a multibit encryption version. In particular, we use the single-bit output of that statistically smooth rounding function R (see Lemma 6) to encrypt the first bit of the message, for retaining the property that messy public key can be testable efficiently and independently of the LWE modulus size. Moreover, we add the key reconciliation scheme $\pounds$ (see Section 2.4) into a framework. By taking one of R's inputs during its multiple executions (under the same public key) as the input of $\pounds$, we can obtain multiple random bits to hide the residue bits of the message. Since R and $\pounds$ both utilize the same public key (possibly malformed), the messy public key property is naturally inherent in our extended LWE-based encryption.

3.1.1. Parameters Setting. Consider the randomized rounding function R (see Lemma 6) and key reconciliation scheme $\pounds$ (see Section 2.4) together used in the scheme. We show all the parameters set in Table 4 to satisfy the correctness and security of the following LWE-based messy encryption scheme.

3.1.2. Construction. Now we show our extended LWE-based encryption scheme $(\mathsf{LWEKeyGen}, \mathsf{LWEEnc}, \mathsf{LWEDec})$ over message space $\mathcal{M} = \{0, 1\}^{\ell}$.

(1) $\mathsf{LWEKeyGen}(1^n)$: Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \longleftarrow \chi^m$, $\mathbf{f} \xleftarrow{\$} [-B', B']^m$ and set $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f}$. Output:

$$\mathsf{pk} = (\mathbf{A}, \mathbf{c}),\ \mathsf{sk} = \mathbf{s}. \tag{18}$$

(2) $\mathsf{LWEEnc}(\mathsf{pk}, \mu \in \{0,1\}^{\ell})$: For $i \in [N]$, doing as follows:

1. Sample $\mathbf{r}_i \longleftarrow D^m_{\mathbb{Z},\tau}$, and compute $\mathbf{p}_i^T = \mathbf{r}_i^T \cdot \mathbf{A} \in \mathbb{Z}_q^{1\times n}$.
2. Compute $(\sigma_k, \xi_{\text{alice}}) \longleftarrow \mathfrak{L}_{\text{alice}}(\mathbf{r}_k^T \cdot \mathbf{c})$, where $\mathbf{r}_k$ is set as any vector chosen from $\{\mathbf{r}_i\}_{i\in[N]}$ and $k \in [N]$.
3. Split the message μ into two segments, i.e., $\mu = (\mu_R, \mu_{\mathfrak{L}})$, where μ_R is the first bit of μ, and $\mu_{\mathfrak{L}}$ is the residual $\ell - 1$ bits of μ. Then compute:

$$\begin{bmatrix} \beta_i \\ \beta_{\mathfrak{L}} \end{bmatrix} \longleftarrow \begin{bmatrix} R(\mathbf{r}_i^T \cdot \mathbf{c}) \oplus \mu_R \\ \xi_{alice} \oplus \mu_{\mathfrak{L}} \end{bmatrix}. \tag{19}$$

4. Output the ciphertext ct:

$$\mathsf{ct} = \left(\{\mathbf{p}_i, \beta_i\}_{i\le N}, \{k, \sigma_k, \beta_{\mathfrak{L}}\}\right). \tag{20}$$

(3) $\mathsf{LWEDec}(\mathsf{sk}, \mathsf{ct})$: For all $i \in [N]$, doing as follows:

1. Compute $b_i \longleftarrow R(\mathbf{p}_i^T \cdot \mathbf{s}) \oplus \beta_i$, and set the majority bit of the b_i's as μ_R.
2. Compute $\xi_{\text{bob}} \longleftarrow \mathfrak{L}_{\text{bob}}(\sigma_k, \mathbf{p}_k^T \cdot \mathbf{s})$ and set $\mu_{\mathfrak{L}} = \xi_{\text{bob}} \oplus \beta_{\mathfrak{L}}$.
3. Output the message $\mu = (\mu_R, \mu_{\mathfrak{L}})$.

Similar to [6], the term $\mathbf{f}$ added into $\mathbf{c}$ (i.e., noise flooding) is used for arguing the receiver's statistical security in decryption mode. Note that it would not affect any of the following properties without this term in $\mathbf{c}$. Now, we show the correctness of this extended LWE-based encryption scheme.

Lemma 8 (correctness). *Let* $(B + B') \cdot \tau\sqrt{m} = o(q)$, $\tau \ge \omega(\sqrt{\log m})$, *and* $\delta \le q/32$, *then the above extended public-key encryption scheme is correct.*

Proof. By Lemma 3, $\epsilon = \mathsf{negl}(n)$ can be set such that $\tau \ge \eta_\epsilon(\Lambda^{\perp}(\mathbf{A}))$ with overwhelming probability over the choice of $\mathbf{A}$.

If we set $\|\mathbf{e}\| \le B$ and $\|\mathbf{f}\| \le B'$, by the approximate correctness of R (see Lemma 6), for all $i \in [N]$, we have

$$\Pr_{R, \mathbf{r}_i \longleftarrow D^m_{\mathbb{Z},\tau}}[b_i = \beta_i] \ge 2/3, \tag{21}$$

over the internal randomness of R and $\mathbf{r}_i \longleftarrow D^m_{\mathbb{Z},\tau}$. By Cauchy–Schwarz inequality, we have $|\mathbf{r}_i^T(\mathbf{e}+\mathbf{f})| \le \|\mathbf{r}_i\| \cdot \|\mathbf{e}+\mathbf{f}\| \le \tau\sqrt{m} \cdot (B + B') = o(q)$.

We can observe from the above that $d = \mathbf{r}_i^T\mathbf{c}$ and $d' = \mathbf{p}_i^T\mathbf{s}$, therefore, $d - d' = \mathbf{r}_i^T(\mathbf{e}+\mathbf{f})$. If we set $|(d-d')_q| = |\mathbf{r}_i^T(\mathbf{e}+\mathbf{f})| \le \delta$ for some integer $\delta \le q/32 = o(q)$, by the correctness of $\mathfrak{L}$ (see Lemma 7), two participants can agree on a common secret ξ, i.e., $\xi = \xi_{\text{alice}} = \xi_{\text{bob}}$.

Therefore, only using a Chernoff bound for the approximate correctness of R, we can obtain the correct decryption with overwhelming probability in our extended public-key encryption scheme. □

3.1.3. Messy Public Keys. For constructing a dual-mode encryption cryptosystem from LWE, we have to build upon LWE-based encryption with admitting *messy* (short for *message-lossy*) public keys. We say that a public key pk is *messy*, if a ciphertext output by $\mathsf{LWEEnc}(\mathsf{pk}, \cdot)$ carries no information (statistically) about the encrypted message, i.e., for all μ_0, $\mu_1 \in \{0,1\}^{\ell}$ such that $\mathsf{LWEEnc}(\mathsf{pk}, \mu_0) \approx_s \mathsf{LWEEnc}(\mathsf{pk}, \mu_1)$. Moreover, given some appropriate lattice trapdoor in the aforementioned dual-mode cryptosystem, such messy keys can be efficiently identified. More precisely, the ciphertext produced by LWEEnc is $\mathsf{ct} = (\{\mathbf{p}_i, \beta_i\}_{i\le N}, \{k, \sigma_k, \beta_{\mathfrak{L}}\})$. Therefore, for any fixed public key $\mathsf{pk} = (\mathbf{A}, \mathbf{c})$, we have to consider the statistical distance $\delta(\mathsf{pk})$ between $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_2 \times \mathbb{Z}_q)$ and the distribution of $(\mathbf{r}_i^T\mathbf{A}, R(\mathbf{r}_i^T\mathbf{c}), \mathbf{r}_i^T\mathbf{c})$, where $\mathbf{r}_i \longleftarrow D^m_{\mathbb{Z},\tau}$. For any $\mu_0, \mu_1 \in \mathbb{Z}_2^{\ell}$, both $\mathsf{LWEEnc}(\mathsf{pk}, \mu_0)$ and $\mathsf{LWEEnc}(\mathsf{pk}, \mu_1)$ are close to uniform within $\delta(\mathsf{pk})$, then we have the following:

$$\Delta(\mathsf{LWEEnc}(\mathsf{pk}, \mu_0), \mathsf{LWEEnc}(\mathsf{pk}, \mu_1)) \le 2\delta(pk). \tag{22}$$

If $\delta(\mathsf{pk})$ is negligibly small, then pk is a messy public key. The correctness of LWEDec implies that if pk is generated by $\mathsf{LWEKeyGen}$, it has a large $\delta(\mathsf{pk})$.

As shown in prior lattice-based cryptosystems [5, 16, 18], messy public keys have occupied an important position in security proofs. In particular, it requires [5] that the simulator in the UC model can efficiently identify messy keys with trapdoor information, which demands an explicit condition to identify those keys. Since our dual-mode encryption cryptosystem follows the framework of [5], we also present a sufficient condition for messy public keys as follows:

Lemma 9 (sufficient condition for messy public key). *Let* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, *and* $\mathbf{c} \in \mathbb{Z}_q^m$. *Suppose that the rows of* $\mathsf{pk} = (\mathbf{A}, \mathbf{c})$ *generate* $\mathbb{Z}_q^{n+1}$. *Then for any* $\epsilon = (0, \frac{1}{2})$ *and any Gaussian parameter* $\tau \ge \eta_\epsilon(\Lambda^{\perp}(\mathsf{pk}))$ *used by* LWEEnc, *we have* $\delta(\mathsf{pk}) \le 2\epsilon$.

In particular, if $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge q\sqrt{m}/\tau$ *and* $\tau \ge q \cdot \omega(\sqrt{\log m})/\lambda_1^{\infty}(\Lambda(\mathsf{pk}))$, pk *is messy under* LWEEnc. *That is, for all* $\mu_0, \mu_1 \in \{0,1\}^{\ell}$ *such that*

$$\mathsf{LWEEnc}(\mathsf{pk}, \mu_0) \approx_s \mathsf{LWEEnc}(\mathsf{pk}, \mu_1). \tag{23}$$

Proof. First, we can write $\delta(\mathsf{pk})$ as follows:

$$\begin{aligned} \delta(\mathsf{pk}) =& \Delta\left(\mathcal{U}\left(\mathbb{Z}_q^n, \mathbb{Z}_2, \mathbb{Z}_q\right), D(\mathbf{r}_i^T\mathbf{A}, R(\mathbf{r}_i^T\mathbf{c}), \mathbf{r}_i^T\mathbf{c})\right) \\ \le& \Delta\left(\mathcal{U}\left(\mathbb{Z}_q^n, \mathbb{Z}_2, \mathbb{Z}_q\right), D(\mathbf{r}_i^T\mathbf{A}, \mathbb{Z}_2, \mathbf{r}_i^T\mathbf{c})\right) \\ &+ \Delta(D(\mathbf{r}_i^T\mathbf{A}, \mathbb{Z}_2, \mathbf{r}_i^T\mathbf{c}), D(\mathbf{r}_i^T\mathbf{A}, R(\mathbf{r}_i^T\mathbf{c}), \mathbf{r}_i^T\mathbf{c})) \\ =& \Delta\left(\mathcal{U}\left(\mathbb{Z}_q^n, \mathbb{Z}_q\right), D(\mathbf{r}_i^T\mathbf{A}, \mathbf{r}_i^T\mathbf{c})\right) + \Delta(\mathcal{U}(\mathbb{Z}_2), D(R(\mathbf{r}_i^T\mathbf{c}))) \\ =& \delta_{\mathfrak{L}} + \delta_R, \end{aligned} \tag{24}$$

where δ_R denotes the statistical distance between the distribution of $R(\mathbf{r}_i^T\mathbf{c})$ and $\mathcal{U}(\mathbb{Z}_2)$, and $\delta_{\pounds}$ denotes the statistical distance between the distribution of $(\mathbf{r}_i^T\mathbf{A}, \mathbf{r}_i^T\mathbf{c})$ and $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. Note that in the second part of $\mathsf{ct} = (\{\mathbf{p}_i, \beta_i\}_{i\le N}, \{k, \sigma_k, \beta_{\pounds}\})$ (encrypted by £), we only consider whether the distribution of $(\mathbf{r}_i^T\mathbf{A}, \mathbf{r}_i^T\mathbf{c})$ is nearly-uniform. This is due to the fact that the security of $\beta_{\pounds}$ comes down to whether the distribution of $\sigma_k = f + (\mathbf{r}_k^T \cdot \mathbf{c}) \bmod q$ is close to uniform.

Given $\mathbf{A}$, $\mathbf{c}$, $\mathbf{p}_i^T = \mathbf{r}_i^T\mathbf{A}$ such that $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge q\sqrt{m}/\tau$, by the statistical smoothness of R (see Lemma 6), the distribution of $R(\mathbf{r}_i^T\mathbf{c})$ is *statistically close to uniform* over the randomness of R and $\mathbf{r}_i \leftarrow D_{\mathbb{Z},\tau}^m$, i.e., $\delta_R = \mathsf{negl}(n)$. That is, $\{\beta_i\}_{i\le N}$ are statistically close to uniform bits. Therefore, we only consider whether $\delta_{\pounds}$ is negligibly small.

We can claim $\delta_{\pounds} \le 2\epsilon$ by Lemma 1 (for dimension $n+1$ instead of n). It directly implies that $(\mathbf{r}_i^T\mathbf{A}, \mathbf{r}_i^T\mathbf{c})$ is close to the uniform distribution over $\mathbb{Z}_q^{n+1}$ for $\mathbf{r}_i \leftarrow D_{\mathbb{Z},\tau}^m$ within statistical distance 2ϵ. Then we can claim that $\delta_{\pounds} = \mathsf{negl}(n)$ for the nearly uniform distribution of $(\mathbf{r}_i^T\mathbf{A}, \mathbf{r}_i^T\mathbf{c})$, which directly follows from Lemma 3 (i.e., a consequence of Lemma 2.6 in [16] and the duality between $\Lambda^{\perp}(\mathsf{pk})$ and $\Lambda(\mathsf{pk})$) with the statistically hiding property of £ (see Lemma 7).

More precisely, the first bit of the message (i.e., μ_R) is information-theoretically hidden by $R(\mathbf{r}_i^T\mathbf{c})$, then we must show that the second part of the message (i.e., $\mu_{\pounds}$) is statistically hidden by ξ_{alice} (output by $\pounds_{\text{alice}}$). Here, ξ_{alice} is the first $t - g - 2$ bits of f (randomly chosen from $\{0,1\}^{t-g-2}$), thus f works as a one-time pad for hiding $\mu_{\pounds}$. As a part of ciphertext ct, $\sigma_k = f + (\mathbf{r}_k^T\mathbf{c}) \bmod q$ can be regarded as a one-time pad encryption for hiding f by $\mathbf{r}_k^T\mathbf{c}$. By Lemma 7, f is independent of σ_k, then ξ_{alice} is independent of σ_k. The claim that $\mu_{\pounds}$ is statistically hidden by ξ_{alice} follows the messiness of pk, i.e., f is statistically hidden by $\mathbf{r}_k^T\mathbf{c}$. Therefore, the claim follows.□

Now, we state two following lemmas, one of which claims that most public keys are messy for appropriate parameters, and the other one argues that our extended messy public-key encryption scheme is secure under the LWE assumption.

Lemma 10 (most public keys are messy). *Let* $m \ge 2(n+1)\log q$, $\tau \ge 4\sqrt{m}$, *and* $\mathsf{pk} = (\mathbf{A}, \mathbf{c}) \xleftarrow{\$} \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$. *Then we have* $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge q/4$ *with overwhelming probability, in particular,* $(\mathbf{A}, \mathbf{c})$ *is messy.*

Proof. Let $\mathsf{pk} \in \mathbb{Z}_q^{m\times(n+1)}$ be comprised of $\mathbf{A}$ and $\mathbf{c}$ as above. By Lemma 2 (a consequence of Lemmas 5.1 and 5.3 in [16]), the rows of pk generate $\mathbb{Z}_q^{n+1}$ for all but an at most $q^{-(n+1)} < q^{-n}$ fraction of all pk (by Lemma 5.1 of [16]), and we have $\lambda_1^{\infty}(\Lambda(\mathsf{pk})) \ge q/4$ for all but an at most q^{-n} fraction of all pk (by Lemma 5.3 of [16]). Furthermore, since the set of points that close to $\Lambda(\mathbf{A})$ within distance $q/4$ (in ℓ_∞ norm) has size at most $q^n(q/2)^m$, we have $d_\infty(\mathbf{c}, \Lambda(\mathbf{A})) \ge q/4$ with overwhelming probability over the choice of $\mathbf{c}$ for any fixed $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$. As $m \ge 2n\log q$, the probability that $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$ belongs to those points is at most $q^{-n} = \mathsf{negl}(n)$. Therefore, for any fixed $\mathbf{A}$, with overwhelming probability over the randomness of $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$, we have $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge d_\infty(\mathbf{c}, \Lambda(\mathbf{A})) \ge q\sqrt{m}/\tau$. By Lemma 9, it implies that such $\mathsf{pk} = (\mathbf{A}, \mathbf{c})$ is a messy public key. □

Lemma 11 (security). *Suppose* $m \ge 2(n+1)\log q$, $\tau \ge 4\sqrt{m}$. *Then the above extended messy public-key encryption scheme is secure under the* $\mathsf{LWE}_{q,\chi,n}$ *assumption.*

Proof. With the $\mathsf{LWE}_{q,\chi,n}$ assumption, the public key $(\mathbf{A}, \mathbf{c})$ generated from $\mathsf{LWEKeyGen}$ is computationally indistinguishable from $\mathcal{U}(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m)$. If $(\mathbf{A}, \mathbf{c}) \xleftarrow{\$} \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$, then by Lemma 10, $(\mathbf{A}, \mathbf{c})$ is messy with overwhelming probability and security follows. □

Next, we show that given an appropriate trapdoor, messy public keys can be efficiently identified in the following lemma, which is further used for arguing the sender's statistical security in the messy mode execution of our dual-mode encryption.

Lemma 12 (see [6] Lemma 3.5). *Suppose* $\mathbf{A}$ *is full-rank and* $\tau \ge 6m$. *Let* $\mathsf{TrapGen}(1^n, 1^m, q) \longrightarrow (\mathbf{A}, \mathbf{T})$. *Then there exists an efficient algorithm* $\mathsf{IsMessy}$ *which given a vector* $\mathbf{c}$ *as input, decides whether* $d(\mathbf{c}, \Lambda(\mathbf{A})) \ge q\sqrt{m}/\tau$ *(i.e., the public key* $(\mathbf{A}, \mathbf{c})$ *is identified as messy). The algorithm* $\mathsf{IsMessy}(\mathbf{T}, \mathbf{A}, \mathbf{c})$ *works as follows:*

(1) *Run* $\mathsf{Invert}(\mathbf{T}, \mathbf{A}, \mathbf{c})$ *in Lemma 4.*
(2) *Output not sure, if the output is* $(\mathbf{s}, \mathbf{e})$ *with* $\|\mathbf{e}\| < q/6\sqrt{m}$. *Otherwise, the output is messy.*

That is, if $d(\mathbf{c}, \Lambda(\mathbf{A}) \ge q/6\sqrt{m} \ge q\sqrt{m}/\tau$, *then* $\mathsf{IsMessy}$ *outputs messy by Lemma 4.*

3.2. Dual-Mode Encryption over Multibit Messages. For achieving a UC-secure string OT (as shown in Figure 1), we take the above extended LWE-based messy public-key encryption (see Section 3.1) as the underlying encryption to build a dual-mode encryption over lattices. Here, we slightly change the Gaussian parameter τ to $\tau \ge 6m$ since test messy keys is required (see Lemma 12).

3.2.1. Construction. Now we follow the framework of [6] to show our LWE-based dual-mode cryptosystem for obliviously transferring multibit strings, where the prior encryption scheme $(\mathsf{LWEKeyGen}, \mathsf{LWEEnc}, \mathsf{LWEDec})$ is served as its underlying encryption.

(1) $\mathsf{SetupMessy}(1^n) \longrightarrow (\mathsf{crs}, \mathsf{td}_M)$: Sample $(\mathbf{A}, \mathbf{T}) \longleftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. Pick $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. Output:

$$\mathsf{crs} = (\mathbf{A}, \mathbf{v}),\ \mathsf{td}_M = \mathbf{T}. \tag{25}$$

(2) $\mathsf{SetupDec}(1^n) \longrightarrow (\mathsf{crs}, \mathsf{td}_D)$: Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$. Pick $\mathbf{s}^* \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}^* \longleftarrow \chi^m$. Set $\mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*$ and output:

$$\mathsf{crs} = (\mathbf{A}, \mathbf{v}),\ \mathsf{td}_D = \mathbf{s}^*. \tag{26}$$

(3) $\mathsf{KeyGen}(\mathsf{crs}, b) \longrightarrow (\mathsf{pk}_0, \mathsf{sk}_b)$: Pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \longleftarrow \chi^m$, $\mathbf{f} \xleftarrow{\$} [-B', B']^m$. Output:

$$\mathsf{pk}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f} - b \cdot \mathbf{v},\ \mathsf{sk}_b = \mathbf{s}. \tag{27}$$

It always satisfies that $\mathsf{pk}_b = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f}$ and $\mathsf{pk}_1 - \mathsf{pk}_0 = \mathbf{v}$.

(4) $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}_0, b', \mu) \longrightarrow \mathsf{ct}$: Compute $\mathsf{pk}_{b'} = \mathbf{c} := \mathsf{pk}_0 + b'\mathbf{v}$. Output $\mathsf{ct} \longleftarrow \mathsf{LWEEnc}((\mathbf{A}, \mathbf{c}), \mu)$.

(5) $\mathsf{Dec}(\mathsf{sk}_b, \mathsf{ct}) \longrightarrow \mu$: Parse the ciphertext as $\mathsf{ct} = (\{\mathbf{p}_i, \beta_i\}_{i \le N}, \{k, \sigma_k, \beta_{\pounds}\})$. Output $\mu \longleftarrow \mathsf{LWEDec}(\mathsf{sk}_b, \mathsf{ct})$.

(6) $\mathsf{FindMessy}(\mathsf{td}_M, \mathsf{pk}_0) \longrightarrow \bar{b}$: Run $\mathsf{IsMessy}(\mathsf{pk}_0)$ (defined in Lemma 12). If it outputs messy, output 0. Otherwise, output 1.

(7) $\mathsf{TrapKeyGen}(\mathsf{td}_D) \longrightarrow (\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$: Pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \longleftarrow \chi^m$, $\mathbf{f} \xleftarrow{\$} [-B', B']^m$. Output:

$$\mathsf{pk}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f},\ \mathsf{sk}_0 = \mathbf{s},\ \mathsf{sk}_1 = \mathbf{s} + \mathbf{s}^*. \tag{28}$$

3.3. Dual-Mode Properties. According to Definition 1, we show the above-proposed cryptosystem satisfies the required dual-mode properties.

Lemma 13 (completeness on decryptable branch). *Suppose* $(B + B') \cdot \tau \cdot \sqrt{m} = o(q)$, $\tau \ge \omega(\sqrt{\log m})$, *and* $\delta \le q/32$. *Then, the above scheme is correct.*

Proof. Since the scheme ($\mathsf{LWEKeyGen}, \mathsf{LWEEnc}, \mathsf{LWEDec}$) is taken as the underlying encryption in the above cryptosystem, therefore, the correctness (i.e., on decryptable branch $b \in \{0, 1\}$) of our dual-mode encryption directly follows by Lemma 8. □

Lemma 14 (indistinguishability of modes). *By the hardness of* $\mathsf{LWE}_{q,\chi,n}$, *the above dual-mode encryption satisfies indistinguishability of modes.*

Proof. The difference between two modes is due to the distribution of crs produced by two different setup algorithms (i.e., $\mathsf{SetupMessy}(1^n)$ or $\mathsf{SetupDec}(1^n)$). By Lemma 4, $\mathsf{crs}_M = (\mathbf{A}, \mathbf{v}) \longleftarrow \mathsf{SetupMessy}(1^n)$ is statistically close to $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$. By the $\mathsf{LWE}_{q,\chi,n}$ assumption, $\mathsf{crs}_D = (\mathbf{A}, \mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*) \longleftarrow \mathsf{SetupDec}(1^n)$ is computationally indistinguishable from $(\mathbf{A}, \mathbf{v}) \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. Therefore, computational indistinguishability between two modes follows. □

We hope that Alice (the sender) can achieve statistical security in the messy mode execution of derived OT, which is followed by the security in messy mode, as shown in Definition 1. The security in messy mode (see the undermentioned Lemma 16) can be obtained directly by a consequence of Lemmas 9 and 10 regarding messy public keys to guarantee that at least one of two branches on $(\mathsf{ct}_0, \mathsf{ct}_1)$ is message-lossy under the (possibly malformed) public key $(\mathbf{A}, \mathbf{c})$ given by Bob (the receiver).

As another flavor for clarity, we show in Lemma 15 that the ciphertext $\mathsf{ct}_{\bar{b}}$ of message $\mu_R^{\bar{b}}$ on branch $\bar{b} = 1 - b$ is message-lossy.

Since the encryption of $\mu_R^{\bar{b}}$ (i.e., the former part of $\mu_{\bar{b}}$) encrypted under the messy public key $(\mathbf{A}, \mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f} + \mathbf{v})$ (where $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$) is message-lossy by the statistical smoothness of R, we only need to prove that $\mu_{\pounds}^{\bar{b}}$ (i.e., the latter part of message $\mu_{\bar{b}}$) can be statistically hidden by $\xi_{\text{alice}}^{\bar{b}}$. Moreover, by the correctness of $\pounds$, Bob (the receiver) can decrypt $\beta_{\pounds}^b$ (i.e., the latter part of the ciphertext ct_b on decryptable branch b). That is, on branch b, the key ξ_{bob}^b computed by Bob with given σ_k^b (a.k.a. $\pounds_{\text{bob}}(\sigma_k^b, d_b') \longrightarrow \xi_{\text{bob}}^b$) is equal to the key ξ_{alice}^b computed by Alice (a.k.a. $\pounds_{\text{alice}}(d_b) \longrightarrow (\sigma_k^b, \xi_{\text{alice}}^b)$). Moreover, it requires that Bob cannot recover Alice's encryption key $\xi_{alice}^{\bar{b}}$ on the (messy) branch $\bar{b}$. In particular, the advantage of Bob can correctly recover $\xi_{\text{alice}}^{\bar{b}}$ on branch $\bar{b}$ is $\mathsf{negl}(n)$, i.e., $\Pr[\xi_{\text{bob}}^{\bar{b}} = \xi_{\text{alice}}^{\bar{b}}] = \frac{1}{2^{t-g-2}} + \mathsf{negl}(n)$. This guarantees the encryption of $\mu_{\pounds}^{\bar{b}}$ can be statistically hidden by $\xi_{\text{alice}}^{\bar{b}}$. The following claim follows.

Lemma 15. *For any* $b \in \{0, 1\}$, *the encryption of message* $\mu_{\bar{b}}$ *on branch* $\bar{b} = 1 - b$ *is message-lossy.*

Proof. In messy mode on branch $\bar{b}$, we have $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f} + \mathbf{v}$, where $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. Once $(\mathbf{p}_k^{\bar{b}}, \sigma_k^{\bar{b}})$ is obtained from Alice, Bob can compute $\xi_{\text{bob}}^{\bar{b}}$ as follows:

$$\begin{aligned}\xi^{\bar{b}}_{\text{bob}} &= \left\lfloor \frac{\left(\sigma^{\bar{b}}_k - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s}\right) \bmod q}{2^{g+2}} \right\rfloor \\ &= \left\lfloor \frac{\left(f_{\bar{b}} + \mathbf{r}^{\bar{b}}_k T \cdot \mathbf{c} - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s}\right) \bmod q}{2^{g+2}} \right\rfloor \\ &= \left\lfloor \frac{\left(f_{\bar{b}} + \mathbf{r}^{\bar{b}}_k T \cdot (\mathbf{As} + \mathbf{e} + \mathbf{f} + \mathbf{v}) - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s}\right) \bmod q}{2^{g+2}} \right\rfloor \\ &= \left\lfloor \frac{\left(f_{\bar{b}} + \mathbf{r}^{\bar{b}}_k T \cdot (\mathbf{As} + \mathbf{e} + \mathbf{f}) - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s} + \mathbf{r}^{\bar{b}}_k T \cdot \mathbf{v}\right) \bmod q}{2^{g+2}} \right\rfloor \\ &= \left\lfloor \frac{\left(f_{\bar{b}} + \mathbf{r}^{\bar{b}}_k T \cdot (\mathbf{As} + \mathbf{e} + \mathbf{f}) - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s}\right) \bmod q + \left(\mathbf{r}^{\bar{b}}_k T \cdot \mathbf{v}\right) \bmod q}{2^{g+2}} \right\rfloor .\end{aligned} \tag{29}$$

At the side of Alice, $\mu^{\bar{b}}_{\pounds}$ is encrypted by $\xi^{\bar{b}}_{\text{alice}}$ (i.e., the first $t - g - 2$ bits of $f_{\bar{b}}$ used for encryption of message $\mu_{\bar{b}}$). Therefore, by the mechanism of £, $\xi^{\bar{b}}_{\text{alice}}$ can be recovered at the side of Bob by computing

$$\left\lfloor \frac{\left(f_{\bar{b}} + \mathbf{r}^{\bar{b}}_k T \cdot (\mathbf{As} + \mathbf{e} + \mathbf{f}) - \mathbf{p}^{\bar{b}}_k T \cdot \mathbf{s}\right) \bmod q}{2^{g+2}} \right\rfloor . \tag{30}$$

Now, we analyze that how can Bob recover a correct encryption key $\xi^{\bar{b}}_{alice}$. First, Bob could obtain $(\sigma^{\bar{b}}_k, \mathbf{p}^{\bar{b}}_k)$ from the ciphertext $\mathsf{ct}_{\bar{b}}$ on $\mu_{\bar{b}}$. Since $\sigma^{\bar{b}}_k = f_{\bar{b}} + (\mathbf{r}^{\bar{b}}_k T \cdot \mathbf{c})$, $f_{\bar{b}}$ can be viewed as encrypted by $\mathbf{r}^{\bar{b}}_k T\mathbf{c}$, where the messy key $\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{f} + \mathbf{v}$ (referring to Lemma 16), $f_{\bar{b}}$ is message-lossy under the key $\mathbf{r}^{\bar{b}}_k T\mathbf{c}$. Therefore, $\xi^{\bar{b}}_{\text{alice}}$ is statistically hidden.

Second, we can observe from the above computation of $\xi^{\bar{b}}_{\text{bob}}$ that $\xi^{\bar{b}}_{\text{bob}} \neq \xi^{\bar{b}}_{\text{alice}}$ by the syndrome $\mathbf{r}^{\bar{b}}_k T\mathbf{v}$ except for the case that $\mathbf{v} = \mathbf{0}$. Therefore, the proof for $\Pr[\xi^{\bar{b}}_{\text{bob}} = \xi^{\bar{b}}_{\text{alice}}] = \frac{1}{2^{t-g-2}} + \mathsf{negl}(n)$ turns to show the proof for $\Pr[(\mathbf{r}^{\bar{b}}_k T\mathbf{v}) \bmod q = 0] = \frac{1}{q} + \mathsf{negl}(n)$. More precisely, we can show that the syndrome $(\mathbf{r}^{\bar{b}}_k T\mathbf{v}) \bmod q$ corresponds to a nearly-uniform distribution over $\mathbb{Z}_q$ as the following argument.

Let $\mathbf{v} \xleftarrow{\$} \mathbb{Z}^m_q$ and $\mathbf{r}^{\bar{b}}_k \longleftarrow D^m_{\mathbb{Z},\tau}$. Let $(\mathbf{r}^{\bar{b}}_k T\mathbf{v}) \bmod q = \sum^m_{j=1} r^{\bar{b}}_{kj} \cdot v_j \bmod q$. We have that for $\forall v_j \xleftarrow{\$} \mathbb{Z}_q$, $i \in [m]$, as long as $\exists j \in [m]$, $r^{\bar{b}}_{kj} \neq 0$, then $(\mathbf{r}^{\bar{b}}_k T\mathbf{v}) \bmod q$ is uniform distributed over $\mathbb{Z}_q$. We denote the event $\mathbf{r}^{\bar{b}}_k = \mathbf{0}$ as E_0, then $\Pr[E_0] = D_{\mathbb{Z},\tau}(0)^m$. For clarity, we denote $X = (\mathbf{r}^{\bar{b}}_k T\mathbf{v}) \bmod q$ and Y is a random variable uniformly distributed over $\mathbb{Z}_q$.

$$\begin{aligned}&\Delta(D_1(X), D_2(Y)) \\ &= \frac{1}{2} \sum_{i=0}^{q-1} |\Pr[X = i] - \Pr[Y = i]| \\ &= \frac{1}{2} \sum_{i=0}^{q-1} \left| \Pr[X = i] - \frac{1}{q} \right| \\ &= \frac{1}{2} \left(\left| \Pr[X = 0] - \frac{1}{q} \right| + \sum_{i=1}^{q-1} \left| \Pr[X = i] - \frac{1}{q} \right| \right) \\ &= \frac{1}{2} \left(\left| \Pr[E_0] + (1 - \Pr[E_0]) \frac{1}{q} - \frac{1}{q} \right| + \sum_{i=1}^{q-1} \left| (1 - \Pr[E_0]) \frac{1}{q} - \frac{1}{q} \right| \right) \\ &= \frac{1}{2} \left(\left| \Pr[E_0] - \frac{1}{q} \Pr[E_0] \right| + \sum_{i=1}^{q-1} \frac{1}{q} \Pr[E_0] \right) \\ &= \frac{1}{2} \left(\frac{q-1}{q} \Pr[E_0] + \frac{q-1}{q} \Pr[E_0] \right) \\ &= \frac{q-1}{q} \Pr[E_0] < \Pr[E_0] = D_{\mathbb{Z},\tau}(0)^m < \mathsf{negl}(n)\end{aligned} \tag{31}$$

As long as $m \geq O(n)$, we have $0 < D_{\mathbb{Z},\tau}(0) = \frac{1}{\sum_{y\in\mathbb{Z}}\rho_\tau(y)} < 1$. □

Lemma 16 (security in messy mode). *Suppose that $\tau \geq 6m$, $m \geq 2(n+1)\log q$, and $\delta \leq q/32$. Then, the above scheme satisfies security in messy mode.*

Proof. First, for all pk_0, at least one of the public key $\mathsf{pk}_0 = \mathbf{c}_0$ or $\mathsf{pk}_1 = \mathbf{c}_1$ satisfies $d(\mathbf{c}_b, \Lambda(\mathbf{A})) \geq q/6\sqrt{m}$. This is because if $\mathbf{c}_0$ and $\mathbf{c}_1$ are both close to $\Lambda(\mathbf{A})$, by triangular inequality, $\mathbf{v} = \mathsf{pk}_1 - \mathsf{pk}_0 = \mathbf{c}_1 - \mathbf{c}_0$ is close to $\Lambda(\mathbf{A})$ as well. In particular, if $d(\mathbf{c}_b, \Lambda(\mathbf{A})) \leq q/6\sqrt{m}$ for both $b \in \{0,1\}$, then $d(\mathbf{v}, \Lambda(\mathbf{A})) \leq q/3\sqrt{m}$ with negligible probability over the randomness of $\mathsf{SetupMessy}$ by Lemma 10. Therefore, for all pk_0, at least one of the public key $\mathsf{pk}_0 = \mathbf{c}_0$ or $\mathsf{pk}_1 = \mathbf{c}_1$ is messy by Lemma 10 with overwhelming probability over the choice of $\mathbf{A}$ by Lemmas 2 and 3.

In addition, by Lemma 12, we can efficiently identify a messy branch, i.e., for all pk_0, we use $\mathsf{FindMessy}(\mathbf{T}, \mathbf{A}, \mathsf{pk}_0) \longrightarrow \bar{b}$ to identify the messy branch as $\bar{b}$ and it holds:

$$\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, \bar{b}, \mu_0) \approx_s \mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, \bar{b}, \mu_1). \tag{32}$$

□

Lemma 17 (security in decryption mode). *Assuming $B'/B = \mathsf{negl}(n)$, the above scheme satisfies security in decryption mode.*

Proof. Now we prove that for all $(\mathsf{crs}, \mathsf{td}_D) \longleftarrow \mathsf{SetupDec}(1^n)$, the distributions $(\mathsf{pk}_b, \mathsf{sk}_b)$ generated by either $\mathsf{KeyGen}(\mathsf{crs}_D, b)$ or $\mathsf{TrapKeyGen}(\mathsf{td}_D)$ are statistically close to each other for any $b \in \{0,1\}$.

For any $(\mathsf{crs}_D, \mathsf{td}_D) \longleftarrow \mathsf{SetupDec}(1^n)$, where $\mathsf{crs}_D = (\mathbf{A}, \mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*)$ and $\mathsf{td}_D = \mathbf{s}^*$, we let $(\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{sk}_1) \longleftarrow \mathsf{TrapKeyGen}(\mathsf{td}_D)$. We set the following:

$$\mathsf{pk}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f},\ \mathsf{sk}_0 = \mathbf{s}; \tag{33}$$

$$\mathsf{pk}_1 = \mathbf{A}(\mathbf{s} + \mathbf{s}^*) + (\mathbf{e} + \mathbf{e}^*) + \mathbf{f},\ \mathsf{sk}_1 = \mathbf{s} + \mathbf{s}^*. \tag{34}$$

By Lemma 5 (i.e., $\mathbf{e}$ is statistically close to $\mathbf{e} + \mathbf{e}^*$), the above $(\mathsf{pk}_1, \mathsf{sk}_1)$ is statistically close to the following:

$$\mathsf{pk}_1 = \mathbf{A}(\mathbf{s} + \mathbf{s}^*) + \mathbf{e} + \mathbf{f},\ \mathsf{sk}_1 = \mathbf{s} + \mathbf{s}^*. \tag{35}$$

We denote the regular key pair on decryptable branch b generated by $\mathsf{KeyGen}(\mathsf{crs}_D, b)$ as follows:

$$\widehat{\mathsf{pk}}_b = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f},\ \widehat{\mathsf{sk}}_b = \mathbf{s}, \tag{36}$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \longleftarrow \chi^m$, $\mathbf{f} \xleftarrow{\$} [-B', B']^m$, and $\widehat{\mathsf{pk}}_1 - \widehat{\mathsf{pk}}_0 = \mathbf{v}$.

Therefore, for all $b \in \{0,1\}$, the joint distribution of $(\mathsf{crs}_D, \mathsf{pk}_b, \mathsf{sk}_b)$ is statistically close to that of $(\mathsf{crs}_D, \widehat{\mathsf{pk}}_b, \widehat{\mathsf{sk}}_b)$ by using noise flooding technique (see Lemma 5). □

Corollary 1. *Assuming the hardness of $\mathsf{LWE}_{q,\chi,n}$ with the parameters defined in the above dual-mode encryption cryptosystem, therefore, a UC-secure string OT as shown in Figure 1 with the specifications of Theorem 2 can be achieved.*

Proof. Once a full-fledged dual-mode encryption scheme relying on the hardness of $\mathsf{LWE}_{q,\chi,n}$ is achieved, by Theorem 2, we can directly obtain a UC-secure OT for transmitting multibit strings over lattice (as shown in Figure 1). Specifically, Alice acts as the sender and Bob as the receiver. They both execute the setup phase to obtain crs by selecting messy or decryption mode. In OT session, Bob first runs $\mathsf{KeyGen}(\mathsf{crs}, b)$ for sending pk_0, and then Alice uses pk_0 to encrypt each message $\mu_{b'}$ by running $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}_0, b', \mu_{b'})$. After Bob received two encryptions $(\mathsf{ct}_0, \mathsf{ct}_1)$, he can obtain his chosen message μ_b by running $\mathsf{Dec}(\mathsf{sk}_b, \mathsf{ct}_b)$.

The UC security proof of this proposed string OT is highly similar to that of [5]. Please refer to the following remark and capture a proof sketch of our string OT in the UC model. □

Remark 2 (illustration for simulation). Our dual-mode encryption over multibit messages mainly follows the framework of [6], whose simulation-based security proof is similar to the counterpart of [5], except that in the messy mode, the trapdoor inversion algorithm is simply run once by the crucial property of R. Since our scheme retains the advantage by using R in the trapdoor inversion part, our simulation-based proof also follows [5, 6]. For clarity, we make a sketchy simulation-based proof for our string OT protocol as follows:

Simulator for the case when only the receiver $\mathscr{R}$ is corrupted: Regardless of which mode the protocol runs in the real world, the simulator $\mathcal{S}im$ for a corrupted receiver $\mathscr{R}$ in the ideal world works as follows: run the algorithm $\mathsf{SetupMessy}$ to generate $(\mathsf{crs}, \mathsf{td}_M)$ and follow the simulation steps specified in [5]. We only need to run the trapdoor inversion once for identifying a messy key by the crucial property of R. Then, we can build an efficient simulator when only $\mathscr{R}$ is corrupted.

Simulator for the case when only the sender $\mathscr{S}$ is corrupted: Regardless of which mode the protocol runs in the real world, the simulator $\mathcal{S}im$ for a corrupted receiver $\mathscr{S}$ in the ideal world works as follows: run the algorithm $\mathsf{SetupDec}$ to generate $(\mathsf{crs}, \mathsf{td}_D)$ and follow the simulation steps specified in [5]. Note that we simply need one modification in the reply of the adversary. After $\mathcal{S}im$ sends pk_0 to the corrupted $\mathscr{S}$, the external adversary (or the corrupted $\mathscr{S}$) will reply $(\mathsf{ct}_0, \mathsf{ct}_1)$ to $\mathcal{S}im$. Since the simulator $\mathcal{S}im$ has the trapdoors on both branches, then both messages can be recovered correctly by td_D.

Along with all the aforementioned dual-mode properties, therefore, we can obtain a two-round UC-secure string OT from LWE in the CRS model, as shown in Theorem 1.

4. Conclusions

Targeting to design a UC-secure OT for transmitting multibit strings, we follow up the work of [5, 6] and propose an improved LWE-based dual-mode encryption cryptosystem.

Our scheme not only satisfies the well-defined dual-mode encryption notion but also avoids some costly vector sampling in simple repetitions of sing-bit OT execution for string OT applications. By a comprehensive analysis on both security and efficiency, we show that our scheme performs better than the other two most related works (i.e., [5, 6]).

In addition, a natural problem comes to mind is that whether an OT construction along with the properties, as shown in Theorem 1, is compatible with a polynomial LWE modulus. We believe it is nontrivial due to the use of the noise flooding technique. Another interesting question is to extend this work into their ring-setting version (even over module-lattice) for efficiency in practice. It seems easy to extend R with one-bit hash value output in the ring-setting. However, some building blocks (e.g., the key mechanism scheme and lattice trapdoor techniques) should also be adapted into the ring-setting properly.

Data Availability

No underlying data were collected or produced in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (grant nos. 61902303, 61972457, 62172266, 62002288, U19B2021), the Natural Science Basic Research Program of Shaanxi (program nos. 2024JC-YBMS-475, 2021JM-514), the Key Research and Development Program of Shaanxi (grant no. 2020ZDLGY08-04), the Scientific Research Program Funded by Shaanxi Provincial Education Department (program no. 23JP058), the Young Talent fund of University Association for Science and Technology in Shaanxi, China (program no. 20210116), the Shaanxi Key Laboratory of Blockchain and Secure Computing (no. N-KY-XZ-1101-202110-7349), the Fundamental Research Funds for the Central Universities (no. GK202103093), the Henan Key Laboratory of Network Cryptography Technology (no. LNCT2021-A03), the Innovation Scientists and Technicians Troop Construction Projects of Henan Province, the MOE Layout Foundation of Humanities and Social Sciences (no. 19YJA790007).

References

[1] M. O. Rabin, "How to exchange secrets with oblivious transfer," "Technical Report TR-81, Aiken Computation Lab," Harvard University, 1981.

[2] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pp. 162–167, IEEE, Toronto, ON, Canada, October 1986.

[3] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *STOC '87: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pp. 218–229, Association for Computing Machinery, New York, NY, USA, January 1987.

[4] R. Canetti, "Universally composable security: a new paradigm for cryptographic protocols," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 136–145, IEEE, Newport Beach, CA, USA, October 2001.

[5] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Proceedings of the Annual International Cryptology Conference-CRYPTO 2008*, pp. 554–571, Springer Berlin Heidelberg, Santa Barbara, USA, August 2008.

[6] W. Quach, "UC-secure OT from LWE, revisited," in *Proceedings of International Conference on Security and Cryptography for Networks-SCN 2020*, pp. 192–211, Springer, Cham, Amalfi, Italy, September 2020.

[7] F. Benhamouda, O. Blazy, L. Ducas, and W. Quach, "Hash proof systems over lattices revisited," in *Proceedings of the IACR International Conference on Public-Key Cryptography-PKC 2018, Rio de Janeiro*, pp. 644–674, Springer, Cham, Brazil, March 2018.

[8] Y. T. Kalai, "Smooth projective hashing and two-message oblivious transfer," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques-EUROCRYPT 2005*, pp. 78–95, Springer, Berlin, Heidelberg, Aarhus, Denmark, May 2005.

[9] S. Jiang, G. Gong, J. He, K. Nguyen, and H. Wang, "PAKEs: new framework, new techniques and more efficient lattice-based constructions in the standard model," in *Proceedings of the IACR International Conference on Public-Key Cryptography-PKC 2020*, pp. 396–427, Springer, Cham, Edinburgh, UK, May 2020.

[10] J. Katz and V. Vaikuntanathan, "Smooth projective hashing and password-based authenticated key exchange from lattices," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security-ASIACRYPT 2009*, pp. 636–652, Springer, Berlin, Heidelberg, Tokyo, Japan, December 2009.

[11] Z. Brakerski and N. Döttling, "Two-message statistically sender-private OT from LWE," in *Proceedings of the Theory of Cryptography Conference-TCC 2018*, pp. 370–390, Springer, Cham, Panaji, India, November 2018.

[12] N. Döttling, S. Garg, M. Hajiabadi, D. Masny, and D. Wichs, "Two-round oblivious transfer from CDH or LPN," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques-EUROCRYPT 2020*, pp. 768–797, Springer, Cham, Zagreb, Croatia, May 2020.

[13] R. Canetti, Y. Chen, J. Holmgren et al., "Fiat-Shamir: from practice to theory," in *STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1082–1090, Association for Computing Machinery, Phoenix, AZ, USA, June 2019.

[14] C. Peikert and S. Shiehian, "Noninteractive zero knowledge for NP from (plain) learning with errors," in *Proceedings of the Annual International Cryptology Conference-CRYPTO 2019*, pp. 89–114, Springer, Cham, Santa Barbara, USA, August 2019.

[15] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," in *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 372–381, IEEE, Rome, Italy, October 2004.

[16] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *STOC '08: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 197–206, Association for Computing Machinery, Victoria, BC, Canada, May 2008.

[17] C. Peikert, "Limits on the hardness of lattice problems in ℓ_p norms," *Computational Complexity*, vol. 17, pp. 300–351, 2008.

[18] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 84–93, Association for Computing Machinery, Baltimore, MD, USA, May 2005.
[19] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques-EUROCRYPT 2012*, pp. 700–718, Springer, Berlin, Heidelberg, UK, April 2012.
[20] G. Asharov, A. Jain, A. Lóopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty computation with low communication, computation and interaction via threshold FHE," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques-EUROCRYPT 2012*, pp. 83–501, Springer, Berlin, Heidelberg, Cambridge, UK, April 2012.