UC-Secure OT from LWE, Revisited

Willy Quach
Northeastern University*

Abstract

We build a two-round, UC-secure oblivious transfer protocol (OT) in the common reference string (CRS) model under the Learning with Errors assumption (LWE) with sub-exponential modulus-to-noise ratio. We do so by instantiating the dual-mode encryption framework of Peikert, Vaikuntanathan and Waters (CRYPTO'08). The resulting OT can be instantiated in either one of two modes: one providing statistical sender security, and the other statistical receiver security. Furthermore, our scheme allows the sender and the receiver to reuse the CRS across arbitrarily many executions of the protocol. To the best of our knowledge, this gives the first construction of a UC-secure OT from LWE that achieves both statistical receiver security and unbounded reusability of the CRS. For comparison, there was, until recently, no such construction from LWE satisfying either one of these two properties. In particular, the construction of UC-secure OT from LWE of Peikert, Vaikuntanathan and Waters only provides computational receiver security and bounded reusability of the CRS.

Our main technical contribution is a public-key encryption scheme from LWE where messy public keys (under which encryptions hide the underlying message statistically) can be recognized in time essentially independent of the LWE modulus q.

1 Introduction

Oblivious Transfer (OT), introduced by Rabin [Rab81], is now one of the most fundamental cryptographic primitives, especially in the context of secure multi-party computation [Yao86, GMW87]. Using OT, a sender with two messages m_0, m_1 can send, to a receiver with choice bit b, the message m_b. Intuitively, security ensures that the sender does not learn anything about the receiver's choice bit, and that the receiver does not learn anything about the other message m_{1-b}.

We would like OTs to provide security against *malicious* adversaries, who might deviate arbitrarily from the specifications of the protocol, and ideally achieve the strong guarantees of simulation-based security, where any malicious adversary induces an ideal adversary against an ideal OT functionality. Among the different flavors of simulation-based security

*Email: quach.w@husky.neu.edu.

is the powerful notion of Universal Composability (UC) [Can01], which additionally ensures that security is preserved whenever the OT is executed within larger protocols.

Independently, one would ideally guarantee security against *computationally unbounded* adversaries. While OTs cannot simultaneously ensure statistical security for both receivers and senders, we can hope to provide statistical security for one specific party at a time.

Another desirable property is to require minimal interaction between the sender and the receiver. Two-round OT, which consists of a message from the receiver to the sender and a response from the sender to the receiver, is the best we can hope for. Unfortunately two-round, simulation-secure OT is impossible to achieve in the plain model. We therefore need to rely on some trusted setup assumption, the most standard one (at least in theory) being the availability of a *common reference string* (CRS) to both parties. In this context, we would like to generate a CRS once for all and be able to *reuse* it across many executions of the OT, as opposed to using a fresh CRS every time.

Two-round UC-secure OTs in the CRS model exist under several widely believed assumptions such as the Decisional Diffie-Hellman assumption (DDH), the Quadratic Residuosity assumption (QR), or the Learning with Errors assumption (LWE) [PVW08]. However, to the best of our knowledge, all known techniques to construct UC-secure OT from LWE only achieve weaker security guarantees compared to their group-based or number-theoretic counterparts. In particular, the UC-secure OTs from LWE of [PVW08] only achieves computational receiver security and each CRS can only be securely used a bounded number of times. This is all the more surprising as LWE seems in general much stronger at enabling powerful cryptographic primitives than DDH or QR.

Our Results. Our main result is the construction of a UC-secure OT scheme from the LWE assumption, with *statistical receiver security* and where the CRS can be reused an *unbounded* number of times (between a fixed sender and a fixed receiver). We more precisely obtain the following "dual-mode" OT:

Theorem 1.1 (informal)**.** *Assuming LWE with sub-exponential modulus-to-noise ratio, there exists a two-round UC-secure OT in the common reference string (CRS) model, where the common reference string can be instantiated in two modes:*

- *One provides statistical receiver security and computational sender security;*
- *The other provides statistical sender security and computational receiver security. Furthermore, the CRS in this mode is a common random string.*

In either case, one single CRS can be reused for arbitrarily many executions between the sender and the receiver. Moreover, the two modes for the CRS are computationally indistinguishable.

To the best of our knowledge, there was, until the recent techniques of [CCH$^+$19, PS19] or [DGH$^+$20], no known techniques to achieve *either* statistical receiver security

or unbounded reusability of the CRS from LWE. We develop more on these (orthogonal) approaches later.

For comparison, [PVW08] constructs OTs with the same properties as Theorem 1.1 from either DDH or QR. However, their construction from LWE (with polynomial modulus-to-noise ratio) only achieves weaker security guarantees. Namely, the CRS of the resulting OT can only be reused a bounded number of times, and receiver security is always computational (regardless of the mode). We stress that, in order to reuse the construction of [PVW08] from LWE arbitrarily many times, one would need the trusted setup to generate as many CRSs.

We point out that the original OT construction of [PVW08] from LWE (with weaker security guarantees) uses a *polynomial* LWE modulus, whereas our construction requires a super-polynomial one. As such, our construction requires a comparatively stronger assumption (but still widely believed), and is therefore technically incomparable to the one of [PVW08]. We leave the construction of an OT with security properties similar to Theorem 1.1 from LWE with polynomial modulus as a natural open question.

We also note that the reusability property of the CRS only holds between a *fixed* (ordered) pair of sender and receiver (which is the same reusability property achieved by the constructions from DDH or QR of [PVW08]). While it is possible to generate a fresh CRS for every pair of parties executing the OT, one would ideally have one single "short" CRS of length independent of the number of parties. We leave such a possibility as another interesting open problem.

Related Work. The work of [PVW08] provides a construction of a UC-secure OT from LWE. Even though the latter construction only requires a *polynomial* LWE modulus (while ours requires a super-polynomial one), it only achieves weaker security guarantees, namely a non-reusable CRS and receiver security against computationally bounded senders. We provide a more detailed overview of their construction in Section 1.1. In terms of efficiency, our scheme essentially computes λ instances of [PVW08] (where λ is the security parameter), while using a larger modulus q.

There has been recent works building maliciously-secure OT from LWE. The work of [BD18] builds a *statistically sender-private* OT from LWE in the *plain model* (from LWE with polynomial modulus-to-noise ratio). For comparison, our construction achieves the stronger simulation-based security of UC (as opposed to indistinguishability-based security), at the cost of relying on a trusted common reference string,[1] and can further be instantiated to provide statistical receiver security. However, our construction is less practically efficient: on top of using a larger modulus q, our sender messages are essentially λ times larger, where λ is the security parameter.

The recent work of [DGH+20] gives a generic construction of UC-secure OT starting with any OT with relatively weak security properties, which can be instantiated from CDH

[1] One such setup is necessary to achieve simulation-based security.

or LPN. As far as we understand, their construction provides unbounded reusability of the CRS, and is instantiable from LWE with *polynomial* modulus by using the constructions of [PVW08] or [BD18] as the base OT. For comparison, our construction can provide statistical security for either one of the parties (depending on the mode of Theorem 1.1), while [DGH+20] only provides computational security for both sides. Furthermore, our construction is significantly simpler and arguably more efficient.[2] Notably, we do not require the use of any non black-box techniques.

Non-interactive Zero-Knowledge Proofs. Recently, [CCH+19, PS19] obtained the first construction of non-interactive zero-knowledge proofs (NIZK) (for all NP) from LWE. This NIZK is *dual-mode*, meaning that according to the distribution of the CRS, the resulting NIZK is either statistically sound or statistical zero-knowledge; and those distributions are computationally indistinguishable. Therefore, starting with any *semi-maliciously* secure *dual-mode* OT (where the mode of the CRS gives either statistical receiver privacy or statistical sender privacy), one could potentially obtain a maliciously-secure dual-mode OT using the NIZK of [CCH+19, PS19].

There are, however, several caveats to this approach. First, in order to build dual-mode OT, we would have to start with a dual-mode (semi-malicious) OT. As is, [PVW08], even seen as a semi-malicious protocol, only achieves computational receiver security from LWE. Similarly to our approach, this can be fixed using noise flooding, and would therefore result on also relying on LWE with sub-exponential modulus-to-noise ratio. Second, the NIZKs of [CCH+19, PS19] are *not* adaptively sound when instantiated in statistical zero-knowledge mode. This seems inherent as the reductions for the soundness of [CCH+19, PS19] are black-box [Pas13]. This can be generically fixed using complexity leveraging, but would result in further relying on the *sub-exponential hardness* of LWE. Third, because [CCH+19, PS19] are generic NIZKs for all NP, compiling the OT of [PVW08] would most likely result in practially quite inefficient proofs. As a result, our approach results in an arguably simpler and more efficient protocol, and is provably secure under a weaker assumption (namely, the *polynomial* hardness of LWE with sub-exponential modulus-to-noise ratio). Even though the need for sub-exponential hardness seems hard to avoid in the approach above, building a semi-malicious dual-mode OT from LWE with polynomial modulus-to-noise ratio does not seem out of reach, and we leave it as a natural open question.

Hash proof systems [CS98, CS02] are well-known to enable constructions of OTs. Notably, [Kal05] builds maliciously secure OTs starting from hash proof systems over languages with special properties. However, the resulting constructions only achieve the weaker guarantees of game-based security. Interestingly, one can interpret our construction as following a blueprint similar to [Kal05], using the (weak) hash proof system of Benhamouda et al. [BBDQ18]. However, our strong simulation-security guarantees seem

[2] Our construction essentially computes λ Regev ciphertexts, where λ is the security parameter. For comparison, [DGH+20] uses (among others) a generic zero-knowledge proof (for all NP) to ensure honest evaluation of a garbled circuit encoding an encryption procedure.

to mainly stem from algebraic properties of LWE, as opposed to the hash proof system blueprint itself. In some sense, we use the hash proof system of [BBDQ18] to relax the task of our simulator (used to argue sender security) to a regime where lattice trapdoor techniques directly apply.

Curiously, while hash proof systems are usually defined over languages of *ciphertexts*, we implicitly consider in this work the language of valid *public keys* (and indeed the hash proof system of [BBDQ18] is originally defined over a *dual* Regev scheme, in which ciphertexts correspond to public keys in our construction).

1.1 Technical Overview

Our construction instantiates the dual-mode encryption framework introduced in [PVW08], which results in an OT with the properties of Theorem 1.1. In the same paper, [PVW08] only builds a weaker variant of dual-mode encryption from LWE, which results in a weaker form of OT, namely with neither a reusable CRS nor statistical receiver security. In this work, we build on this original construction of [PVW08] to obtain the original (stronger) version of dual-mode encryption from LWE.

We first upgrade reusability and receiver security by using a standard noise flooding technique, which requires the LWE modulus q to be super-polynomial. Unfortunately, the proof of *sender security* breaks down if we do so. This is because the simulator of [PVW08, GPV08] used to argue sender security runs in time linear in q, and therefore does not run in polynomial time if combined with noise flooding.

We therefore modify the scheme further by incorporating an appropriate *randomized rounding function* to the encryption scheme, which enables an alternative, polynomial time simulator for sender security. Such a rounding function was introduced by Benhamouda et al. [BBDQ18] in the seemingly unrelated context of hash proof systems over languages related to lattices. In a nutshell, while the simulator of [PVW08, GPV08] needs to test that q different points are far from a certain lattice, ours only tests a single point. More details follow.

Dual-Mode Encryption. The *dual-mode encryption* framework, introduced in [PVW08], serves as a modular way to build UC-secure OTs. A dual-mode encryption scheme uses a common reference string (CRS). Given this CRS, a receiver can, given some *branch* $b \in \{0, 1\}$, create a pair of public/secret keys. Using the receiver's public key and the CRS, a sender can encrypt messages with respect to a branch $b' \in \{0, 1\}$. The receiver can then use his secret key to decrypt the message corresponding to the branch $b' = b$ he initially used to create his pair of keys. Looking ahead, in an OT, the branch b corresponds to the receiver's choice bit, and the sender encrypts each his messages $m_{b'}$ to branch b'.

A dual-mode encryption scheme is set up in either one of two modes - messy or decryption - which determines the distribution of the CRS.

In *messy* mode, for all potentially maliciously generated public keys, (at least) one of the encryption branches hides its underlying message *statistically*. Combined with an efficient procedure to identify such so-called messy branches (given an appropriate trapdoor to the CRS), this ensures simulation-based statistical sender security.

In *decryption* mode, one can sample (given an appropriate trapdoor to the CRS) one public key along with two secret keys, one for each of the two branches, such that each of the two potential public/secret key pairs are individually *statistically indistinguishable* from honestly generated keys. This in particular implies that public keys statistically hide their branch b, and more generally, enables a simulator to extract messages encrypted to both branches, thus ensuring simulation-based receiver security.

Finally, a dual-mode encryption requires the two setup modes to be *computationally indistinguishable*. This allows us to argue both computational receiver security in messy mode, and computational sender security in decryption mode, by first switching to the other mode and then relying on the security of the latter.

Overall, [PVW08] showed that dual-mode encryption directly implies a UC-secure OT, where the mode used to pick the CRS - messy or decryption - induces which side is provided statistical security - sender or receiver, respectively. They furthermore show that the CRS can be reused between a fixed pair of sender and receiver, using the Joint-state UC framework of [CR03].

Weak Dual-Mode Encryption from LWE ([PVW08]). Our starting point is the construction of a weak form of dual-mode encryption from LWE of [PVW08]. The construction is a tweak on the (primal) Regev encryption scheme [Reg05], and works as follows. The CRS is set to be a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, along with an offset vector $\mathbf{v} \in \mathbb{Z}_q^m$. Key generation for a branch $b \in \{0,1\}$ works as the Regev scheme, that is, by picking a uniform secret vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, a "short" error term $\mathbf{e}$, and setting $\mathsf{pk}_b = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$, and $\mathsf{sk}_b = \mathbf{s}$. In particular $(\mathbf{A}, \mathsf{pk}_b)$ is a properly generated Regev public key with secret key sk_b. A crucial feature of the construction is that the two public keys (one for each branch) differ by the public offset $\mathbf{v} \in \mathbb{Z}_q^m$, that is: $\mathsf{pk}_1 - \mathsf{pk}_0 = \mathbf{v}$. This in particular defines pk_{1-b}. The public key of the dual-mode encryption scheme is then set to be, say, pk_0 (which given $\mathbf{v}$ determines pk_1), and the secret key sk_b.

To encrypt a message μ with respect to a branch b', one computes a Regev encryption using $(\mathbf{A}, \mathsf{pk}_{b'})$ as the Regev public key. That is, if $\mathsf{pk}_{b'} = \mathbf{A}\mathbf{s} + \mathbf{e}$, one samples a "short" vector $\mathbf{r} \in \mathbb{Z}_q^m$, and outputs $\mathbf{r}^t\mathbf{A}, \mathbf{r}^t(\mathbf{A}\mathbf{s} + \mathbf{e}) + \mathsf{Encode}(\mu)$, where Encode is a fixed encoding procedure. In particular, using sk_b, one can decrypt ciphertexts for branch $b' = b$.

In messy mode, the offset term $\mathbf{v}$ is chosen uniformly at random in $\mathbb{Z}_q^m$. To argue security, the works of [GPV08, PVW08] introduce the notion of *messy* public keys, under which (Regev) encryptions statistically hide their message (and in our context the index of a messy public key corresponds to a messy branch). A core observation, made in [GPV08], is that for $\mathsf{pk} = \mathbf{c}$, if a certain quantity called the *smoothing parameter* of a certain lattice

$\Lambda^{\perp}(\mathbf{A}\|\mathbf{c})$ is sufficiently small, then the public key pk is messy. Using a counting argument, [PVW08] shows that with overwhelming probability over the choice of $\mathbf{A}$ and $\mathbf{v}$, we have that *for all* public key $\mathsf{pk}_0 \in \mathbb{Z}_q^m$, (at least) one of pk_0 or $\mathsf{pk}_1 = \mathsf{pk}_0 + \mathbf{v}$ is messy. Finally, one builds an *extractor* that efficiently identifies one such messy public key. This is done, given an appropriate trapdoor for $\mathbf{A}$, by testing whether all (non-zero) multiples of $\mathsf{pk} = \mathbf{c}$ modulo q are sufficiently far from the lattice $\Lambda(\mathbf{A})$. If so, $\Lambda(\mathbf{A}\|\mathbf{c})$ essentially has a large minimum distance, which in turn implies that $\Lambda^{\perp}(\mathbf{A}\|\mathbf{c})$ indeed has a small smoothing parameter [MR04, Pei08, GPV08].[3]

In decryption mode, the offset term $\mathbf{v}$ is set to be the difference of two LWE samples: $\mathbf{v} = (\mathbf{A}\mathbf{s}_1 + \mathbf{e}_1) - (\mathbf{A}\mathbf{s}_0 + \mathbf{e}_0)$, which is pseudorandom by the LWE assumption; and therefore the two modes are computationally indistinguishable. In particular, one can now set $\widetilde{\mathsf{pk}}_0 = \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$ (and implicitly $\widetilde{\mathsf{pk}}_1 = \mathbf{A}\mathbf{s}_1 + \mathbf{e}_1$), so that the secret keys of *both* branches are known (namely $\mathbf{s}_0$ and $\mathbf{s}_1$, respectively), while all the keys follow the proper distribution. Doing so, however, presents several drawbacks. First, the "trapdoored" public key pk_0 is *fixed* by the CRS. This is ultimately why the CRS can only be reused a bounded number of times fixed in advance. Second, an unbounded adversary can potentially learn non-trivial information about $\mathbf{e}_0$ (and $\mathbf{e}_1$) from $\mathbf{v}$, in which case the trapdoored key $\widetilde{\mathsf{pk}}_0$ does *not* look like a freshly sampled public key. While one can actually argue security against a computationally bounded sender (by relying on LWE), this prevents the scheme from achieving statistically receiver security.

Upgrading security in decryption mode via noise flooding. Our first observation is that all the issues in decryption mode pointed above can be swiftly solved using *noise flooding*. Namely, we define the new public key as $\mathsf{pk}_b = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f}$, where we flood the LWE error $\mathbf{e}$ using a much larger error term $\mathbf{f} \in \mathbb{Z}_q^m$ (which should still not be too large so as to allow decryption). Doing so hides the initial error term $\mathbf{e}$ *statistically*.

We now set the offset term to be a regular LWE sample $\mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*$, and sample our trapdoored public key as $\widetilde{\mathsf{pk}}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f}$ using a fresh secret $\mathbf{s}$, error $\mathbf{e}$, and flooding term $\mathbf{f}$, along with $\mathsf{sk}_0 = \mathbf{s}$ and $\mathsf{sk}_1 = \mathbf{s} + \mathbf{s}^*$. Now the flooding term $\mathbf{f}$ statistically hides the error $\mathbf{e}^*$ in $\widetilde{\mathsf{pk}}_1 = \mathbf{A}(\mathbf{s} + \mathbf{s}^*) + (\mathbf{e} + \mathbf{e}^*) + \mathbf{f}$, and we therefore obtain both statistical receiver security and reusability of the CRS.

Fixing the extractor in messy mode. One drawback of noise flooding is that it requires a *super-polynomial* modulus q. This is because the flooding term $\mathbf{f}$ should be super-polynomially larger than the LWE error $\mathbf{e}$. Therefore, we now have to rely on the hardness of LWE with sub-exponential modulus-to-noise ratio, as opposed to polynomial.

But the most dire issue is that the extractor used to argue security in messy mode is now inefficient. Recall that the extractor of [PVW08] tests that all of the $q - 1$ multiples

[3] More precisely, the counting argument of [PVW08] actually shows that for all pair of public keys, such a test exhibits (at least) one messy public key.

of $\mathsf{pk} = \mathbf{c} \in \mathbb{Z}_q^m$ are far from the lattice $\Lambda(\mathbf{A})$: its runtime is inherently (at least) *linear in* q, and in particular now runs in super-polynomial time if combined with noise flooding. This, in turn, makes the simulator used to argue sender security of the OT run in super-polynomial time. Fixing this issue is the main technical insight of this work.

Instead, we focus on designing an encryption scheme such that messy public keys can be recognized more efficiently. In other words, we would like an efficiently checkable condition on $\mathbf{c} \in \mathbb{Z}_q^m$ under which $\mathbf{r}^t \cdot \mathbf{c}$ (which is the masking term computed during an encryption) is uniform given $\mathbf{r}^t\mathbf{A}$, where $\mathbf{r}$ is drawn from a "small" distribution. To do so, we use the techniques developed in [BBDQ18], which introduces an explicit *randomized rounding function* R (with output $\{0,1\}$) with the following (informal) properties:

1. If $\mathbf{c} \in \mathbb{Z}_q^m$ is "*sufficiently far*" from the lattice $\Lambda(\mathbf{A})$, then $R(\mathbf{r}^t \cdot \mathbf{c})$ is *statistically close* to uniform, even given $\mathbf{r}^t \cdot \mathbf{A}$;

2. If $\mathbf{c} = \mathbf{A}\mathbf{s}+\mathbf{e} \in \mathbb{Z}_q^m$ is "*sufficiently close*" to the lattice $\Lambda(\mathbf{A})$, then $R(\mathbf{r}^t \cdot \mathbf{A}\mathbf{s}) = R(\mathbf{r}^t \cdot \mathbf{c})$ with good probability (say $\geq 2/3$).

In a nutshell, the rounding function R is defined in such a way so that the other multiples $k \cdot \mathbf{c}$, $k \neq 1$ are in some sense filtered out by (the absence of) corresponding harmonics of its density function. We refer the reader to Lemma 2.7 or [BBDQ18] for more details on the construction of this rounding function.

This induces a variant of the Regev encryption scheme, where the Regev public key is $(\mathbf{A}, \mathbf{c})$, but the message is now masked using $R(\mathbf{r}^t \cdot \mathbf{c})$, where approximate correctness is ensured by providing $\mathbf{r}^t \cdot \mathbf{A}$ in the ciphertext, and relying on Property 2. Correctness can then be amplified by giving many independent such ciphertexts.

By Property 1, public keys $(\mathbf{A}, \mathbf{c})$ are messy as soon as $\mathbf{c}$ is "sufficiently far" from $\Lambda(\mathbf{A})$ - and crucially, independently of the other multiples of $\mathbf{c}$ - which can be tested efficiently using an appropriate trapdoor for $\mathbf{A}$ [AR03, Pei08, GPV08, MP12]. In our construction, we use the LWE decoder of [MP12], which (arguably) results in a substantially simpler extractor than the original versions [GPV08, PVW08].

To finish the proof, it suffices to note that the random offset $\mathbf{v}$ is, with high probability, "sufficiently far" from the lattice $\Lambda(\mathbf{A})$, in which case *for all* public key $\mathsf{pk}_0 = \mathbf{c}$, either $\mathbf{c}$ or $\mathbf{c}+\mathbf{v}$ is "sufficiently far" from $\Lambda(\mathbf{A})$. Otherwise their difference $\mathbf{v}$ would not be "sufficiently far" from the lattice. Therefore at least one of them is messy and recognized as such by the extractor.

2 Preliminaries

Notations Throughout the paper, λ will denote a security parameter, and $n = n(\lambda)$ the dimension of the LWE problem. We will often abuse notation and use n as the security parameter.

We denote by $\mathsf{poly}(n)$ any function f such that $f(n) = O(n^c)$ for some constant c; and $\mathsf{negl}(n)$ denotes any function such that $f(n) = n^{-\omega(1)}$. We will denote (column) vectors by bold lower cases (e.g., $\mathbf{c}$) and matrices by bold upper cases (e.g, $\mathbf{A}$). We denote the transposition operation by $\cdot^t$ (e.g., $\mathbf{c}^t$). For $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{B} \in \mathbb{Z}_q^{m\times k}$, we denote by $(\mathbf{A}\|\mathbf{B}) \in \mathbb{Z}_q^{m\times(n+k)}$ their horizontal concatenation. Unless specifically stated otherwise, all the distances $d(\cdot,\cdot)$ and norms $\|\cdot$ we use are in the ℓ_2 norm. $\|\cdot\|_\infty$ denotes the infinity norm. We use the notation $[k]$ for the set of integers $[1,\ldots,k]$. For a set E, we will sometimes denote by $\mathcal{U}(E)$ the uniform distribution over E, and we will use $x \xleftarrow{\$} E$ to denote the uniform sampling $x \leftarrow \mathcal{U}(E)$.

We define the statistical distance between two random variables X and Y over some domain Ω as $\mathsf{SD}(X,Y) = \frac{1}{2}\sum_{w\in\Omega} |X(w) - Y(w)|$. We say that two ensembles of random variables $X = \{X_\lambda\}_\lambda, Y = \{Y_\lambda\}_\lambda$ are *statistically indistinguishable* if $\mathsf{SD}(X_\lambda, Y_\lambda) \leq \mathsf{negl}(\lambda)$; and we denote it with $X \approx_s Y$.

We say that two ensembles of random variables $X = \{X_\lambda\}_\lambda, Y = \{Y_\lambda\}_\lambda$ are *computationally indistinguishable* if for all probabilistic, polynomial time (PPT) distinguisher $\mathcal{A} \to \{0,1\}$, we have: $|\Pr[\mathcal{A}(X_\lambda) = 1] - \Pr[\mathcal{A}(Y_\lambda) = 1]| \leq \mathsf{negl}(\lambda)$; and we denote it with $X \approx_c Y$.

For $B \in \mathbb{R}$, we say that a distribution ψ is *B-bounded* if $\Pr_{x\leftarrow\psi}[|x| \geq B] \leq \mathsf{negl}(\lambda)$.

2.1 Dual Mode Encryption

We recall the definition of dual-mode encryption ([PVW08]).

Definition 2.1 (Dual Mode Encryption)**.** A *Dual-Mode Encryption scheme* with message space $\{0,1\}^k$ is a tuple of PPT algorithms ($\mathsf{SetupMessy}$, $\mathsf{SetupDec}$, KeyGen, Enc, Dec, $\mathsf{FindMessy}$, $\mathsf{TrapKeyGen}$) with the following syntax:

- $\mathsf{SetupMessy}(1^\lambda) \to (\mathsf{crs}, \mathsf{td}_M)$: Given the security parameter λ, the setup algorithm outputs a common reference string crs along with a trapdoor td_M.
- $\mathsf{SetupDec}(1^\lambda) \to (\mathsf{crs}, \mathsf{td}_D)$: Given the security parameter λ, the setup algorithm outputs a common reference string crs along with a trapdoor td_D.
- $\mathsf{KeyGen}(\mathsf{crs}, b) \to (\mathsf{pk}, \mathsf{sk}_b)$: Given a reference string crs and a branch $b \in \{0,1\}$, the key-generation algorithm outputs a public key pk and a secret key sk_b for branch b.
- $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, b', \mu) \to \mathsf{ct}$: Given a reference string crs, a public key pk, a branch $b' \in \{0,1\}$ and a message $\mu \in \{0,1\}^k$, the encryption algorithm outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{crs}, \mathsf{sk}, \mathsf{ct}) \to \mu$: Given a reference string crs, a secret key sk and a ciphertext ct, the decryption algorithm outputs a message μ.

- $\mathsf{FindMessy}(\mathsf{crs}, \mathsf{td}_M, \mathsf{pk}) \to \bar{b}$: Given a reference string crs, a trapdoor in messy mode td_M and a (possibly malformed) public key pk, the algorithm outputs a branch $\bar{b} \in \{0,1\}$.
- $\mathsf{TrapKeyGen}(\mathsf{crs}, \mathsf{td}_D)$: Given a reference string crs, a trapdoor in decryption mode td_D, the algorithm outputs keys $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$ where pk is a public-key, and sk_0 and sk_1 are secret keys for branches 0 and 1, respectively.

We require the following properties to hold:

- **Completeness on decryptable branch:** For all $\mu \in \{0,1\}^k$ and $b \in \{0,1\}$:

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

  whether $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupMessy}(1^\lambda)$ or $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupDec}(1^\lambda)$, and where $(\mathsf{pk}, \mathsf{sk}_b) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, b)$.
- **Indistinguishability of modes:** We have:

$$\mathsf{crs}_M \approx_c \mathsf{crs}_D,$$

  where $(\mathsf{crs}_M, \mathsf{td}_M) \leftarrow \mathsf{SetupMessy}(1^\lambda)$ and $(\mathsf{crs}_D, \mathsf{td}_D) \leftarrow \mathsf{SetupDec}(1^\lambda)$.
- **Security in messy mode** (a.k.a. trapdoor identification of a messy branch): With overwhelming probability over $(\mathsf{crs}, \mathsf{td}_M) \leftarrow \mathsf{SetupMessy}(1^\lambda)$, it holds that for all (possibly malformed) pk and all messages $\mu_0, \mu_1 \in \{0,1\}^k$:

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

  where $\bar{b} \leftarrow \mathsf{FindMessy}(\mathsf{crs}, \mathsf{td}_M, \mathsf{pk})$.
- **Security in decryption mode** (a.k.a. trapdoor generation of keys decryptable on both branches): With overwhelming probability over $(\mathsf{crs}, \mathsf{td}_D) \leftarrow \mathsf{SetupDec}(1^\lambda)$, we have that for every $b \in \{0,1\}$:

$$(\mathsf{crs}, \mathsf{pk}, \mathsf{sk}_b) \approx_s (\mathsf{crs}, \mathsf{KeyGen}(1^\lambda)),$$

  where $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{TrapKeyGen}(\mathsf{td}_D)$.

[PVW08] showed that any dual-mode encryption scheme implies a "dual-mode" UC-secure OT. We refer to [PVW08] for more precise definitions of UC security.

Theorem 2.1 (Dual-Mode Encryption implies UC-Secure OT [PVW08])**.** *Assume* $(\mathsf{SetupMessy}, \mathsf{SetupDec}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FindMessy}, \mathsf{TrapKeyGen})$ *is a dual-mode encryption scheme. Then, there exists a protocol realizing the multi-session functionality* $\widehat{\mathcal{F}}_{OT}$ *in the* $\mathcal{F}_{CRS}$*-hybrid model, under static corruptions.*

Furthermore the protocol can be instantiated in two modes (each over a distinct functionality $\mathcal{F}_{CRS}$*): one providing statistical sender security and computational receiver security; and the other statistical receiver security and computational sender security.*

2.2 Lattices and Learning with Errors

We will use the following lemma:

Lemma 2.2 (Noise flooding (e.g [AJL+12]))**.** *Let $B = B(\lambda)$, $B' = B'(\lambda) \in \mathbb{Z}$ be two integers, and let $e_1 \in [-B, B]$. Suppose that $B/B' = \mathsf{negl}(\lambda)$. Then:*

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

The following lemma states that for appropriate parameters, random q-ary lattices have a large minimum distance and are full-rank:

Lemma 2.3. *[GPV08, Lemmas 5.1 and 5.3] Suppose $m \geq 2n \log q$. Then:*

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

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

$$\rho_\tau : x \mapsto \exp(-\pi \|x\|^2 / \tau^2).$$

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

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

The following lemma states that random q-ary lattices have a small smoothing parameter:

Lemma 2.4 ([MR04, Pei08, GPV08])**.** *For any m-dimensional lattice Λ and real $\epsilon > 0$, we have:*

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

In particular, with overwhelming probability over the choice of $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, we have that for any function $\omega(\sqrt{\log m})$, there exists a negligible function $\epsilon(m)$ such that

$$\eta_\epsilon(\Lambda^\perp(\mathbf{A})) \leq \omega(\sqrt{\log m}).$$

Lattices and gaussians. We recall basic definitions related to lattices.

For an integer m, an m-dimensional lattice Λ is a discrete subgroup of $\mathbb{R}^m$. For a lattice Λ, its dual Λ^* is defined as $\Lambda^* = \{\mathbf{r} \in \mathsf{Span}_{\mathbb{R}}(\Lambda) \,|\, \forall \mathbf{x} \in \Lambda, \langle \mathbf{x}, \mathbf{r} \rangle \in \mathbb{Z}\}$.

The *minimum distance* (in infinity norm) of a lattice is defined as $\lambda_1^\infty(\Lambda) = \min_{x \in \Lambda \setminus \{0\}} \|x\|_\infty$.

For $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, we will use the following q-ary lattices defined by $\mathbf{A}$:

$$\Lambda(\mathbf{A}) = \{\mathbf{A}\mathbf{s} \,|\, \mathbf{s} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m, \quad \Lambda^\perp(\mathbf{A}) = \{\mathbf{r} \in \mathbb{Z}^m \,|\, \mathbf{r}^t \mathbf{A} = \mathbf{0}^t \bmod q\}.$$

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

We say, for $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, that $\mathbf{A}$ is *full-rank* the columns of $\mathbf{A}$ are linearly independent.

For $\epsilon > 0$, the *smoothing parameter* of a lattice Λ, introduced in [MR04] and denoted $\eta_\epsilon(\Lambda)$, is the smallest $\tau > 0$ such that $\rho_{1/\tau}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$. Intuitively, for $\tau \geq \eta_\epsilon(\Lambda)$ for some small ϵ, we have that for $\mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}$, $\mathbf{r} \bmod \Lambda$ is roughly uniform. In particular, if $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is full-rank and $\tau \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{A}))$, then for $\mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}$, $\mathbf{r}^t\mathbf{A} \bmod q$ is roughly uniform in $\mathbb{Z}_q^{1 \times n}$.

Learning with Errors. We recall the definition of the Learning with Errors assumption.

Definition 2.5 (Decisional Learning with Errors assumption [Reg05])**.** Let n and $q = q(n) \geq 2$ be integers, and χ a distribution over $\mathbb{Z}$. The *Learning with Errors* assumption $\mathsf{LWE}_{q,\chi,n}$ states that for all $m = \mathsf{poly}(n)$ the following distributions are computationally indistinguishable:

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \approx_c (\mathbf{A}, \mathbf{b}),$$

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

[Reg05] showed that for all $B \geq \widetilde{\Omega}(\sqrt{n})$, there exists a B-bounded distribution $\chi = \chi(n)$ such for all $q = q(n) \geq 2$, breaking $\mathsf{LWE}_{q,\chi,n}$ is as hard as (quantumly) solving GapSVP_γ and SIVP_γ within approximation factor $\gamma = \widetilde{O}(\sqrt{n}q/B)$. For comparison, the best known (provable) algorithm for GapSVP_γ runs in time $2^{\widetilde{\Omega}(n/\log\gamma)}$ [Sch87].

Lattice trapdoors.

Lemma 2.6 (Lattice trapdoors [MP12])**.** *There exists a PPT algorithm* $\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T})$*, which on input some integers* n*,* $q \geq 2$ *and* $m \geq \Omega(n \log q)$*, satisfies the following properties:*

- *The distribution of* $\mathbf{A}$ *is within negligible statistical distance from* $\mathcal{U}(\mathbb{Z}_q^{m \times n})$*;*
- *There exists a polynomial-time, deterministic algorithm* $\mathsf{Invert}(\mathbf{T}, \mathbf{A}, \mathbf{c})$*, which on input* $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$ *where* $\mathbf{s} \in \mathbb{Z}_q^m$ *and* $\mathbf{e} \in \mathbb{Z}_q^m$ *such that* $\|\mathbf{e}\| < q/6\sqrt{m}$*, outputs* $(\mathbf{s}, \mathbf{e})$*.*

Without loss of generality, the algorithm Invert only outputs some $(\mathbf{s}, \mathbf{e})$ whenever $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$ and $\|\mathbf{e}\| < q/6(\sqrt{m})$, as these conditions can be checked efficiently.

2.3 Smooth rounding over Lattices

We recall the properties of the rounding function defined in [BBDQ18].

Lemma 2.7 (Statistically smooth rounding [BBDQ18])**.** *Suppose $m = \Theta(n \log q)$. Let $R : \mathbb{Z}_q \mapsto \{0,1\}$ be a randomized rounding function defined as:*

$$R(x) = \begin{cases} 1 \text{ with probability } & \frac{1}{2} + \frac{\cos(2\pi x/q)}{2} \\ 0 \text{ with probability } & \frac{1}{2} - \frac{\cos(2\pi x/q)}{2} \end{cases}.$$

Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{p} \in \mathbb{Z}_q^n$, and $\tau \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ for some $\epsilon = \mathsf{negl}(n)$. Then the following properties hold:

- ***Statistical Smoothness:*** *Suppose $\mathbf{A}$ is full rank. Then, for all $\mathbf{c} \in \mathbb{Z}_q^m$ such that $d(\mathbf{c}, \Lambda(\mathbf{A})) \geq q\sqrt{m}/\tau$, we have:*

$$\left| \Pr_{R, \mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}} \left[R(\langle \mathbf{r}, \mathbf{c} \rangle) = 1 \,|\, \mathbf{r}^t \mathbf{A} = \mathbf{p}^t \right] - 1/2 \right| \leq \mathsf{negl}(n),$$

  where the probability is taken over $\mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}$ and the internal randomness of R.

- ***Approximate Correctness:*** *For all $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}_q^m$ satisfies $\|\mathbf{e}\| \leq B$ (i.e., $d(\mathbf{c}, \Lambda(\mathbf{A})) \leq B$) where $B \cdot \tau \cdot \sqrt{m} = o(q)$, then for all large enough n:*

$$\Pr_{R, \mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}} [R(\mathbf{r}^t \mathbf{A}\mathbf{s}) = R(\mathbf{r}^t \mathbf{c}\rangle)] \geq 2/3.$$

Remark 2.8 (Statistically correct rounding)**.** In addition to the rounding function presented above, [BBDQ18] also defines a rounding function with *statistically correctness* and *approximate smoothness* (meaning that its bias is bounded). For our ultimate purpose of building a dual-mode encryption scheme from LWE, such a rounding scheme would also suffice (modulo direct modifications in the encryption scheme). However the parameters imposed by such a rounding function are slightly more constraining, and in particular require a super-polynomial modulus q in the first place. This for instance disallows the use of the scheme described in Section 3.1 with a polynomial modulus q (even though our final construction requires a super-polynomial modulus anyway).

Remark 2.9 (Rounding function as an (approximate) hash proof system)**.** [BBDQ18] introduced the rounding function of Lemma 2.7 in the context of (approximate) *hash proof systems*. Throughout the paper, one can alternatively view our use of the rounding function R as using the following approximate hash proof system.

The system is defined over the language of points "close" to $\Lambda(\mathbf{A})$, where a witness for "closeness" is $\mathbf{s} \in \mathbb{Z}_q^n$ so that $\mathbf{A} \cdot \mathbf{s}$ is a close lattice point. The hashing key of the scheme is defined as $\mathbf{r} \leftarrow D^m_{\mathbb{Z},\tau}$, and the projection key as $\mathbf{p}^t = \mathbf{r}^t \cdot \mathbf{A}$. The hash of a point $\mathbf{c} \in \mathbb{Z}_q^m$ is defined as $R(\langle \mathbf{r}, \mathbf{c} \rangle)$ and the projected hash as $R(\mathbf{r}^t \mathbf{A}\mathbf{s})$. (Approximate) *correctness* is ensured for points "very close" to the lattice $\Lambda(\mathbf{A})$ (at distance at most B), while (statistical) *smoothness* is ensured for points "very far" from $\Lambda(\mathbf{A})$ (at distance at least $q\sqrt{m}/\tau$). Note that these two bounds differ by (at least) a factor m, and the hash proof system provides no guarantees for points $\mathbf{c}$ that lie in between.

3 Dual-Mode Encryption from LWE

We now focus on building a dual-mode encryption scheme from LWE. In Section 3.1, we introduce a public-key encryption scheme where most messy public keys can be tested efficiently. This serves as a basis for our actual construction of a dual-mode encryption scheme in Section 3.2.

3.1 A Messy Public-Key Encryption Scheme

We use the rounding function defined in Lemma 2.7 to define a variant of the (primal) Regev encryption scheme, where the message is now masked by a *rounded* bit (instead of a value in $\mathbb{Z}_q$). Looking ahead, this scheme has the crucial property that public key messiness is efficiently testable (given an appropriate trapdoor) in time essentially independent of the LWE modulus q.

Parameters. Let $n = n(\lambda) = \lambda$, $q = q(\lambda) \geq 2$ be integers. Let $m \geq 2(n+1)\log q$.

Let $\tau \geq 4\sqrt{m}$ (and $\tau \geq 6(m)$ if one wants to test messy public keys).

Let $\chi = \chi(n)$ given by Definition 2.5 be a $B = B(n)$ bounded distribution where $B = \widetilde{\Omega}(\sqrt{n})$.

Let $B' \in \mathbb{Z}$ be such that $(B + B') \cdot \tau\sqrt{m} = o(q)$ (which implies $q \geq \omega(B' + \sqrt{n})m)$).[4]

Let R be the rounding function defined in Lemma 2.7.

Construction. We define our public key encryption scheme ($\mathsf{SmoothKeyGen}, \mathsf{SmoothEnc}, \mathsf{SmoothDec}$) over message space $\mathcal{M} = \{0,1\}$ as follows:

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

- $\mathsf{Smooth.Enc}(\mathsf{pk}, \mu \in \{0,1\})$: For $i \in [\lambda]$, sample $\mathbf{r}_i \leftarrow D^m_{\mathbb{Z},\tau}$. Compute $\mathbf{p}_i^t = \mathbf{r}_i^t \cdot \mathbf{A} \in \mathbb{Z}_q^{1 \times n}$, and:
$$\beta_i \leftarrow R(\mathbf{r}_i^t \cdot \mathbf{c}) \oplus \mu,$$
and output:
$$\mathsf{ct} = \left(\{\mathbf{p}_i, \beta_i\}_{i \leq \lambda}\right).$$

- $\mathsf{Smooth.Dec}(\mathsf{sk}, \mathsf{ct})$: Compute, for all $i \in [\lambda]$:
$$b_i \leftarrow R(\mathbf{p}_i^t \cdot \mathbf{s}) \oplus \beta_i,$$
and output the majority bit of the b_i's.

[4] We will only need $B' \neq 0$ when building a dual-mode encryption in Section 3.2.

Looking ahead, the additional term $\mathbf{f}$ added to $\mathbf{c}$ will be used in the dual-mode encryption scheme to help arguing security in decryption mode. We note that removing this term from $\mathbf{c}$ does not affect any of the properties listed below.

Properties. We first argue correctness the scheme.

Lemma 3.1 (Correctness)**.** *Suppose* $(B + B') \cdot \tau \cdot \sqrt{m} = o(q)$ *and* $\tau \geq \omega(\sqrt{\log m})$*. Then the scheme above is correct.*

Proof. By Lemma 2.4, there exists some $\epsilon = \mathsf{negl}(n)$,[5], such that with overwhelming probability over the choice of $\mathbf{A}$, we have $\tau \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$.

By approximate correctness of the rounding function R (Lemma 2.7), for all $i \in [\lambda]$, we have:

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

over the internal randomness of R and $\mathbf{r}_i \leftarrow D^m_{\mathbb{Z},\tau}$ alone. Using a Chernoff bound, we obtain that decryption is correct with overwhelming probability. □

Next, we give a sufficient condition over public-keys so that the associated encryption hides the message information-theoretically. Looking ahead, this will be used to argue both security of the scheme above, and messy mode security of the derived dual-mode encryption scheme.

Following the terminology of [PVW08, GPV08], we say that a public key pk is *messy* (which stands short for *message-lossy*) if $\mathsf{SmoothEnc}(\mathsf{pk}, m)$ *statistically hides* the message m for all m, that is:

$$\mathsf{SmoothEnc}(\mathsf{pk}, 0) \approx_s \mathsf{SmoothEnc}(\mathsf{pk}, 1).$$

Lemma 3.2 (Sufficient condition for public key messiness)**.** *Let* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$*, and* $\mathbf{c} \in \mathbb{Z}_q^m$*. Fix* $\epsilon = \mathsf{negl}(n)$*, and suppose* $\tau \geq \eta_\epsilon(\Lambda(\mathbf{A}))$*.*

If $d(\mathbf{c}, \Lambda(\mathbf{A}) \geq q\sqrt{m}/\tau$ *and* $\mathbf{A}$ *is full rank, then the public key* $(\mathbf{A}, \mathbf{c})$ *is messy, that is:*

$$\mathsf{SmoothEnc}(\mathsf{pk}, 0) \approx_s \mathsf{SmoothEnc}(\mathsf{pk}, 1).$$

Proof. By statistical smoothness of the rounding function (Lemma 2.7), every bit $R(\mathbf{r}_i^t \cdot \mathbf{c})$ is *statistically close to uniform* given $\mathbf{A}, \mathbf{c}, \mathbf{p}_i$ over the internal randomness of R and $\mathbf{r}_i \leftarrow D^m_{\mathbb{Z},\tau}$ alone. In particular, $\{\beta_i\}_{i \in [\lambda]}$ are statistically close to uniform bits. □

Lemma 3.3 (Most public keys are messy)**.** *Suppose* $m \geq 2(n+1)\log q$ *and* $\tau \geq 4\sqrt{m}$*.*

Let $(\mathbf{A}, \mathbf{c}) \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$*. Then with overwhelming probability,* $d(\mathbf{c}, \Lambda(\mathbf{A})) \geq q/4$ *and in particular* $(\mathbf{A}, \mathbf{c})$ *is messy.*

[5] Looking more closely at Lemma 2.4 ϵ can be exponentially small if $\tau \geq m$

Proof. By Lemma 2.3, $(\mathbf{A}\|\mathbf{c})$ is full-rank except with negligible probability $q^{-(n+1)}$.

Furthermore, we have that for any fixed $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $d_\infty(\mathbf{c}, \Lambda(\mathbf{A})) \geq q/4$ with overwhelming probability over the choice of $\mathbf{c}$. This is because the set of points within distance $q/4$ (in ℓ_∞ norm) from $\Lambda(\mathbf{A})$ has size at most $q^n \cdot (q/2)^m$. As $m \geq 2n \log q$, the probability that $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$ belongs to those points is at most q^{-n}, which is negligible.

This implies that for any fixed $\mathbf{A}$, $d(\mathbf{c}, \Lambda(\mathbf{A})) \geq d_\infty(\mathbf{c}, \Lambda(\mathbf{A})) \geq q\sqrt{m}/\tau$ with overwhelming probability over the randomness of $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$ alone. The result then follows by Lemma 3.2. □

The observation above allows us to argue security of the scheme:

Lemma 3.4 (Security)**.** *Suppose $m \geq 2(n+1)\log q$ and $\tau \geq 4\sqrt{m}$. Then the encryption scheme is secure under the $\mathsf{LWE}_{q,\chi,n}$ assumption.*

Proof. By the $\mathsf{LWE}_{q,\chi,n}$ assumption, given $\mathbf{A}$, the vector $\mathbf{c}$ in the public key is computationally indistinguishable from uniform in $\mathbb{Z}_q^m$. Now, if $(\mathbf{A}, \mathbf{c}) \xleftarrow{\$} \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$, we have that $(\mathbf{A}, \mathbf{c})$ is messy with overwhelming probability by Lemma 3.3, and security follows. □

Next, we describe how to identify messy public keys given an appropriate trapdoor.

Lemma 3.5 (Weak identification of messy public keys)**.** *Suppose $\tau \geq 6m$. Let $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. Suppose $\mathbf{A}$ is full-rank and $\tau \geq 6m$. Then there exists a polynomial-time algorithm $\mathsf{IsMessy}$ which on input a vector $\mathbf{c}$ decides whether $d(\mathbf{c}, \Lambda(\mathbf{A}) \geq q\sqrt{m}/\tau)$. In particular, if this is the case, the public key $(\mathbf{A}, \mathbf{c})$ is identified as messy.*

Proof. We define the algorithm $\mathsf{IsMessy}$ as follows:

$\mathsf{IsMessy}(\mathbf{T}, \mathbf{A}, \mathbf{c})$:

1. Run $\mathsf{Invert}(\mathbf{T}, \mathbf{A}, \mathbf{c})$ from Lemma 2.6.
2. If the output is $(\mathbf{s}, \mathbf{e})$ with $\|\mathbf{e}\| \leq q/6\sqrt{m}$, then output $\mathsf{not\ sure}$,

   Otherwise output messy.

By Lemma 2.6, if $d(\mathbf{c}, \Lambda(\mathbf{A})) \geq q/6\sqrt{m} \geq q\sqrt{m}/\tau$, then $\mathsf{IsMessy}$ outputs messy. □

As in [PVW08], $\mathsf{IsMessy}$ might output $\mathsf{not\ sure}$ even though the public key is actually messy. This is because we only test for a sufficient condition for messiness in Lemma 3.3. However if $\mathsf{IsMessy}(\mathbf{T}, \mathbf{A}, \mathbf{c})$ outputs messy, then the public key $(\mathbf{A}, \mathbf{c})$ is indeed messy. Looking ahead, in our construction of a dual-mode encryption scheme, we will ensure that at least one of the two branches is recognized as messy.

3.2 Dual-Mode Encryption

We now describe our dual-mode encryption scheme.

Parameters. The constraints over the parameters mostly inherits from Section 3.1:

Let $n = n(\lambda)$, $q = q(\lambda) \geq 2$ be integers. Let $m \geq 2(n+1)\log q$.

Let $\tau \geq 6m$ (from Lemma 3.5);

Let $\chi = \chi(n)$ given by Definition 2.5 be a $B = B(n)$ bounded distribution where $B = \widetilde{\Omega}(\sqrt{n})$.

Let $B' \in \mathbb{Z}$ be such that $(B + B') \cdot \tau\sqrt{m} = o(q)$ (which implies $q \geq \omega(B' + \sqrt{n})m)$).

Let R be the randomized rounding function defined in Lemma 2.7.

Suppose furthermore that:

- $B/B' = \mathsf{negl}(n)$.

For instance, one can set (without trying to optimize the parameters): $n = \lambda$, $q = n^{\omega(1)}$,[6] $m = 2\log q$, $B = n$, $B' = q/n^3$, $\tau = 6m$.

Construction. In the following, the input and output public keys pk of the dual-mode scheme are implicitly public keys for the branch $b = 0$.

- $\mathsf{SetupMessy}(1^\lambda) \to (\mathsf{crs}, \mathsf{td}_M)$: Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. Pick $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. Output:
$$\mathsf{crs} = (\mathbf{A}, \mathbf{v}), \quad \mathsf{td}_M = \mathbf{T}.$$

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

- $\mathsf{KeyGen}(\mathsf{crs}, b) \to (\mathsf{pk}_0, \mathsf{sk}_b)$: Pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{f} \xleftarrow{\$} [-B', B']$. Output:
$$\mathsf{pk}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f} - b \cdot \mathbf{v}, \quad \mathsf{sk}_b = \mathbf{s}.$$
In particular, we have $\mathsf{pk}_b = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f}$ and $\mathsf{pk}_1 - \mathsf{pk}_0 = \mathbf{v}$.

- $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk}_0, b', \mu) \to \mathsf{ct}$: Compute $\mathsf{pk}_{b'} = \mathbf{c} \coloneqq \mathsf{pk}_0 + b' \cdot \mathbf{v}$.
For $i \in [\lambda]$, sample $\mathbf{r}_i \leftarrow D_{\mathbb{Z},\tau}^m$. Compute $\mathbf{p}_i^t = \mathbf{r}_i^t \cdot \mathbf{A} \in \mathbb{Z}_q^{1 \times n}$, and:
$$\beta_i \leftarrow R(\mathbf{r}_i^t \cdot \mathbf{c}) \oplus \mu,$$
and output:
$$\mathsf{ct} = (\{\mathbf{p}_i, \beta_i\}_{i \leq \lambda}).$$

[6] Looking ahead, setting q to be slightly super-polynomial suffices to ensure simulation security for the receiver (in decryption mode) with negligible statistical distance. For more practical purposes, one would prefer this distance to be (at least) sub-exponentially small, in which case one would rather set $q = 2^{n^\delta}$ for some $0 < \delta < 1/2$.

- $\mathsf{Dec}(\mathsf{crs}, \mathsf{sk}_b, \mathsf{ct}) \to \mu$: Parse the ciphertext as $\mathsf{ct} = (\{\mathbf{p}_i, \beta_i\}_{i \le \lambda})$. Compute, for all $i \in [\lambda]$:

  $$b_i \leftarrow R(\mathbf{p}_i^t \cdot \mathbf{s}) \oplus \beta_i,$$

  and output the majority bit of the b_i's as μ.

- $\mathsf{FindMessy}(\mathsf{td}_M, \mathsf{pk}_0) \to \bar{b}$: Run $\mathsf{IsMessy}(\mathsf{pk}_0)$ (defined in Lemma 3.5). If it outputs messy, output 0.

  Otherwise, output 1.

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

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

Remark 3.6 (Common random string in messy mode)**.** The CRS in messy mode is *statistically* close to uniform. As the trapdoor is only used in the proof of security, we can replace the CRS in messy mode with a common *random* string instead, and adding an appropriate hybrid in the proof of security. The original construction of [PVW08] also satisfies this property.

3.3 Dual-mode properties

Lemma 3.7 (Completeness on decryptable branch)**.** *Suppose* $(B + B') \cdot \tau \cdot \sqrt{m} = o(q)$ *and* $\tau \ge \omega(\sqrt{\log m})$*. Then the scheme above is correct.*

Proof. This follows directly by correctness of $(\mathsf{SmoothKeyGen}, \mathsf{SmoothEnc}, \mathsf{SmoothDec})$, as $(\mathbf{A}, \mathsf{pk}_b)$ (where $\mathsf{pk}_0 \leftarrow \mathsf{KeyGen}(\mathsf{crs}, b)$ and $\mathsf{pk}_b = \mathsf{pk}_0 + b \cdot \mathbf{v}$) is distributed identically as $\mathsf{SmoothKeyGen}(1^\lambda)$, and Enc and Dec procede as $\mathsf{SmoothEnc}$ and $\mathsf{SmoothDec}$, respectively. □

Lemma 3.8 (Indistinguishability of modes)**.** *Assuming* $\mathsf{LWE}_{q,\chi,n}$*, the scheme satisfies indistinguishability of modes.*

Proof. By Lemma 2.6, the CRS in messy mode $\mathsf{crs}_M = (\mathbf{A}, \mathbf{v}) \leftarrow \mathsf{SetupMessy}(1^\lambda)$ is statistically close to uniform over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$.

Now, by the $\mathsf{LWE}_{q,\chi,n}$ assumption, $(\mathbf{A}, \mathbf{v}) \xleftarrow{\$} (\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$ is computationally indistinguishable from $(\mathbf{A}, \mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*)$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{s}^* \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}^* \leftarrow \chi^m$, which is identically distributed as $\mathsf{crs}_D \leftarrow \mathsf{SetupDec}(1^\lambda)$. □

Lemma 3.9 (Security in messy mode)**.** *Suppose that* $\tau \ge 6m$*, and* $m \ge 2(n+1)\log q$*. Then the scheme satisfies security in messy mode.*

Proof. We first argue that with overwhelming probability over the probability of $(\mathbf{A}, \mathbf{v}) \leftarrow \mathsf{SetupMessy}(1^\lambda)$ alone, we have that *for all* public key pk_0, at least one of the public keys $\mathsf{pk}_0 = \mathbf{c}_0$ or $\mathsf{pk}_1 = \mathbf{c}_1$ satisfies $d(\mathbf{c}_b, \Lambda(\mathbf{A})) \geq q/6\sqrt{m}$, and is in particular messy by Lemma 3.3 (conditioned on $\mathbf{A}$ being full-rank and $\tau \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$, which happen with overwhelming probability over the choice of $\mathbf{A}$ by Lemma 2.3 and Lemma 2.4).

This is simply because if both $\mathbf{c}_0$ and $\mathbf{c}_1$ are close to $\Lambda(\mathbf{A})$, then by triangular inequality $\mathbf{v} = \mathbf{c}_1 - \mathbf{c}_0$ must be as well, which only happens with negligible probability over the randomness of $\mathsf{SetupMessy}$. More precisely, if $d(\mathbf{c}_b, \Lambda(\mathbf{A})) \leq q/6\sqrt{m}$ for both $b \in \{0,1\}$, then $d(\mathbf{v}, \Lambda(\mathbf{A})) \leq q/3\sqrt{m}$, which only happens with negligible probability over the randomness of $(\mathbf{A}, \mathbf{v}) \leftarrow \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, by Lemma 3.3.

Now conditioned on the above, by Lemma 3.5, we have that if $\mathsf{FindMessy}(\mathbf{T}, \mathbf{A}, \mathsf{pk}_0)$ does not output 0, then it outputs 1, which is therefore a messy branch. In particular, for all pk_0, the output branch of $\bar{b} = \mathsf{FindMessy}(\mathsf{td}_M, \mathsf{pk}_0)$ is messy and therefore:

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

□

Lemma 3.10 (Security in decryption mode)**.** *Assuming $B'/B = \mathsf{negl}(n)$, the scheme satisfies security in decryption mode.*

Proof. Because $\mathsf{pk}_1 - \mathsf{pk}_0 = \mathbf{v}$ is in the CRS, it suffices to argue that the distributions $(\mathsf{pk}_b, \mathsf{sk}_b)$ generated using either $\mathsf{KeyGen}(\mathsf{crs}_D, b)$ or $\mathsf{TrapKeyGen}(\mathsf{td}_D)$ are statistically close.

Fix $(\mathsf{crs}_D, \mathsf{td}_D) \leftarrow \mathsf{SetupDec}(1^\lambda)$, where $\mathsf{crs}_D = (\mathbf{A}, \mathbf{v} = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*)$ and $\mathsf{td}_D = \mathbf{s}^*$.

Let $(\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{TrapKeyGen}(\mathsf{td}_D)$. We have:

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

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

which is, by Lemma 2.2, distributed statistically close to:

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

Regular keys for branch b (output by $\mathsf{KeyGen}(\mathsf{crs}, b)$) are generated as:

$$\widetilde{\mathsf{pk}}_b = \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{f} \quad , \quad \widetilde{\mathsf{sk}}_b = \mathbf{s},$$

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

Therefore, for all $b \in \{0,1\}$ the joint distributions $(\mathsf{crs}_D, \mathsf{pk}_b, \mathsf{sk}_b)$ and $(\mathsf{crs}_D, \widetilde{\mathsf{pk}}_b, \widetilde{\mathsf{sk}}_b)$ are statistically close to each other. □

Finally, using Theorem 2.1, we obtain the following:

Corollary 3.11. *Assuming $\mathsf{LWE}_{q,\chi,n}$ with the parameters defined in the construction, there exists an UC-secure OT with the specifications of Theorem 2.1.*

Acknowledgements. We thank Vinod Vaikuntanathan and Daniel Wichs for helpful discussions and comments about this work. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing for the Spring 2020 program "Lattices: Algorithms, Complexity, and Cryptography".

References

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In Pointcheval and Johansson [PJ12], pages 483–501.

[AR03] Dorit Aharonov and Oded Regev. A lattice problem in quantum NP. In *44th FOCS*, pages 210–219. IEEE Computer Society Press, October 2003.

[BBDQ18] Fabrice Benhamouda, Olivier Blazy, Léo Ducas, and Willy Quach. Hash proof systems over lattices revisited. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 644–674. Springer, Heidelberg, March 2018.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In *51st ACM STOC*, pages 1082–1090. ACM Press, 2019.

[CR03] Ran Canetti and Tal Rabin. Universal composition with joint state. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 265–281. Springer, Heidelberg, August 2003.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002.

[DGH+19] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from cdh or lpn. Cryptology ePrint Archive, Report 2019/414, 2019. `https://eprint.iacr.org/2019/414`.

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 768–797. Springer, Heidelberg, May 2020.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 78–95. Springer, Heidelberg, May 2005.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [PJ12], pages 700–718.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354. Springer, Heidelberg, March 2013.

[Pei08] Chris Peikert. Limits on the hardness of lattice problems in lp norms. *Comput. Complex.*, 17(2):300–351, May 2008.

[PJ12] David Pointcheval and Thomas Johansson, editors. *EUROCRYPT 2012*, volume 7237 of *LNCS*. Springer, Heidelberg, April 2012.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Hovav Shacham and Alexandra Boldyreva,

editors, *CRYPTO 2019, Part I*, LNCS, pages 89–114. Springer, Heidelberg, August 2019.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Heidelberg, August 2008.

[Rab81] Michael O. Rabin. How to Exchange Secrets with Oblivious Transfer, 1981. Harvard Aiken Computational Laboratory TR-81.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Sch87] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(2):201–224, June 1987.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.