On the Hardness of LWE with Trapdoor Hints and Applications to Fuzzy Cryptography

Divesh Aggarwal*
National University of Singapore

Aayush Jain†
Carnegie Mellon University

Huijia Lin‡
University of Washington

Yuetian Wu§
Peking University

Abstract

Learning with Errors (LWE) and its variants have revolutionized modern cryptography, enabling a wide range of applications that previously seemed out of reach. In recent years, there has been growing interest in *hinted* variants of LWE, in which the adversary receives carefully structured side information, and these assumptions have further expanded the cryptographic toolkit. In this work, we study a natural hinted LWE assumption that we call *LWE with Trapdoor Hints*. Concretely, we consider the hardness of distinguishing an LWE sample ($\mathbf{A}$, $\mathbf{s}^\mathsf{T}\mathbf{A}+\mathbf{e}^\mathsf{T}$ mod q) from uniform even in the presence of *a full-rank homogeneous trapdoor* $\mathbf{T}$ *for* $\mathbf{A}$. We show that when the error $\mathbf{e}$ is drawn from a Gaussian distribution that is only slightly wider than the decoding radius associated with $\mathbf{T}$, this problem remains hard under the standard LWE assumption. We then present two compelling applications of this problem that were previously out of reach:

First, we construct the first Euclidean-ball obfuscator from LWE satisfying distributional virtual black-box security. Our construction achieves the functional preserving property, reusability, and distributional VBB security, assuming the obfuscated inputs are drawn from a suitable LWE-compatible distribution such as a discrete Gaussian. To the best of our knowledge, such a notion was not previously known even assuming indistinguishability obfuscation. As a byproduct, we resolve the open question posed by Fuller, Meng, and Reyzin (Asiacrypt 2013) concerning using lattice trapdoors to construct computational fuzzy extractors supporting the Euclidean metric and with drastically larger error tolerance.

Second, we introduce a new primitive that we call *Threshold Difference Revealing Encryption* (TDRE), which can be viewed as a public-key, two-input analogue of fuzzy extractors. Standard fuzzy extractors are highly effective for fuzzy authentication, but they are fundamentally asymmetric and do not directly support the task of comparing two independently encoded, hidden biometrics. We believe that TDRE can help fill this important conceptual gap in current techniques for fuzzy cryptography.

*dcsdiva@nus.edu.sg
†aayushja@andrew.cmu.edu
‡rachel@cs.washington.edu
§yuetian-wu@outlook.com

Contents

1 Introduction **2**
- 1.1 Applications of LWE with Trapdoor Hints . . . 4
  - 1.1.1 Euclidean Ball Obfuscation with Distributional Virtual Black Box Security . . . 4
  - 1.1.2 Threshold Difference Revealing Encryption . . . 6
- 1.2 Related Works . . . 9
- 1.3 Open Questions . . . 11

2 A Brief Overview of Techniques **12**
- 2.1 From LWE with Trapdoor Hints to Applications . . . 14
- 2.2 Achieving Strong Simulation Security with Leakage . . . 16

3 Preliminaries **17**
- 3.1 Goldreich-Levin for Large Fields . . . 18
- 3.2 Lattices and Gaussians . . . 19
- 3.3 Lattice Smoothing . . . 22
- 3.4 Learning with Errors . . . 24
- 3.5 Trapdoor Sampling . . . 24
- 3.6 Entropic LWE . . . 28

4 Hardness Results of the LWE with Trapdoor Hints Problem **29**
- 4.1 Vanilla LWE with Trapdoor Hints . . . 29
- 4.2 LWE with Error Leakage . . . 35

5 Euclidean Ball Obfuscation with Distributional VBB security **40**
- 5.1 Definitions . . . 40
- 5.2 Our Construction . . . 40

6 Threshold Difference Revealing Encryption **42**
- 6.1 Definitions . . . 42
  - 6.1.1 Correctness . . . 43
  - 6.1.2 Security . . . 43
- 6.2 Our Construction . . . 44

A Proof sketch for Theorem 4.3 **50**

1 Introduction

Since its introduction by Regev [Reg05], the Learning with Errors (LWE) problem has become a central foundation of modern cryptography. Beyond playing a leading role in post-quantum cryptography, LWE has enabled a striking range of advanced primitives, including fully homomorphic encryption and others [Reg05, Gen09, BV11, BGG+14, GVW15, WZ17, GKW17]. Over the past few years, there has been a concerted effort to further enlarge the expressive power of lattice-based cryptography by formulating stronger assumptions on the hardness of the LWE problem.

A powerful family of assumptions that have stood out in this revolution is what we call a "hinted LWE" assumption. These are LWE variants in which the adversary gets some helper auxiliary string. Most of the popular hinted LWE assumptions consider LWE (or related lattice problems such as SIS) in presence of hints that contain short vectors or trapdoors related to the matrices in the assumption. Recall that a trapdoor with respect to the source matrix $\mathbf{A}$ and target matrix $\mathbf{B}$ is a small-normed matrix such that $\mathbf{AT} = \mathbf{B} \bmod q$. Depending on whether $\mathbf{B}$ is the all zero matrix or not, they are called homogeneous or inhomogeneous trapdoors respectively. A full trapdoor, on the other hand, is a homogeneous trapdoor with full-rank over the integers.

Many hinted LWE assumptions with trapdoor or short vector hints have been proposed including evasive LWE [Wee22, Tsa22, VWW22], succinct LWE [Wee24], decomposed LWE [AMR25], and k-LWE/k-SIS [BF11, LPSS14]. These richer assumptions have already led to new constructions of witness encryption and null-iO, optimal or near-optimal broadcast and attribute-based encryption, laconic function evaluation, and fully succinct randomized encodings for RAM programs [Wee22, VWW22, Wee25, CHW25, AMR25].

It is therefore important to understand which forms of trapdoor/short vector hints preserve security, and which new applications can genuinely be based on these hinted variants of LWE. Indeed, the status of these hinted LWE assumptions is far from uniform: some, such as k-LWE and k-SIS, are polynomially related to standard LWE/SIS [BF11, LPSS14], whereas stronger forms such as evasive LWE [Wee22, Tsa22, VWW22] that underlie recent approaches to iO now admit explicit counterexamples and zeroizing attacks [BÜW24, AMYY25, DJM+25, HJL25]. Some others such as succinct LWE, decomposed LWE or evasive LWE that underlie witness encryption or null iO [VWW22] are heuristics without known attacks. Progress on understanding this boundary of the security of hinted LWE assumptions with trapdoor/short vector hints could have direct consequences for longstanding problems at the frontier of lattice-based cryptography.

In this work, we study the security and applications of the following natural hinted LWE problem, where one is given an LWE sample with respect to a random matrix $\mathbf{A}$ *together with a full trapdoor* for $\mathbf{A}$.

Definition 1.1 (LWE with Trapdoor Hints)**.** *The distinguishing LWE with Trapdoor Hints problem* $\mathsf{TLWE}_{n,m,q,\sigma_e,\sigma_t}$ *is parameterized with dimension* n*, modulus* q*, number of samples* $m = \Omega(n \log q)$ *and Gaussian width parameters* σ_e, σ_t *and consists of distinguishing:*

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q)$$

from

$$(\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}),$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ *is a random matrix,* $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ *is a random vector,* $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_e}^m$ *is a discrete Gaussian error*

vector, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ *is a random vector and* $\mathbf{T}$ *is a Gaussian random full trapdoor sampled in the lattice* $\mathcal{L}_q^\perp(\mathbf{A})$ *with width parameter* σ_t [1].

One could observe that if σ_e is sufficiently small in some way related to σ_t, the LWE with Trapdoor Hints problem can be solved in polynomial time. Indeed, one could multiply the sample $\mathbf{b} = \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q$ with $\mathbf{T}$ to get $\mathbf{e}^\mathsf{T}\mathbf{T} \bmod q$. If $\sigma_e \cdot \sigma_t = o(q/\sqrt{m})$, then with high probability $\mathbf{e}^\mathsf{T}\mathbf{T} \bmod q = \mathbf{e}^\mathsf{T} \cdot \mathbf{T}$ over the reals. This lets one compute $\mathbf{e}$ rendering the assumption trivially broken. For this attack to work, one needs $\sigma_e = o\big(q/(\sigma_t \cdot \sqrt{m})\big)$. The main contribution of this work is to show that if σ_e is barely larger than this threshold (a small polynomial in n factor larger[2]), then LWE with Trapdoor Hints can be shown to be at least as hard as LWE itself. In more detail, we show that:

Theorem 1.1 (LWE with Trapdoor Hints is at Least as Hard as LWE, informal)**.** *Let* n *and* q *be LWE dimension and modulus, and* m *be the number of samples such that* $4n \log q \le m \le \mathsf{poly}(n)$.

For any integer base parameter τ *subject to constraint* $\tau^c/2 < q < \tau^c$ *for some integer* c [3]*, the LWE with trapdoor hints problem* $\mathsf{TLWE}_{n,m,q,\sigma_e,\sigma_t}$ *is hard, when the noise and trapdoor width* σ_e, σ_t *are sufficiently large w.r.t.* τ*, i.e.,*

$$\sigma_t \ge m \cdot \tau \cdot \omega(\sqrt{\log n}) \,, \qquad \sigma_e \ge (q/\tau) \cdot \omega(\sqrt{\log n}) \,,$$

assuming the hardness of LWE with dimension $\Omega(n/\log q)$ *and modulus-to-noise ratio* $\rho = \tau m^{1.5}$.

We now highlight what is distinctive about this theorem compared to prior assumptions. Prior LWE assumptions with short-vector or trapdoor hints largely follow one of two templates. The first, exemplified by k-LWE/k-SIS, reveals a small number of short vectors for the homogeneous system $\mathbf{A}\mathbf{v} \bmod q = \mathbf{0}$. The second, which includes more recent trapdoor-hint assumptions such as evasive LWE and succinct LWE (which implies decomposed LWE), reveals inhomogeneous trapdoors associated with matrices appearing in the assumption. Moreover, prior works on these assumptions—whether proving security from standard LWE or giving new attacks—have focused primarily on structural properties of the source and target matrices, rather than on the magnitudes of the LWE error or the trapdoor itself. In particular, none of these assumptions study the hardness of the problem when a full trapdoor for the LWE matrix itself is revealed. By contrast, our theorem considers the setting where a full homogeneous trapdoor for $\mathbf{A}$ is revealed, rather than merely a few short vectors or an inhomogeneous trapdoor. To avoid the standard textbook LWE attack in this regime, it crucially requires both the error and the trapdoor to be sufficiently large. We complement this mere avoidance of attack with a provable guarantee under standard LWE.

In fact, something even stronger is true. As we show in Theorem 4.1, LWE with respect to the matrix $\mathbf{A}$ that is of the form $[\mathbf{B}|\mathbf{B}\mathbf{R} + \mathbf{G}_\tau]$, where $\mathbf{B}$ is a random matrix, $\mathbf{R}$ is a random Boolean matrix, and $\mathbf{G}_\tau$ is a gadget matrix [MP12] with a suitably chosen base τ (instead of 2), remains hard even given the matrix $\mathbf{R}$. $\mathbf{R}$ can be seen as an inhomogeneous trapdoor for $\mathbf{A}$ with target

[1] More specifically, $\mathcal{L}_q^\perp(\mathbf{A})$ refers to the lattice that contains all integer vectors $\mathbf{v}$ satisfying $\mathbf{A} \cdot \mathbf{v} \bmod q = \mathbf{0}$. $\mathbf{T}$ is generated by repeatedly sampling discrete Gaussian vectors with width parameter σ_t on lattice $\mathcal{L}_q^\perp(\mathbf{A})$ until obtaining a full-rank set of independent vectors (see Definition 3.4).

[2] Although, we believe that this attack should be the best one can do and the hardness should hold even with $\sigma_e = \tilde{O}\big(q/(\sigma_t \cdot \sqrt{m})\big)$. Refer to the open questions for more details.

[3] This implicitly requires q to be chosen so that such an integer τ exists. See more discussions about this condition on q in open questions and the technical overview.

matrix $\mathbf{G}_\tau$ in the sense that $\mathbf{A}\left[\begin{smallmatrix}-\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] = \mathbf{G}_\tau$, which enables one to sample a full trapdoor $\mathbf{T}$. This gives an "inhomogeneous-trapdoor" variant of the LWE with Trapdoor Hints problem above. A good feature about this version is that the sampling of $\mathbf{A}$ and $\mathbf{R}$ is "public-coin", which can be desirable in applications that need a transparent setup.

We also show stronger versions of LWE with Trapdoor Hints with twofold strengthening: enabling the LWE secret to be reused, and showing resilience to short error leakage. First, we show that the LWE secret $\mathbf{s}$ in a LWE with Trapdoor Hints instance $(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q)$ can be reused in additional LWE samples $(\mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)$, and these additional samples are pseudorandom. Second, we show that the security of LWE with Trapdoor Hints holds even at the presence of leakage $h(\mathbf{e})$ on the error vector $\mathbf{e}$, where h can be an arbitrary (possibly randomized) efficiently computable function with short output (up to $O(\log \lambda)$ bits). These strengthened results are of independent interest, and moreover enable us to achieve stronger functionality and security in our applications. We refer the readers to Section 4 for details.

1.1 Applications of LWE with Trapdoor Hints

We explore two major applications of LWE with trapdoor hints, which, thanks to Theorem 1.1, can be based under hardness of standard LWE. The formal definitions and full constructions are deferred to Sections 5 and 6.

1.1.1 Euclidean Ball Obfuscation with Distributional Virtual Black Box Security

We study the natural problem of obfuscating fuzzy point functions. Enabling obfuscation for restricted yet useful function classes under standard assumptions is an important research agenda, complementary to research on obfuscation for general circuits. Unlike the case for general circuits, where virtual black-box security is impossible [BGI+01], and constructions of iO based on well-studied assumptions remain notoriously complex [JLS21, JLS22], relying on a strong set of assumptions, obfuscation for useful function classes, such as, constructions of point functions [Can97, Wee05, BS16], conjunctions and wildcards [BVWW16, BKM+18, BLMZ19, BW19], and compute-and-compare obfuscation [WZ17, GKW17], often admit stronger security than iO, simpler constructions, and can be based on more diverse assumptions.

Fuzzy point functions naturally extend point functions by allowing correctness to tolerate noise in the input, and are closely connected to the theory of fuzzy extractors and biometric matching [DRS04, Boy04]. In a fuzzy point obfuscation scheme, one samples the hidden input $\mathbf{x}$ from a high-entropy distribution $\mathcal{D}$, and the goal is to obfuscate a circuit $\mathsf{B}_{t,\mathbf{x}}(\mathbf{x}')$ that, on input $\mathbf{x}'$, checks whether $\mathsf{dis}(\mathbf{x}, \mathbf{x}') \le t$ under an appropriate distance metric, outputting 1 if the check passes and 0 otherwise. One may also consider a payload-carrying variant $\mathsf{B}_{t,\mathbf{x},\mathbf{y}}(\mathbf{x}')$ (also called multi-bit fuzzy point obfuscation), in which the circuit outputs a multi-bit message $\mathbf{y} \in \{0,1\}^*$ when the check passes and $\perp$ otherwise [CD08].

Challenges in Fuzzy Point Obfuscation. For fuzzy point obfuscation, several properties are particularly challenging:

- Since fuzzy point obfuscation has applications for dealing with highly sensitive information source such as biometrics, a strong simulation guarantee is especially appealing. A gold

standard is the *distributional virtual black-box (VBB)* security. Informally, a scheme satisfies VBB security w.r.t. an (unpredictable) input distribution $x \leftarrow \mathcal{D}$ if for every predicate h, the pair $(O_{\mathbf{x}}, h(\mathbf{x}))$ is computationally indistinguishable from $(\widetilde{O}, h(\mathbf{x}))$, where $O_{\mathbf{x}}$ is the obfuscated program and $\widetilde{O}$ is produced by a simulator given only the security parameter. [4] Intuitively, this says that given the fuzzy ball obfuscation of a random point $\mathbf{x}$, the adversary cannot learn any bit of information about $\mathbf{x}$.

- Furthermore, while one-sided correctness—only guarantee detection of matches when the obfuscated program is evaluated on every input that is t-close to the hidden point—is the common correctness requirement in the fuzzy extractor literature and may suffice in some settings, it is desirable to achieve functionality preserving in the context of obfuscation, which requires *two-sided correctness*. For fuzzy point obfuscation, this requires also rejection of mismatches, rejecting every t-far input. It ensures that the obfuscated program perfectly implements the radius-t fuzzy ball centered at a point $\mathbf{x}$ (with overwhelming probability over the randomness of obfuscation).

- Finally, prior cryptographic approaches to fuzzy matching have focused mostly on Hamming-type metrics. For biometrics represented by continuous measurements or learned real-valued embeddings, however, cosine similarity and Euclidean distance (i.e., ℓ_2 distance) are often the more natural measures. This makes Euclidean fuzzy-ball obfuscation especially desirable [BDHV07, FP19, GRD19, ZCY21, PXS25].

Distributional-VBB Secure Euclidean Ball Obfuscator. A central challenge, therefore, is to construct a scheme satisfying most or all of the above desiderata simultaneously. To the best of our knowledge, the only known construction for fuzzy point function obfuscation achieving distributional virtual black-box security is due to Galbraith and Zobernig [GZ19]. Their construction relies on a novel subset-product-style pseudorandomness assumption, works for Hamming distance, and is not known to provide reusable security (discussed shortly below). In fact, it is not known whether even iO implies a fuzzy ball obfuscation simultaneously providing the above guarantees.

Using LWE with Trapdoor Hints, we show the first fuzzy point function obfuscation scheme for Euclidean distance that simultaneously achieves two sided correctness, distributional virtual black-box security, and multi-bit payloads (see Section 5.1 for the formal definition). Our scheme achieves distributional-VBB security when the hidden center is drawn from discrete Gaussian distributions with appropriate parameters.[5]

Theorem 1.2 (Euclidean Ball Obfuscator, informal)**.** *Assume* $\mathsf{TLWE}_{n,m,q,\sigma_e,\sigma_t}$ *is hard. Then there exists an obfuscation scheme for (multi-bit) fuzzy point functions over* ℓ_2 *metric with distance threshold* $t < q/(4m\sigma_t)$*. It satisfies distributional VBB security when the hidden center is sampled from the discrete Gaussian distribution* $\mathcal{D}^m_{\mathbb{Z},\sigma_e}$*, and perfect correctness (with overwhelming probability over the randomness of obfuscation).*

[4] This is in fact the notion of distributional indistinguishability, which implies distributional VBB security [WZ17] and is more common in practice.

[5] We believe, however, that as a candidate construction, it should support any sufficiently high-entropy distribution that can be used as LWE errors; see the open questions for further discussion.

In the common reference string model, our construction can be extended to satisfy a stronger security notion of reusability (analogous to the same-point composability in [FF20]), where independent obfuscations are generated for many related inputs $\{\mathbf{x}_i\}_i$ of the form $\mathbf{x}_i = \mathbf{x} + \boldsymbol{\delta}_i$ for adversarially chosen shifts $\boldsymbol{\delta}_i$. Reusability is an important feature in biometric-style applications, where the same underlying entropic secret may be used multiple times across different systems [Boy04, BA13, CFP+16].

Connections to Fuzzy Extractors. A reader may recall the close connection between fuzzy extractors [DORS08, DRS04] and fuzzy point obfuscation. A fuzzy extractor consists of a pair of algorithms $(\mathsf{Gen}, \mathsf{Rep})$. Given input $\mathbf{x}$, the generation algorithm outputs a key together with associated helper data: $(R, P) \leftarrow \mathsf{Gen}(\mathbf{x})$. Correctness requires that any input $\mathbf{x}'$ t-close to $\mathbf{x}$ enables reproduction of the same key from the public helper data: $\mathsf{Rep}(\mathbf{x}', P) = R$. For security, the extracted key R must remain indistinguishable from a uniformly random string even given P, provided that $\mathbf{x}$ is sampled from a distribution with sufficiently high entropy.

A naive attempt to build a fuzzy point obfuscator from a fuzzy extractor is to compute $(R, P) \leftarrow \mathsf{Gen}(\mathbf{x})$ and output $(R \oplus \mathbf{y}, P)$ as the obfuscated program. Given a sufficiently close input, an evaluator could use the reproduction algorithm to recover R from P, and then decrypt the payload $\mathbf{y}$.

However, this approach runs into several obstacles. First, the standard security definition for fuzzy extractors does not imply distributional virtual black-box (VBB) security. Indeed, many existing fuzzy extractor constructions inherently leak structural information about the input $\mathbf{x}$.[6] Second, the correctness guarantee of fuzzy extractors is one-sided: it ensures recovery when $\mathsf{dis}(\mathbf{x}, \mathbf{x}') \leq t$, but gives no guarantee once the distance exceeds the threshold. By contrast, to preserve the exact functionality of the (multi-bit) fuzzy point function $\mathsf{B}_{t,\mathbf{x},\mathbf{y}}(\mathbf{x}')$, we require a sharp transition at the threshold t, namely acceptance inside the ball and rejection outside it.

1.1.2 Threshold Difference Revealing Encryption

In this work, we isolate a basic missing functionality in fuzzy cryptography. We formalize this as a new primitive called Threshold Difference Revealing Encryption (TDRE); see Section 6 for the full definition and construction. Existing primitives, most notably fuzzy extractors, are well suited to *fuzzy authentication*, but they do not directly support *fuzzy search* over encrypted biometric data. The distinction is conceptual. Authentication is a fundamentally *one-input* task: after enrolling a biometric template $\mathbf{x}$, a user later proves possession of some $\mathbf{x}'$ that is close to $\mathbf{x}$. Search is inherently a *two-input* task: given two hidden biometric objects, one would like to determine whether they are close, while revealing as little as possible about either object beyond the outcome of the comparison.

A nearby line of work is property-preserving encryption, and in particular order-revealing encryption [BCLO09, BCO11, PR12, BLR+15, LW16]. At a high level, our target functionality resembles a fuzzy analogue of a property-revealing primitive. Nevertheless, this line of work is not a satisfactory fit for our setting. Beyond the mismatch in precise functionality, order-revealing

[6] This issue is also addressed by the notion of *private* fuzzy extractors [DS05]. Generic compilers that upgrade non-private fuzzy extractors to private ones have been proposed [WZ17, CHRF24], primarily via the distributional VBB obfuscation of multi-bit compute-and-compare (MBCC) programs [WZ17], but these rely on heavy cryptographic machinery such as FHE.

encryption is naturally a secret-key primitive as the public-key version suffers from a trivial attack: if public encryption and public comparison were both available, one could recover plaintexts by binary search. This is undesirable for biometric enrollment, where one would ideally like any party to encode under a public key, or even under public parameters, without relying on secret enrollment state.

This leads naturally to the following question.

Question. Can one construct a *public-key*, *reusable*, *two-input* analogue of fuzzy extractors: a primitive in which each biometric $\mathbf{x}$ is encoded independently, and from the encodings of $\mathbf{x}$ and $\mathbf{x}'$ one can determine whether $\mathrm{dist}(\mathbf{x}, \mathbf{x}') \le t$—or, more generally, recover only a narrowly prescribed output on a match—while otherwise leaking nothing else about either biometric?

To capture this functionality, we introduce a notion that we call *Threshold Difference Revealing Encryption*.

Threshold Difference Revealing Encryption. A basic TDRE scheme is a *public-key function-revealing primitive* for encrypted two-input fuzzy matching. Concretely, it allows anyone to encode an input x, and from two ciphertexts $\mathsf{ct}(x)$ and $\mathsf{ct}(x')$ one can publicly recover the *threshold difference function* $\mathcal{F}_t^{\mathrm{TD}}(x, x')$. We also consider a payload-carrying variant, in which ciphertexts additionally encode messages y, y', and public evaluation reveals a payload-augmented function $\mathcal{F}_t^{\mathrm{TDP}}((x, y), (x', y'))$. Formally:

- *Threshold Difference Function.* Given a metric space (M, dis) and threshold t, define

$$\mathcal{F}_t^{\mathrm{TD}}(x, x') = \begin{cases} x - x', & \text{if } \mathsf{dis}(x, x') \le t, \\ \perp, & \text{otherwise.} \end{cases}$$

  Thus, $\mathcal{F}_t^{\mathrm{TD}}(x, x') \neq \perp$ precisely when x and x' are t-close, so the primitive implicitly performs fuzzy matching.

- *Threshold Difference with Payload.* We augment the above with payloads y, y', and on a fuzzy match also reveal their difference:

$$\mathcal{F}_t^{\mathrm{TDP}}\big((x, y), (x', y')\big) = \begin{cases} (x - x',\ y - y'), & \text{if } \mathsf{dis}(x, x') \le t, \\ \perp, & \text{otherwise.} \end{cases}$$

  In particular, setting $y' = 0$ allows a match to conditionally release a secret y.

The strongest imaginable security guarantee for TDRE would say that, given any collection of ciphertexts, an adversary learns nothing beyond the pairwise outputs of $\mathcal{F}_t^{\mathrm{TD}}$ or $\mathcal{F}_t^{\mathrm{TDP}}$ on the underlying plaintexts. In the public-key setting, however, such a worst-case guarantee is too strong to hope for in general because of the analogue of the *residual-function attack* [HLP11]: since encryption is public, an adversary can create ciphertexts of adaptively chosen probe inputs and compare them against honest ciphertexts using the public revelation procedure. For worst-case inputs, this effectively exposes the residual threshold-difference function of the hidden plaintext and can be enough to reconstruct it.

We therefore adopt a *distributional* notion of security. As in distributional VBB obfuscation and fuzzy extractors, we assume that the sensitive feature vectors x_i are sampled from a source distribution $\mathcal{X}$, while the payloads y_i may still be chosen adversarially. Intuitively, if $\mathcal{X}$ has sufficient entropy, then an adversary cannot trivially recover a hidden biometric by encrypting arbitrary probes: a random probe is unlikely to land inside the small radius-t neighborhood that triggers revelation. Moreover, we take inspiration from distributional-VBB obfuscation and make an analogous requirement that for any predicate h, $\mathsf{ct}(x)$ should not leak $h(x)$, which models a strong protection on the privacy of the distributional biometric input x.

At first sight, revealing $x - x'$ on a match may seem like strong leakage. In many biometric settings, however, this quantity is naturally interpreted as *measurement noise* between two readings of the same underlying source. This viewpoint is also in the spirit of reusable fuzzy extractors, where security is formulated for repeated noisy measurements rather than exact copies of the same source [Boy04, CFP+16, ACEK17]. We refer the reader to Section 6.1 for detailed definitions.

The central feature of TDRE—support for computation on two ciphertexts—is not captured by standard fuzzy-extractor-based primitives and enables natural applications that currently don't have simple solutions.

As an illustrative example, we consider the scenario of *private fuzzy authentication*, where a client wishes to authenticate to a server using a biometric, but without revealing the biometric itself during the protocol. Fuzzy extractors already enable this, but require an additional round of interaction: the server sends the helper data P and a hash function H, after which the client recovers R and responds with $H(R)$. With TDRE, authentication becomes non-interactive: the server stores $\mathsf{ct}(x)$ at enrollment, the client later sends $\mathsf{ct}(x')$ for a new biometric reading x' along with a non-interactive zero knowledge argument of knowledge that $\mathsf{ct}(x')$ is computed honestly. The authenticator can verify that $\mathsf{ct}(x)$ and $\mathsf{ct}(x')$ correspond to close inputs and $\mathsf{ct}(x')$ is honestly generated. The perfect two-sided correctness of TDRE scheme is crucial to authentication: it is precisely the two-sided correctness (along with the knowledge soundness of the NIZK) that guarantees that for a successful authentication an adversary must know x' close to x. This was not possible with tools that only provides one-sided guarantee. For security, TDRE not only provides strong privacy for a single biometric input x, but also guarantees that both x and x' are still hidden even a match (i.e., successful authentication) happens. As a remark, unrolling the structure of our TDRE construction, the NIZK only needs to certify a simple statement that $\mathsf{ct}(x')$ is truly an LWE sample (proving the knowledge of the LWE secret). Therefore, this solution might actually be practical.

Besides private authentication, we propose several more advanced applications.

- *Encrypted biometric search and de-duplication.* A server stores a database $\{\mathsf{ct}(x_i, y_i)\}_i$ of encrypted biometrics with associated payloads. Given a query ciphertext $\mathsf{ct}(x, 0)$, the server can test for fuzzy matches against every record and, on a match, recover the payload y_i together with the local offset $x_i - x$. This is inherently a *two-ciphertext* operation—comparing two independently encoded, hidden biometrics—which falls outside the interface of fuzzy extractors, since the syntax of the latter requires at least one input to be presented in the clear for reproduction. Under the natural assumption that distinct individuals' biometrics are farther apart than the distance threshold, TDRE also supports fuzzy de-duplication of near-duplicate records.

- *Private biometric-assisted voting.* During enrollment, a trusted registrar records ciphertexts

$\mathsf{ct}(x_i, 0)$ for eligible voters without retaining raw biometrics. To cast a ballot, a voter submits $\mathsf{ct}(x', \mathsf{vote})$. The election authority checks whether the ballot matches an enrolled biometric and, only in that case, recovers the vote payload. Thus eligibility can be enforced without revealing identities or raw biometrics to the tallying authority. Neither fuzzy extractors nor standard property-revealing encryption support this combination of public-key enrollment, two-ciphertext comparison, and conditional payload release. Predicate encryption [BW07, KSW08, SSW09] might seem a tool enabling this application, but present constructions also fail to support the fuzzy match functionality and query privacy at the same time.

- *Private cluster discovery.* By evaluating TDRE pairwise across an encrypted dataset, a server can recover the thresholded proximity graph and local offsets among nearby records, while the absolute positions remain hidden. The resulting interesting structure may suffice for tasks such as coarse clustering, duplicate detection, or local anomaly discovery without exposing the underlying plaintext data. TDRE enriches the limited functionalities supported by present property revealing encryption, and provides simple, non-interactive solutions compared to more advanced secure MPC machinery.

1.2 Related Works

LWE/SIS with Hints A long line of work studies lattice problems augmented with hints. Boneh and Freeman [BF11] introduced k-SIS for building homomorphic signature schemes, and Ling, Phan, Stehlé, and Steinfeld [LPSS14] introduced its noisy dual analogue k-LWE, proving a reduction from standard LWE with polynomial loss in k and applying it to traitor tracing. Their hints are $k < m$ short vectors of $\mathcal{L}_q^\perp(\mathbf{A})$, which span only a proper sublattice (m linearly independent ones would already form a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$). A second family instead reveals short preimages of *nonzero, structured* targets, or a trapdoor for a *structured matrix related to* $\mathbf{A}$. It includes k-R-ISIS and k-M-ISIS [ACL+22] (here k may far exceed m) and their variants, used in lattice-based SNARKs and commitments [CLM23, FLV23, BCFL23]; the BASIS assumptions [WW23b]; succinct SIS and LWE [Wee24], the latter implying decomposed LWE [AMR25]; and evasive LWE [Wee22, Tsa22, VWW22]. Their hardness rests on the structure of these hints: only special cases (e.g., BASIS with random matrices [WW23b] and 1-succinct LWE [Wee24]) are known to follow from SIS or LWE, and there is evidence against the knowledge variant of k-R-ISIS [WW23a] and attacks or counterexamples for several variants of evasive LWE [BÜW24, AMYY25, DJM+25, HJL25]. In contrast, our LWE with Trapdoor Hints problem reveals a *full* homogeneous trapdoor for the LWE matrix $\mathbf{A}$ itself, so no structure can prevent the textbook trapdoor attack. Its hardness demands a fundamentally different proof strategy.

The only other work we are aware of in this full-trapdoor setting is the independent and concurrent work of [ALP26] on the SIS side, which bases the hardness of *hinted ISIS* (ISIS given short vectors that may span $\mathcal{L}_q^\perp(\mathbf{A})$) on unstructured lattice assumptions. Their hint-to-challenge gap is as small as a constant, but their reductions run in single-exponential time with polynomial memory, and thus rely on the conjectured space–time hardness of lattice problems, namely that these problems cannot be solved in single-exponential time using only polynomial memory. Moreover, their reduction to SIS without hints needs a *chain* of hinted-ISIS algorithms across multiple parameter settings rather than a single one. In contrast, our work targets LWE—the injective rather than surjective regime, relies on standard polynomial-time reductions, and our reduction only requires

a single adversary for a fixed parameter setting.

Finally, a separate line of work studies LWE with leakage on the error (and possibly the secret). Present results include that decision LWE remains hard given exact linear hints $\langle \mathbf{e}, \mathbf{z} \rangle$ [AP12, BLP+13, AA16] or noisy linear leakage [OPW11, LKK+18, DKL+23, KLSS23, LSW25], and that *search* (M-)LWE remains hard with arbitrary bounded-range leakage that leaves enough min-entropy under suitable parameter conditions [BJTW25]. We prove *decision* hardness under a *short* (up to $O(\log \lambda)$ bits) but *arbitrary* leakage $h(\mathbf{e})$, even given a full trapdoor for $\mathbf{A}$, which may be of independent technical interest.

Obfuscation Some previous works on obfuscation that target related functionalities include obfuscation of compute-and-compare programs/lockable obfuscation [WZ17, GKW17], obfuscated Hamming distance fuzzy matching [GZ19], etc. Compute-and-compare programs are a strong extension of point functions and can be extremely powerful, but present constructions involve heavy machinery including fully/leveled homomorphic encryption and have complex encoding scheme, making them far from practical. In comparison, our construction, though specific to Euclidean-distance fuzzy point functions, is much cleaner and more efficient. The construction of Galbraith and Zobernig [GZ19] works with fuzzy matching according to Hamming distance instead of Euclidean distance, and it does not support multi-bit output. Moreover, they rely on a novel subset-product-style pseudorandomness assumption, while ours is based on standard lattice-based assumptions.

Fuzzy Extractors Fuzzy cryptography develops tools to work directly with entropic but noisy input sources such as biometrics and human generated data, physically unclonable functions, and quantum information. Among this family, the most typical tool is fuzzy extractors [DRS04, Boy04, DKRS06, FMR13, CFP+16, FRS16, ACEK17, WLH18, WL18, DFR21, ACF+22, CHRF24]. They enable a variety of tasks including key agreement, authentication, identification and decryption, etc.

In fuzzy extractors, to reproduce the secret, the input, i.e., the second biometric reading, must be given *in the clear*. In biometric applications where the inputs (e.g., fingerprints, iris scans) are highly sensitive and cannot be revoked or regenerated, it is highly desirable that inputs only appear in the clear in trusted domains, limiting the use cases of fuzzy extractors. In contrast, with our new primitive TDRE, computation on encrypted inputs can happen in untrusted domains, revealing only the threshold difference. In addition, our construction works with Euclidean metric, which is seldom covered by previous works.

As an additional contribution, our hardness results for LWE with Trapdoor Hints directly imply a computational fuzzy extractor using the template in [FMR13] and, moreover, resolve an open question raised in their paper. There, the authors observed that although a full lattice trapdoor may improve the error tolerance of their computational fuzzy extractor construction, it could not be put into the construction of a fuzzy extractor, because there was no secret space to store the trapdoor to pass it to the reproduction process. Our results show that the trapdoor can instead be made public directly, provided the inputs are drawn from a Gaussian distribution with sufficiently large width. Our technique drastically improves the decoding capability by supporting Euclidean metric and a super-exponentially larger decoding ball.

Other Advanced Encryption Schemes Related notions include predicate encryption [BW07, KSW08, SSW09], function-hiding inner-product encryption [BJK15, KLM+18], property/function revealing encryption [PR12, BLR+15, JP18], etc. Differences mainly lie in functionalities achieved and the security models. The Euclidean fuzzy matching (threshold difference revealing) functionality achieved in our work is unsupported by most previous works, unless resorting to heavy machinery such as FHE [GVW15]. In terms of security, our TDRE scheme naturally hides not only the attributes and payloads but also the predicates/functions. Previous works either restrict security guarantees or work in secret-key settings to circumvent inherent obstacles such as "residual function" attacks. We chart another path by considering public-key settings but leveraging distributional inputs.

1.3 Open Questions

A straightforward and important question is whether we can further improve our LWE with Trapdoor Hints hardness results in several dimensions. Firstly, the current gap between the "security regime" and the "decoding regime" is roughly a polynomial factor of $\tilde{O}(m^{1.5})$. Namely, when the product of the error's and the trapdoor's Gaussian widths satisfies $\sigma_e \cdot \sigma_t = o(q/\sqrt{m})$, then with high probability LWE will be broken; while our hardness results ensure security when $\sigma_e \cdot \sigma_t > q \cdot m \cdot \omega(\log \lambda)$ (under certain constraints). However, we conjecture that the security can even hold when $\sigma_e \cdot \sigma_t = (q/\sqrt{m}) \cdot \omega(\log \lambda)$, i.e., the gap can be as small as $\tilde{O}(1)$. In this regime, current LWE attacks do not seem to be boosted with the help of this Gaussian random full trapdoor as hints, and we believe the "textbook LWE attack" using full homogeneous trapdoor is essentially the best one can do. It remains intriguing whether it is possible to further reduce this gap with formal proof and reductions, or prove a non-trivial lower bound for it.

Secondly, it would be more satisfying if we can generalize our LWE with Trapdoor Hints results to other error distributions under which LWE is conjectured to be hard. Currently, our reduction as well as the entropic LWE results we use technically rely on Gaussian distributions due to its desirable properties. However, we conjecture the hardness to hold for more other LWE-compatible error distributions.

In addition, the restriction that the modulus q should be close to some τ^c for $c := \lceil \log_\tau q \rceil$ in Theorem 1.1, although mild, seems unnatural. We will later show in the analysis that this is inherently required if we choose to work with the specific trapdoor sampling scheme of [MP12]. However, it remains open whether this condition can be removed in our final unstructured version of LWE with Trapdoor Hints with random Gaussian trapdoor.

On the application side, the improvements mentioned above on the LWE with Trapdoor Hints hardness results will directly result in better parameters and wider applicability for our Euclidean ball obfuscation and TDRE constructions. It remains open whether more powerful cryptographic primitives can be constructed using this LWE with Trapdoor Hints paradigm. Moreover, the TDRE primitive itself also serves as a point of interest. There exist some caveats for the current formulation and construction: e.g., it would be better if we could construct a scheme that only reveals a bit indicating a fuzzy match, instead of the difference. Alternative efficient construction recipes, better security guarantees, and more real-world application scenarios for TDRE are also waiting to be explored.

2 A Brief Overview of Techniques

We give an informal overview of the proof ideas and techniques underlying our main results. We begin with our core theoretical contribution, namely the hardness result for LWE with Trapdoor Hints (Theorem 1.1). At a high level, our goal is to show

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q) \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}), \tag{1}$$

where $\mathbf{A}$ is a random matrix over $\mathbb{Z}_q$, $\mathbf{s}$ is a random vector over $\mathbb{Z}_q$, $\mathbf{e}$ is a Gaussian random error vector, and $\mathbf{T}$ is a random trapdoor for $\mathbf{A}$, sampled from a discrete Gaussian over $\mathcal{L}_q^\perp(\mathbf{A})$ with respect to a certain width parameter.

To obtain a mathematical handle on this problem, we take inspiration from the trapdoor sampling scheme of [MP12] and attempt to prove the following stronger claim:

$$(\mathbf{A}, \mathbf{R}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q) \approx_c (\mathbf{A}, \mathbf{R}, \mathbf{u}^\mathsf{T}), \tag{2}$$

where we now take $\mathbf{A}$ to have the structured form $[\mathbf{B} \| \mathbf{BR} + \mathbf{G}_\tau]$, with $\mathbf{B}$ a random matrix over $\mathbb{Z}_q$ and $\mathbf{R}$ a random 0/1-matrix. The matrix $\mathbf{G}_\tau$ is a public gadget matrix with base τ and the following special structure:

$$\mathbf{G}_\tau := \begin{bmatrix} \mathbf{g}^\mathsf{T} & & & \\ & \mathbf{g}^\mathsf{T} & & \\ & & \ddots & \\ & & & \mathbf{g}^\mathsf{T} \end{bmatrix}$$

where $\mathbf{g} := [1 \ \tau \ \cdots \ \tau^{c-1}]^\mathsf{T}$ for $c = \lceil \log_\tau q \rceil$. A point worth noting is that [MP12] and most subsequent works default to $\tau := 2$, as this gives the best trapdoor quality. However, it is precisely the use of a larger τ that makes our proof possible. Below, we will also drop the subscript τ in $\mathbf{G}_\tau$ to reduce notational overhead.

As shown in [MP12], one can sample a Gaussian trapdoor $\mathbf{T}$ for $\mathbf{A}$ using $\mathbf{R}$, and the structured matrix $\mathbf{A} := [\mathbf{B} \| \mathbf{BR} + \mathbf{G}]$ can be statistically close to uniformly random. Therefore, proving the second claim, with structured $\mathbf{A}$ and $\mathbf{R}$ in place of $\mathbf{T}$, suffices for our original goal.

Expanding the LWE sample term, we get :

$$\begin{aligned} \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} &= \mathbf{s}^\mathsf{T}[\mathbf{B} \| \mathbf{BR} + \mathbf{G}] + \mathbf{e}^\mathsf{T} \\ &= [\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T} \| \mathbf{s}^\mathsf{T}\mathbf{BR} + \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}]. \end{aligned}$$

Given that $\mathbf{R}$ is public, an easy observation is that, to prove $\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T}$ is indistinguishable from uniform, it is equivalent to show $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T}$ and $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$ are both indistinguishable from uniform even given $\mathbf{A}$ and $\mathbf{R}$. While the first term looks safe, the second term might be a source of caution. Indeed, when $\mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$ is small enough in norm, the gadget $\mathbf{G}$ serves as an error correcting code allowing us to decode the codeword $\mathbf{s}^\mathsf{T}\mathbf{G}$ from the noise, which is exactly how $\mathbf{R}$ serves as a trapdoor and enables LWE inversion.

This is where properly tuning τ becomes important. Increasing τ reduces the decoding radius of $\mathbf{G}$. Furthermore, we can set $\mathbf{e}$ to be larger in norm than this threshold.

For our proof strategy, we must address two core issues:

1. First, we should at least show that $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ is statistically close to random and lossy in $\mathbf{s}$,

2. Second, we must also deal with an issue of dependency. To process $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T}$ we have to deal with the fact that the second term $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$ depends on both $\mathbf{s}$ and $\mathbf{e}_1$.

We now describe how to deal with the above two issues:

Issue 1: Can $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ be close to uniform random? To better understand this, it is instructive to consider the following basic attack. Given a sample of the form $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$, one could try to find a short vector $\mathbf{v}$ in the orthogonal lattice $\mathcal{L}_q^\perp(\mathbf{G})$, and then multiply the sample by $\mathbf{v}$, which eliminates the secret related part and leaves $\mathbf{e}_2^\mathsf{T}\mathbf{v} \bmod q$. If $\mathbf{e}_2$ is sufficiently small, then $\mathbf{e}_2^\mathsf{T}\mathbf{v} \bmod q = \mathbf{e}_2^\mathsf{T}\mathbf{v}$ over the reals, making the resulting value easily distinguishable from uniform random.

Note that it is trivial to find a vector of ℓ_∞-norm $O(\tau)$ in the kernel of $\mathbf{G}$ (over $\mathbb{Z}_q$). This already yields an attack whenever $\|\mathbf{e}\| \ll \frac{q}{\tau\sqrt{n \cdot c}}$. We show that, under certain conditions, this is essentially all that can be done. Namely, that $\mathbf{G}$ does not have any smaller vectors in its kernel.

To formalize this intuition and prove that $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ is statistically close to uniform, we bound the smoothing parameter of the lattice $\mathcal{L}_q(\mathbf{G}^\mathsf{T})$ (see Lemma 3.9). Recall that the smoothing parameter is defined in terms of the dual lattice and controls when a Gaussian reduced modulo the lattice becomes statistically close to uniform [MR04, Definition 3.1, Lemma 4.1]. Therefore, once the Gaussian width of $\mathbf{e}_2$ exceeds $\eta_\varepsilon(\mathcal{L}_q(\mathbf{G}^\mathsf{T}))$, the distribution $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} \bmod q$ is statistically close to uniform (see Lemma 4.2).

Quantitatively, we show the following lemma on the upper bound of $\mathcal{L}_q(\mathbf{G}^\mathsf{T})$'s smoothing parameter (the formal proof is at Lemma 3.9):

Lemma 2.1. *Let $\tau \geq 2$ be a positive integer and $q > \tau$ be a prime modulus. Let $c = \lceil \log_\tau q \rceil$ and dimensions $n > 0$, $w := n \cdot c$. For any function $h(w) = \omega\left(\sqrt{\log w}\right)$, there exists a negligible $\varepsilon = \mathsf{negl}(w)$ s.t.*

$$\eta_\varepsilon(\mathcal{L}_q(\mathbf{G}^\mathsf{T})) \leq \tau^{c-1} h(w) \,.$$

Note that this lemma also implies a technical condition on the modulus. According to the definitions, we have $\tau^{c-1} < q < \tau^c$. To achieve tight results discussed above, $\Omega(\tau^c) < q < \tau^c$ is needed. This requirement is not merely an artifact of the proof, but is also necessary. For example, suppose $q = \tau^{c-1} + 1$, then $\mathbf{v} = [1\ 0\ \ldots\ 0\ 1]^\mathsf{T}$ satisfies

$$\mathbf{g}^\mathsf{T}\mathbf{v} = \tau^{c-1} + 1 \equiv 0 \pmod q,$$

so $[\mathbf{v}^\mathsf{T}\ \cdots\ \mathbf{v}^\mathsf{T}]^\mathsf{T} \in \mathcal{L}_q^\perp(\mathbf{G})$, whose ℓ_∞-norm is well below the desired bound $O(\tau)$. Such an exceptionally short vector directly yields an attack.

Issue 2: Can $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T}$ be pseudorandom given $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$? Assuming we set the parameters such that $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ is marginally random, we must still address the obstacle that the leakage term, $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$, depends on both $\mathbf{s}$ and $\mathbf{e}_1$.

While our main result uses LWE with polynomial ratio to argue this (assuming τ and q are polynomial), we show a simpler argument assuming LWE with super-polynomial ratio. The idea is that to remove the dependency on $\mathbf{e}_1$, one can employ error flooding techniques. By setting both modulus q and $\mathbf{e}_1$ to be superpolynomial, we decompose $\mathbf{e}_1 \approx \mathbf{e}_3 + \mathbf{e}_1'$ for a dominant component $\mathbf{e}_1'$ and a superpolynomially smaller, independent component $\mathbf{e}_3$ overwhelmed by $\mathbf{e}_1'$. By substituting, we get $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R} = \mathbf{s}^\mathsf{T}\mathbf{G} + (\mathbf{e}_2^\mathsf{T} - \mathbf{e}_3^\mathsf{T}\mathbf{R}) - (\mathbf{e}_1')^\mathsf{T}\mathbf{R}$. Note that $\mathbf{e}_2$, which is as large as $\mathbf{e}_1$, also

floods $\mathbf{e}_3^\mathsf{T}\mathbf{R}$ and thus $(\mathbf{e}_2^\mathsf{T} - \mathbf{e}_3^\mathsf{T}\mathbf{R})$ can be switched with $\mathbf{e}_2^\mathsf{T}$ at the expense of negligible statistical loss. Furthermore, we can directly publish $\mathbf{e}_1'$, which results in an even stronger leakage, i.e., $(\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T}, \mathbf{e}_1')$. As a consequence of these changes, it suffices to prove that $\mathbf{s}^\mathsf{T}\mathbf{B}+\mathbf{e}_3^\mathsf{T}$ is pseudorandom given $\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T}$, which is solely a leakage on $\mathbf{s}$.

To make the argument proceed, we show that $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ is not only statistically close to uniform, but it is also lossy in $\mathbf{s}$. We show that $\mathbf{s}$ still has approximately n bits of min-entropy even given $\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T}$ [7] (see Lemma 4.1). We can therefore invoke the hardness of *entropic LWE* [GKPV10, BD20] to conclude that $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_3^\mathsf{T}$ is pseudorandom, assuming the hardness of standard LWE with dimension $n/\log q$.

Though the description above required LWE with subexponential ratio for error flooding, in our actual statement and proof, this problem is circumvented. The main idea for enabling this is to take advantage of the statistical properties of Gaussian distributions. Namely, we utilize the (de-)composability of discrete Gaussian distributions [Pei10, GMPW20] (see Lemma 3.8) to carry out error smudging with much better parameters, for which poly-ratio LWE suffices. Instead of flooding $\mathbf{e}_3^\mathsf{T}\mathbf{R}$ by $\mathbf{e}_2$, we write the discrete Gaussian error as $[\mathbf{e}_1^\mathsf{T}\|\mathbf{e}_2^\mathsf{T}] \approx_{st} [\mathbf{e}_3^\mathsf{T} + (\mathbf{e}_1')^\mathsf{T}\|\mathbf{e}_3^\mathsf{T}\mathbf{R} + (\mathbf{e}_2')^\mathsf{T}]$ for a discrete Gaussian $\mathbf{e}_3$ and an independent (non-spherical) discrete Gaussian $[(\mathbf{e}_1')^\mathsf{T}\|(\mathbf{e}_2')^\mathsf{T}]$. Then $\mathbf{e}_3^\mathsf{T}\mathbf{R}$ cancels exactly in $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} - \mathbf{e}_1^\mathsf{T}\mathbf{R}$, and the decomposition only requires the width of $\mathbf{e}_1$ and $\mathbf{e}_2$ to exceed that of $\mathbf{e}_3$ by about $\|\mathbf{R}\|$, rather than by a superpolynomial factor. We omit the details in this overview.

Another technical detail we have not covered is that on top of Eq. (1), we furthermore show that the secret $\mathbf{s}$ can be used for extra safe LWE samples $(\mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)$, i.e.,

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q) \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, \mathbf{C}, (\mathbf{u}')^\mathsf{T}). \tag{3}$$

This enables us to use the extra LWE samples to encrypt some message payload in our applications. The extension is not technically challenging and can be achieved by integrating the extra samples in our proof. We refer the readers to Section 4 for details.

2.1 From LWE with Trapdoor Hints to Applications

The main applications we propose for our hardness result of LWE with Trapdoor Hints are Euclidean-ball obfuscation and the new primitive Threshold Difference Revealing Encryption (TDRE). We refer the reader to Sections 5 and 6 for formal definitions of these primitives. We first provide a brief intuition on their construction from LWE with Trapdoor Hints, deferring the explanation of how they achieve stronger simulation security to the next subsection.

Constructing Euclidean Ball Obfuscation Recall that the goal of Euclidean-ball obfuscation is to obfuscate a center point $\mathbf{x} \in \mathbb{Z}^m$ together with a message payload $\mathbf{y} \in \{0,1\}^l$, so that, given the obfuscation, one can correctly evaluate the associated fuzzy point function: on input $\mathbf{x}'$, it outputs $\mathbf{y}$ if $\|\mathbf{x} - \mathbf{x}'\| \le t$, and $\perp$ otherwise, for some distance threshold t. At the same time, the obfuscation itself should not leak either $\mathbf{x}$ or $\mathbf{y}$. Consider the following simple candidate obfuscation:

$$\mathsf{Obf}(\mathbf{x}, \mathbf{y}) := (\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}^\mathsf{T} \bmod q).$$

[7] In fact, the exact claim we show in Lemma 4.1 is that the average conditional min-entropy of $\mathbf{s}$ given both $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T}$ and $\mathbf{s} + \mathbf{e}$, for some Gaussian noise $\mathbf{e}$, is approximately n bits. This is known as the "noise lossiness" defined in [BD20] and enables us to utilize the tightest entropic LWE result.

If an evaluator holds a nearby input $\mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\| \leq t$, and if t lies within the decoding radius of the trapdoor $\mathbf{T}$, then it can compute

$$(\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T}) - (\mathbf{x}')^\mathsf{T} = \mathbf{s}^\mathsf{T}\mathbf{A} + (\mathbf{x} - \mathbf{x}')^\mathsf{T}.$$

The term $(\mathbf{x} - \mathbf{x}')$ now acts as a small LWE error, allowing the evaluator to decode $\mathbf{s}$ and $\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}'$ using $\mathbf{T}$, and hence recover the message payload $\mathbf{y}$ using $\mathbf{s}$. Moreover, with overwhelming probability over $\mathbf{A}$, $\mathcal{L}_q(\mathbf{A}^\mathsf{T})$ contains no nonzero vector with norm below q (Lemma 3.2), in which case any input $\mathbf{x}'$ satisfying $\|\mathbf{x}-\mathbf{x}'\| > t$ can be rejected provided that all inputs have norm less than $(q-t)/2$ (Remark 3.3). Indeed, if one obtains $(\mathbf{s}', \boldsymbol{\delta}')$ such that $\|\boldsymbol{\delta}'\| \leq t$ and $\mathbf{A}^\mathsf{T}\mathbf{s} + (\mathbf{x}-\mathbf{x}') = \mathbf{A}^\mathsf{T}\mathbf{s}' + \boldsymbol{\delta}'$, then $\mathcal{L}_q(\mathbf{A}^\mathsf{T})$ contains $\mathbf{x} - \mathbf{x}' - \boldsymbol{\delta}'$, whose norm is below $q - t + t = q$, so $\mathbf{x} - \mathbf{x}' = \boldsymbol{\delta}'$ has norm at most t. Therefore, the obfuscation scheme preserves the functionality of the fuzzy point function.

Conversely, assuming that $\mathbf{x}$ is sampled from a sufficiently wide Gaussian distribution, LWE with Trapdoor Hints implies that the obfuscation is computationally indistinguishable from $(\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, \mathbf{C}, (\mathbf{u}')^\mathsf{T})$ to anyone who does not know such a nearby $\mathbf{x}'$. Moreover, given only the obfuscation, finding $\mathbf{x}$, or even finding a nearby $\mathbf{x}'$, should be hard, while the payload $\mathbf{y}$ remains completely hidden.

The connection to fuzzy cryptography is immediate. First, our obfuscation yields a computational fuzzy extractor for Euclidean distance in the sense of [DORS08, FMR13], with perfect two-sided correctness. The work [FMR13] mentioned the idea of instantiating a code-offset construction with a fine-tuned trapdoor but was blocked by the lack of security guarantees. Without the trapdoor $\mathbf{T}$, the work could only decode from a very small Hamming radius of $O(\log n)$. Our work drastically improves the decoding capability for the [FMR13] approach by supporting Euclidean metric and a polynomial (or even superpolynomial) decoding radius, resulting in a decoding ball that can be super-exponentially larger in volume.

Fuzzy Cryptography Tool Operating on Two Encodings Beyond fuzzy extraction, however, our LWE with Trapdoor Hints results also enable a new fuzzy-search functionality: two encodings of nearby inputs—for example, two readings of the same person's biometrics—reveal only whether the inputs match and, in the matching case, the difference between them (corresponding to the measurement noise), while hiding all other information. This functionality is captured exactly by our new primitive TDRE. In our construction, we take $\mathbf{A}$ and $\mathbf{T}$ to be public parameters. An encoding of $\mathbf{x}$ is simply

$$\mathsf{ct} := \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q,$$

and similarly an encoding of $\mathbf{x}'$ is $\mathsf{ct}' := (\mathbf{s}')^\mathsf{T}\mathbf{A} + (\mathbf{x}')^\mathsf{T} \bmod q$.

The key observation enabling this construction is that, given ct and ct', one can compute their difference:

$$\mathsf{ct}' - \mathsf{ct} = (\mathbf{s}' - \mathbf{s})^\mathsf{T}\mathbf{A} + (\mathbf{x}' - \mathbf{x})^\mathsf{T}.$$

If $\mathbf{x}$ and $\mathbf{x}'$ are close, then the trapdoor $\mathbf{T}$ can decode both $(\mathbf{x}' - \mathbf{x})$ and $(\mathbf{s}' - \mathbf{s})$. To simulate a match, a simulator can generate the second encoding simply as

$$\widetilde{\mathsf{ct}}' = \mathsf{ct} + \Delta\mathbf{s}^\mathsf{T}\mathbf{A} + \Delta\mathbf{x}^\mathsf{T},$$

where the random $\Delta\mathbf{s}$ is used to simulate $(\mathbf{s}' - \mathbf{s})$, and $\Delta\mathbf{x} := \mathbf{x}' - \mathbf{x}$. Since ct is pseudorandom by LWE with Trapdoor Hints, this completely hides the absolute positions of the inputs. As in

the Euclidean-ball obfuscation setting, message payloads can also be supported by including additional LWE samples, namely by defining

$$\mathsf{ct} := (\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}^\mathsf{T} \bmod q),$$

while also including $\mathbf{C}$ among the public parameters.

2.2 Achieving Strong Simulation Security with Leakage

Since one of the primary targets for these applications is biometric cryptography, which is sensitive to privacy, it is very important to work with a strong notion of security. Moreover, formulating security notion with regard to distributional inputs can be tricky. Take the obfuscation as an example. In the case without message payload, it would be vacuous to define security as $\mathsf{Obf}(\mathbf{x})$ being simulatable without access to $\mathbf{x}$, as $\mathbf{x}$ itself comes from a distribution as a simulator can simply sample from the same distribution and run the honest algorithm. Moreover, the security notion should capture the requirement that any additional information on the input $\mathbf{x}$ should not be leaked.

To address this, we consider the *(distributional) virtual black-box (VBB)* security, which explicitly requires that for any one-bit predicate h, $\mathsf{Obf}(\mathbf{x})$ does not reveal any more about $h(\mathbf{x})$ than the black-box access to the fuzzy point function with center point $\mathbf{x}$ does. This in turn can be achieved if we show one can simulate consistently $\mathsf{Obf}(\mathbf{x})$ with this one-bit auxiliary information $h(\mathbf{x})$, i.e., for any predicate h,

$$(\mathsf{Obf}(\mathbf{x}), h(\mathbf{x})) \approx_c (\mathsf{Sim}(1^\lambda), h(\mathbf{x})).$$

We also consider a similar style definition for our TDRE.

As a remark, our constructions support some sort of reusability [Boy04, CFP+16, FF20]. Even if one sees multiple obfuscation results/encodings of TDRE generated with respect to points related to $\mathbf{x}$ (modeled by $\mathbf{x}$ shifted by adversarial shifts), $\mathbf{x}$ are provably hidden. This is captured in our definition of TDRE in Definition 6.3. For Obfuscation, we only consider the standalone definition, but remark how it can be upgraded to satisfy this notion of reusability (see Remark 5.2), as it simply follows from prior techniques [ACEK17].

For proving leakage resilience as required in a distributional-VBB secure obfuscation scheme, we prove the following strengthened version of our LWE with Trapdoor Hints result in the existence of an arbitrary predicate leakage on the error (see the formal theorem at Theorem 4.2):

Theorem 2.1 (Informal)**.** *With regard to the same settings as in Theorem 1.1, we additionally have that for any one-bit predicate h,*

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, h(\mathbf{e})) \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, h(\mathbf{e}))$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_e}^m$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and $\mathbf{T}$ is a Gaussian random full trapdoor sampled in the lattice $\mathcal{L}_q^\perp(\mathbf{A})$ with width parameter σ_t.

This theorem can be strengthened to handle longer (bounded) leakage functions as well, possibly by modifying our proof or using a reduction that runs for a longer time.

The technical route for achieving Theorem 2.1 begins with the observation that such a short leakage $h(\mathbf{e})$ barely affects the hardness of the *search* problem, as the adversary can guess the value

of $h(\mathbf{e})$ itself. On top of this, it is not hard to show a search-to-decision reduction for standard LWE in the presence of auxiliary information on the secret $\mathbf{s}$ and the error $\mathbf{e}$ (see Lemma 4.3), which basically establishes that

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e})) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}, \mathsf{aux}(\mathbf{s}, \mathbf{e})) \tag{4}$$

as long as it is hard to find $\mathbf{s}$ (or $\mathbf{e}$) given $(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e}))$. This step utilizes a version of Goldreich-Levin theorem over large fields [DGK+10].

Relying on this basic lemma, we extend the idea of the search-to-decision reduction to the case with a public trapdoor $\mathbf{T}$. To make progress, we also heavily use the hardness of trapdoor LWE and trapdoor extension techniques [CHKP10, MP12], which intuitively allows us to separate the trapdoor from part of the LWE sample. The main point is to decompose $\mathbf{A} := [\mathbf{A}_L \| \mathbf{A}_R]$ and $\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} = [\mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T} \| \mathbf{s}^\mathsf{T}\mathbf{A}_R + \mathbf{e}_R^\mathsf{T}]$, and $\mathbf{A}$'s trapdoor $\mathbf{T}_\mathbf{A}$ can be generated using $\mathbf{A}_L$'s trapdoor $\mathbf{T}_{\mathbf{A}_L}$. Therefore, it suffices to substitute $\mathbf{T}_\mathbf{A}$ with $\mathbf{T}_{\mathbf{A}_L}$. Next, let

$$\mathsf{aux}(\mathbf{s}, \mathbf{e}_R) := (\mathbf{A}_L, \mathbf{T}_{\mathbf{A}_L}, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R)),$$

and by the hardness of vanilla LWE with Trapdoor Hints it is hard to find $\mathbf{s}$ given $(\mathbf{A}_R, \mathbf{s}^\mathsf{T}\mathbf{A}_R + \mathbf{e}_R^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e}_R))$. Therefore, $\mathbf{s}^\mathsf{T}\mathbf{A}_R + \mathbf{e}_R^\mathsf{T}$ can be switched to a random $\mathbf{u}_R^\mathsf{T}$ by Eq. (4). Next, we switch $\mathbf{T}_\mathbf{A}$ to be generated using $\mathbf{A}_R$'s trapdoor $\mathbf{T}_{\mathbf{A}_R}$ instead, and we are essentially left with proving

$$\begin{aligned}&(\mathbf{A}_L, \mathbf{A}_R, \mathbf{T}_{\mathbf{A}_R}, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T}, \mathbf{u}_R^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R)) \\ &\approx (\mathbf{A}_L, \mathbf{A}_R, \mathbf{T}_{\mathbf{A}_R}, \mathbf{u}_L^\mathsf{T}, \mathbf{u}_R^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R)).\end{aligned}$$

Stripping off the unrelated terms, this is equivalent to

$$(\mathbf{A}_L, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R)) \approx_c (\mathbf{A}_L, \mathbf{u}_L^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R)),$$

which again straightforwardly follows Eq. (4).

Above, we crucially used the fact that $\mathbf{T}_\mathbf{A}$ is a Gaussian random trapdoor that can be simulated in different ways, which however may be undesirable for applications that need transparent setup. The problem with the "public-coin" version of LWE with Trapdoor Hints, i.e., with the inhomogeneous trapdoor $\mathbf{R}$, is that $\mathbf{R}$ reveals more structure about the LWE sample compared to a Gaussian trapdoor $\mathbf{T}$, possibly resulting in some leakage on the error vector. It remains open whether one can show this strengthened security under error leakage for LWE with Trapdoor Hints with inhomogeneous trapdoor $\mathbf{R}$, which supports transparent setup.

3 Preliminaries

The log function defaults to base-2 logarithm without specification. The security parameter in this paper will be λ by default. We say that a function $f(\lambda)$ is negligible if for any constant $c > 0$, there exists $N > 0$ such that for all $\lambda > N$, $f(\lambda) < \lambda^{-c}$. We sometimes denote an unspecified negligible function by negl.

For simplicity, we use the same symbol for a distribution and its probability mass function. For two distributions $\mathcal{X}$ and $\mathcal{Y}$, we use $\mathcal{X} \cong \mathcal{Y}$ to denote that they are identically distributed. The statistical distance of $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\Delta(\mathcal{X}, \mathcal{Y}) := \frac{1}{2}\sum_{d \in D} |\mathcal{X}(d) - \mathcal{Y}(d)|$ for two distributions $\mathcal{X}$ and $\mathcal{Y}$ supported on a discrete set D.

Two ensembles of distributions $\{\mathcal{X}_\lambda\}$ and $\{\mathcal{Y}_\lambda\}$ are *statistically indistinguishable*, denoted by $\mathcal{X} \approx_{st} \mathcal{Y}$, if $\Delta(\mathcal{X}_\lambda, \mathcal{Y}_\lambda) \leq \mathsf{negl}(\lambda)$, i.e., their statistical distance is negligible. For a distinguishing algorithm $\mathcal{A}$, define its advantage in distinguishing $\{\mathcal{X}_\lambda\}$ and $\{\mathcal{Y}_\lambda\}$ as

$$\left| \Pr_{z \leftarrow \mathcal{X}_\lambda} \left[\mathcal{A}(1^\lambda, z) = 1\right] - \Pr_{z \leftarrow \mathcal{Y}_\lambda} \left[\mathcal{A}(1^\lambda, z) = 1\right] \right|.$$

We say $\{\mathcal{X}_\lambda\}$ and $\{\mathcal{Y}_\lambda\}$ are *computationally indistinguishable*, denoted by $\mathcal{X} \approx_c \mathcal{Y}$, if for every probabilistic polynomial-time distinguisher $\mathcal{A}$, its advantage in distinguishing $\{\mathcal{X}_\lambda\}$ and $\{\mathcal{Y}_\lambda\}$ is $\mathsf{negl}(\lambda)$.

The main entropy notion that will be used in this paper is the min-entropy. Let X be a discrete random variable sampled from a distribution $\mathcal{X}$ supported on the set S. Then the min-entropy of the random variable X (sampled from the distribution $\mathcal{X}$) is defined to be

$$H_\infty(X) = H_\infty(\mathcal{X}) := -\log\left(\max_{x \in S} \Pr\left[X = x\right]\right).$$

We will also consider the notion of conditional min-entropy, where the conditioned terms can include both events and random variables. Specifically, For an event E and a random variable Y, we define the average conditional min-entropy of X conditioned on E and Y to be

$$\widetilde{H}_\infty(X|E, Y) := -\log\left(\mathop{\mathbb{E}}_{y \sim \mathcal{Y}}\left[\max_{x \in S} \Pr\left[X = x | E, Y = y\right]\right]\right),$$

where $\mathcal{Y}$ denotes the (marginal) distribution of Y conditioned on E.

We denote the set of real numbers by $\mathbb{R}$ and the integers by $\mathbb{Z}$. For a positive modulus q and a real or integer number x, $x \bmod q$ is defined to be the unique $0 \leq y < q$ satisfying $y = x + q \cdot z$ for some $z \in \mathbb{Z}$. For a positive integer q, $\mathbb{Z}_q := \{x \bmod q : x \in \mathbb{Z}\}$. By default, $\|\mathbf{x}\|$ without subscripts denotes ℓ_2-norm for a vector $\mathbf{x}$ and $\|\mathbf{A}\|$ denotes the spectral norm (the largest singular value) of $\mathbf{A}$, i.e., $\|\mathbf{A}\| := \max_{\mathbf{x}} \{\|\mathbf{A}\mathbf{x}\| / \|\mathbf{x}\|\}$. For an n-dimensional vector $\mathbf{x} = [x_1, \ldots, x_n]$ and a subset $T \subseteq \{1, \ldots, n\}$, we use the shorthand notation $\mathbf{x}_T$ to denote vector $\mathbf{x}$ restricted to the indices in T (e.g. $\mathbf{x}_{[1\ldots k]} := [x_1, \ldots, x_k]$).

3.1 Goldreich-Levin for Large Fields

In this work, we will use the classical result of Goldreich-Levin theorem on large fields presented in [DGK+10].

Theorem 3.1 ([DGK+10, Theorem 1])**.** *Let q be a prime and H be an arbitrary subset of $\mathbb{Z}_q$. Let $f : H^n \to \{0,1\}^*$ be any (possibly randomized) function. If there is a distinguisher $\mathcal{D}$ that runs in time t such that*

$$\left| \Pr\left[\mathcal{D}(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s}\rangle) = 1 \;\middle|\; \begin{array}{l} \mathbf{s} \xleftarrow{\$} H^n \\ y \leftarrow f(\mathbf{s}) \\ \mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n \end{array}\right] - \Pr\left[\mathcal{D}(y, \mathbf{r}, u) = 1 \;\middle|\; \begin{array}{l} \mathbf{s} \xleftarrow{\$} H^n \\ y \leftarrow f(\mathbf{s}) \\ \mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n, u \xleftarrow{\$} \mathbb{Z}_q \end{array}\right] \right| \geq \varepsilon$$

then there is an inverter $\mathcal{A}$ that runs in time $t' = t \cdot \mathsf{poly}(n, |H|, 1/\varepsilon)$ such that

$$\Pr\left[\mathcal{A}(y) = \mathbf{s} \;\middle|\; \begin{array}{l} \mathbf{s} \xleftarrow{\$} H^n \\ y \leftarrow f(\mathbf{s}) \end{array}\right] \geq \Omega\left(\frac{\varepsilon^3}{n \cdot q^2}\right)$$

3.2 Lattices and Gaussians

Lattices For $\mathbf{B} = \left[\mathbf{b}_1 \dots \mathbf{b}_n\right] \in \mathbb{R}^{m \times n}$ where $\mathbf{b}_1 \dots \mathbf{b}_n$ are a set of linearly independent vectors, the lattice $\Lambda = \mathcal{L}(\mathbf{B})$ with dimension m and rank n is defined to be the discrete subgroup of $\mathbb{R}^m$ generated by $\mathbf{B}$'s columns i.e.

$$\mathcal{L}(\mathbf{B}) := \mathbf{B}\mathbb{Z}^n = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\},$$

and we say $\mathbf{b}_1 \dots \mathbf{b}_n$ form a *basis* of Λ. Λ is said to be *full-rank* if $n = m$. The span of a lattice $\mathcal{L}(\mathbf{B})$ is the linear space spanned by its vectors i.e. $\mathsf{span}(\mathcal{L}(\mathbf{B})) := \mathbf{B}\mathbb{R}^n = \{\mathbf{B}\mathbf{y} : \mathbf{y} \in \mathbb{R}^n\}$.

The fundamental parallelepiped of $\Lambda = \mathcal{L}(\mathbf{B})$ is defined as $\mathcal{P}(\mathbf{B}) := \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_1, \dots, x_n \in [0,1)\}$. And the determinant of a lattice is defined as the n-dimensional volume of its fundamental parallelepiped, denoted by $\det(\Lambda) := \sqrt{\det(\mathbf{B}^\mathsf{T}\mathbf{B})}$ or more simply $\det(\Lambda) := |\det(\mathbf{B})|$ when Λ is full-rank. This is well-defined because if $\mathbf{B}$ and $\mathbf{B}'$ are two bases for the same lattice Λ if and only if there exists a unimodular matrix $\mathbf{U}$ s.t. $\mathbf{B} = \mathbf{B}'\mathbf{U}$ and hence $\det(\mathbf{B}^\mathsf{T}\mathbf{B}) = \det((\mathbf{B}')^\mathsf{T}\mathbf{B}')$.

We denote by $\lambda_1(\Lambda)$ the first minimum of Λ, defined as the length of a shortest non-zero vector of Λ. More generally, we can define the successive minima of a lattice as follows.

Definition 3.1 (Successive Minima)**.** *Given a lattice Λ of rank n and an integer $1 \le k \le n$, the k-th successive minimum $\lambda_k(\Lambda)$ is defined to be the smallest r such that a closed ball of radius r centered at the origin contains k linearly independent lattice vectors.*

Remark 3.1. Here, the ball radius is measured in ℓ_2 norm by default, but we can also consider other norms (e.g. $\lambda_1^\infty(\Lambda)$ denotes the minimum ℓ_∞ norm of vectors in Λ).

The dual lattice Λ^* of a lattice Λ is defined as the set of all vectors in $\mathsf{span}(\Lambda)$ that have integer inner products with all points in the lattice Λ, i.e,

$$\Lambda^* := \{\mathbf{w} \in \mathsf{span}(\Lambda) : \forall \mathbf{y} \in \Lambda, \langle \mathbf{w}, \mathbf{y} \rangle \in \mathbb{Z}\}.$$

An important property of the dual lattice is the following transference theorem, which gives a relation between the successive minima of a lattice and its dual.

Lemma 3.1 ([Ban93], Theorem 2.1)**.** *For any rank-n lattice Λ, $1 \le \lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \le n$.*

We also define the following notations for some q-ary integer lattices, which are frequently used in lattice-based cryptography. For $q \in \mathbb{N}^*$, a q-ary lattice generated by the column vectors of a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is defined as

$$\mathcal{L}_q(\mathbf{A}) := \left\{\mathbf{y} \in \mathbb{Z}^m : \exists \mathbf{x} \in \mathbb{Z}^n \text{ s.t. } \mathbf{y} \equiv \mathbf{A}\mathbf{x} \bmod q\right\},$$

and the q-ary perpendicular lattice of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is

$$\mathcal{L}_q^\perp(\mathbf{A}) := \left\{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} \equiv 0 \bmod q\right\}.$$

For any $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, both $\mathcal{L}_q(\mathbf{A}^\mathsf{T})$ and $\mathcal{L}_q^\perp(\mathbf{A})$ are full-rank m-dimensional lattices because they contain the full-rank sublattice $q\mathbb{Z}^m$. Furthermore, let Λ^* be the dual lattice of $\mathcal{L}_q(\mathbf{A}^\mathsf{T})$, then $q\Lambda^* = \mathcal{L}_q^\perp(\mathbf{A})$.

Since $q\mathbb{Z}^m \subseteq \mathcal{L}_q(\mathbf{A}^\mathsf{T})$, we always have $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) \le q$. The following lemma shows that equality holds with overwhelming probability over a uniformly random $\mathbf{A}$ when m is large enough.

Lemma 3.2. *Let q be a prime modulus and n, m be positive integers such that $m \geq 2n \log q$ and $q \geq \sqrt{m}$. Then*

$$\Pr_{\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}} \left[\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) < q\right] \leq \left(\frac{77}{m}\right)^{m/2}.$$

Proof. Let $Y := \{\mathbf{y} \in \mathbb{Z}^m : 0 < \|\mathbf{y}\| < q\}$. Every nonzero vector of $q\mathbb{Z}^m$ has norm at least q. So if $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) < q$, there exists $\mathbf{y} \in Y \cap \mathcal{L}_q(\mathbf{A}^\mathsf{T})$ with $\mathbf{y} \notin q\mathbb{Z}^m$, i.e., $\mathbf{y} \equiv \mathbf{A}^\mathsf{T}\mathbf{s} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$. For each fixed $\mathbf{s} \neq \mathbf{0}$, the vector $\mathbf{A}^\mathsf{T}\mathbf{s} \bmod q$ is uniform over $\mathbb{Z}_q^m$: its entries are the inner products of $\mathbf{s}$ with the independent uniform columns of $\mathbf{A}$, and each of them is uniform over $\mathbb{Z}_q$ because $\mathbf{s}$ has an entry that is invertible modulo the prime q. By a union bound over $\mathbf{s}$ and $\mathbf{y}$,

$$\Pr_{\mathbf{A}}\left[\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) < q\right] \leq \sum_{\mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}} \sum_{\mathbf{y} \in Y} \Pr_{\mathbf{A}}\left[\mathbf{A}^\mathsf{T}\mathbf{s} \equiv \mathbf{y} \bmod q\right] \leq q^{n-m} \cdot |Y|.$$

To bound $|Y|$, note that the unit cubes $\mathbf{y} + [-1/2, 1/2)^m$ for $\mathbf{y} \in Y$ are disjoint, and they all lie in the ball of radius $q + \sqrt{m}/2 \leq 3q/2$ centered at the origin. The ball of radius R in $\mathbb{R}^m$ has volume $\pi^{m/2}R^m/\Gamma(m/2+1)$, and $\Gamma(x+1) = \int_0^\infty t^x e^{-t}\,dt \geq x^x \int_x^\infty e^{-t}\,dt = (x/e)^x$ for any $x > 0$. Comparing volumes, we get

$$|Y| \leq \frac{\pi^{m/2}}{\Gamma(m/2+1)}\left(\frac{3q}{2}\right)^m \leq \left(\frac{2\pi e}{m}\right)^{m/2}\left(\frac{3q}{2}\right)^m = \left(\frac{9\pi e \cdot q^2}{2m}\right)^{m/2}.$$

Finally, $m \geq 2n \log q$ gives $q^n \leq 2^{m/2}$. Combining the two bounds,

$$\Pr_{\mathbf{A}}\left[\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) < q\right] \leq 2^{m/2}\left(\frac{9\pi e}{2m}\right)^{m/2} = \left(\frac{9\pi e}{m}\right)^{m/2} \leq \left(\frac{77}{m}\right)^{m/2}. \qquad \square$$

Gaussian function A symmetric matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is said to be positive definite if for any $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, $\mathbf{x}^\mathsf{T}\mathbf{\Sigma}\mathbf{x} > 0$, and positive semidefinite if for any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\mathsf{T}\mathbf{\Sigma}\mathbf{x} \geq 0$. For any positive definite matrix $\mathbf{\Sigma}$, there exists a unique positive definite matrix $\sqrt{\mathbf{\Sigma}}$ such that $\sqrt{\mathbf{\Sigma}}^\mathsf{T}\sqrt{\mathbf{\Sigma}} = \mathbf{\Sigma}$. We write $\mathbf{\Sigma}_1 \geq \mathbf{\Sigma}_2$ if $\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2$ is positive semidefinite.

For a positive definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, define the Gaussian function $\rho_{\sqrt{\mathbf{\Sigma}}} : \mathbb{R}^n \to (0, 1]$ by

$$\rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x}) := e^{-\pi \mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x}},$$

and specially for scalar $\sigma > 0$,

$$\rho_\sigma(\mathbf{x}) := \rho_{\sigma\mathbf{I}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}\|^2/\sigma^2}.$$

Sometimes we consider the more general case where the Gaussian has a center $\mathbf{c} \in \mathbb{R}^n$ and define $\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) := \rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x} - \mathbf{c})$.

By straightforward calculation, one can show the following statement regarding the ratio of the Gaussian function on two vectors.

Lemma 3.3. *Let $\sigma > 0$ be some Gaussian parameter. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, if $\|\mathbf{x}\| \leq a$ and $\|\mathbf{x} - \mathbf{y}\| \leq b$, then*

$$\frac{\rho_\sigma(\mathbf{y})}{\rho_\sigma(\mathbf{x})} \geq 1 - \frac{\pi}{\sigma^2}(2ab + b^2).$$

Proof.

$$\begin{aligned}\frac{\rho_\sigma(\mathbf{y})}{\rho_\sigma(\mathbf{x})} &= \exp\left(-\frac{\pi}{\sigma^2}\left(\|\mathbf{x}+(\mathbf{y}-\mathbf{x})\|^2 - \|\mathbf{x}\|^2\right)\right)\\ &\geq \exp\left(-\frac{\pi}{\sigma^2}\left(2\|\mathbf{x}\|\cdot\|\mathbf{y}-\mathbf{x}\| + \|\mathbf{y}-\mathbf{x}\|^2\right)\right)\\ &\geq \exp\left(-\frac{\pi}{\sigma^2}(2ab+b^2)\right)\\ &\geq 1-\frac{\pi}{\sigma^2}(2ab+b^2)\end{aligned}$$

□

Discrete Gaussian For a discrete set S, we extend ρ to sets by $\rho_{\sqrt{\mathbf{\Sigma}}}(S) = \sum_{\mathbf{x}\in S}\rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x})$. For an n-dimensional lattice Λ, the discrete Gaussian distribution on Λ with parameter $\sqrt{\mathbf{\Sigma}}$, for a positive definite matrix $\mathbf{\Sigma}\in\mathbb{R}^{n\times n}$, is the distribution over Λ such that the probability mass on a vector $\mathbf{x}\in\Lambda$ is proportional to $\rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x})$:

$$\mathcal{D}_{\Lambda,\sqrt{\mathbf{\Sigma}}}(\mathbf{x}) := \frac{\rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x})}{\rho_{\sqrt{\mathbf{\Sigma}}}(\Lambda)}.$$

A common discrete Gaussian in this work is the discrete Gaussian over the integer lattice $\mathbb{Z}^n$ with parameter σ, $\mathcal{D}_{\mathbb{Z}^n,\sigma}(\mathbf{x}) := \rho_\sigma(\mathbf{x})/\rho_\sigma(\mathbb{Z}^n)$, which is equivalent to $\mathcal{D}^n_{\mathbb{Z},\sigma}$ due to the independence among different coordinates.

Note that the discrete Gaussian $\mathcal{D}_{\mathbb{Z},\sigma}$ is a sub-Gaussian distribution with parameter σ (see [MP12, Lemma 2.8]). This gives the following tail bound on linear combinations of discrete Gaussians.

Lemma 3.4. *Let $X_1,\ldots,X_k$ be independent random variables with $X_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_i}$ for some $\sigma_i > 0$. Then for any reals $a_1,\ldots,a_k$ and $t>0$,*

$$\mathsf{Pr}\left[\left|\sum_i a_i X_i\right| > t\sqrt{\sum_i a_i^2\sigma_i^2}\right] \leq 2e^{-\pi t^2}.$$

Proof. By [MP12, Lemma 2.8], $\mathbb{E}[e^{2\pi u X_i}] \leq e^{\pi\sigma_i^2 u^2}$ for every $u\in\mathbb{R}$. Let $Y := \sum_i a_i X_i$ and $s := \sqrt{\sum_i a_i^2\sigma_i^2}$, and assume $s>0$, as otherwise $Y=0$. By independence, $\mathbb{E}[e^{2\pi u Y}] = \prod_i \mathbb{E}[e^{2\pi u a_i X_i}] \leq e^{\pi s^2 u^2}$, so Markov's inequality with $u = t/s$ gives

$$\mathsf{Pr}[Y > ts] \leq e^{\pi s^2u^2 - 2\pi uts} = e^{-\pi t^2}.$$

The same bound holds for $-Y$ by taking $u=-t/s$, and a union bound finishes the proof. □

By applying a standard concentration bound on the norm of random vectors with independent sub-Gaussian coordinates [Ver26, Theorem 3.1.1], we have the following tail bound on the norm of a discrete Gaussian random vector.

Lemma 3.5. *Let $\sigma>0$ be some Gaussian parameter. For any $\delta(\lambda)\in\omega(\sqrt{\log\lambda})$ function, it holds that*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{D}^n_{\mathbb{Z},\sigma}}\left[\|\mathbf{x}\| \geq \frac{\sigma}{\sqrt{2\pi}}\left(\sqrt{n}+\delta(\lambda)\right)\right] \leq \mathsf{negl}(\lambda).$$

3.3 Lattice Smoothing

We use the following notion of lattice smoothing parameter proposed in the seminal work by Micciancio and Regev ([MR04]), which on intuition measures the Gaussian noise needed to make a "uniform distribution on a lattice" statistically close to a "uniform distribution on the entire span".

Definition 3.2 (Smoothing parameter)**.** *For any $\varepsilon > 0$, define the smoothing parameter of a lattice Λ with parameter ε, denoted by $\eta_\varepsilon(\Lambda)$, as the smallest σ such that $\rho_{1/\sigma}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \varepsilon$.*

There are a variety of bounds on the smoothing parameter, most of which relate it to either the minimum distance of the dual lattice $\lambda_1(\Lambda^*)$ or the last successive minima of the primal lattice $\lambda_n(\Lambda)$. In this paper, however, we choose a bound that relates the smoothing parameter of a lattice to the minimum distance of its dual lattice in the ℓ_∞-norm, which suits our analysis best and gives the tightest parameters.

Lemma 3.6 ([Pei07])**.** *Let Λ be an arbitrary rank-n lattice, we have*

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

Then for any function $h(n) \in \omega\left(\sqrt{\log n}\right)$, there exists a negligible $\varepsilon = \varepsilon(n)$ for which $\eta_\varepsilon(\Lambda) \leq h(n)/\lambda_1^\infty(\Lambda^*)$.

A property of the smoothing parameter that will be used in this paper is that when the Gaussian parameter exceeds the smoothing parameter of a lattice, the total Gaussian measure with different centers is essentially the same.

Lemma 3.7 ([Pei10, Lemma 2.4])**.** *Let $\Lambda \subset \mathbb{R}^n$ be a full-rank lattice, let $\varepsilon \in (0,1)$, and let $\mathbf{\Sigma}$ be positive definite with $\mathbf{\Sigma} \geq \eta_\varepsilon(\Lambda)^2 \mathbf{I}_n$. Then for any $\mathbf{c} \in \mathbb{R}^n$, we have*

$$\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\Lambda) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot \rho_{\sqrt{\mathbf{\Sigma}}}(\Lambda).$$

Linear combinations of discrete Gaussians The following lemma is the discrete analogue of the fact that a Gaussian vector splits into a linear image of a narrower Gaussian vector plus an independent Gaussian vector. It is a consequence of discrete Gaussian convolution [Pei10, GMPW20], and we include a proof for the parameters we need.

Lemma 3.8. *Let $0 < \varepsilon < 1$, let $\mathbf{F} \in \mathbb{Z}^{n \times m}$, and let $\mathbf{\Sigma}_1$ be positive definite. Suppose that*

$$\mathbf{\Sigma}_1 \geq 2\eta_\varepsilon(\mathbb{Z}^n)^2 \mathbf{I}_n, \qquad \sigma^2 \geq 2\|\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}\|$$

for some $\sigma > 0$, and set $\mathbf{\Sigma}_2 := \sigma^2\mathbf{I}_m - \mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}$. For independent $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_1}}$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}_2}}$, we have

$$\Delta\left(\mathbf{F}^\mathsf{T}\mathbf{x} + \mathbf{y}, \mathcal{D}^m_{\mathbb{Z},\sigma}\right) \leq \frac{\varepsilon}{1-\varepsilon}.$$

Proof. The second inequality implies $\mathbf{\Sigma}_2 \geq (\sigma^2/2)\mathbf{I}_m$, so $\mathbf{\Sigma}_2$ is positive definite. Fix $\mathbf{z} \in \mathbb{Z}^m$. To write $\rho_{\sqrt{\mathbf{\Sigma}_1}}(\mathbf{x})\rho_{\sqrt{\mathbf{\Sigma}_2}}(\mathbf{z} - \mathbf{F}^\mathsf{T}\mathbf{x})$ as a Gaussian function of $\mathbf{x}$, we complete the square in its exponent, which is $-\pi$ times

$$\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_1^{-1}\mathbf{x} + (\mathbf{z} - \mathbf{F}^\mathsf{T}\mathbf{x})^\mathsf{T}\mathbf{\Sigma}_2^{-1}(\mathbf{z} - \mathbf{F}^\mathsf{T}\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{\Sigma}_3^{-1}\mathbf{x} - 2\mathbf{x}^\mathsf{T}\mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{z} + \mathbf{z}^\mathsf{T}\mathbf{\Sigma}_2^{-1}\mathbf{z},$$

where $\mathbf{\Sigma}_3 := (\mathbf{\Sigma}_1^{-1} + \mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{F}^\mathsf{T})^{-1}$ is the inverse of the matrix of the quadratic term; it is well defined and positive definite because $\mathbf{\Sigma}_1^{-1}$ is positive definite and $\mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{F}^\mathsf{T}$ is positive semidefinite. We set $\mathbf{c} := \mathbf{\Sigma}_3\mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{z}$, so that the linear term equals $-2\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_3^{-1}\mathbf{c}$ and the right-hand side becomes

$$(\mathbf{x}-\mathbf{c})^\mathsf{T}\mathbf{\Sigma}_3^{-1}(\mathbf{x}-\mathbf{c}) + \mathbf{z}^\mathsf{T}\mathbf{\Sigma}_2^{-1}\mathbf{z} - \mathbf{c}^\mathsf{T}\mathbf{\Sigma}_3^{-1}\mathbf{c}.$$

To compute the terms without $\mathbf{x}$, and later to bound $\mathbf{\Sigma}_3$, we use the identity[8]

$$\mathbf{\Sigma}_3 = \mathbf{\Sigma}_1 - \sigma^{-2}\mathbf{\Sigma}_1\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1. \tag{5}$$

Multiplying Eq. (5) on the right by $\mathbf{F}$, we get $\mathbf{\Sigma}_3\mathbf{F} = \mathbf{\Sigma}_1\mathbf{F}(\mathbf{I}_m - \sigma^{-2}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}) = \sigma^{-2}\mathbf{\Sigma}_1\mathbf{F}\mathbf{\Sigma}_2$, so $\mathbf{c} = \sigma^{-2}\mathbf{\Sigma}_1\mathbf{F}\mathbf{z}$. Together with $\mathbf{\Sigma}_3^{-1}\mathbf{c} = \mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{z}$, this gives

$$\mathbf{z}^\mathsf{T}\mathbf{\Sigma}_2^{-1}\mathbf{z} - \mathbf{c}^\mathsf{T}\mathbf{\Sigma}_3^{-1}\mathbf{c} = \mathbf{z}^\mathsf{T}\mathbf{\Sigma}_2^{-1}\mathbf{z} - \sigma^{-2}\mathbf{z}^\mathsf{T}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{z} = \sigma^{-2}\mathbf{z}^\mathsf{T}(\sigma^2\mathbf{I}_m - \mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F})\mathbf{\Sigma}_2^{-1}\mathbf{z} = \frac{\|\mathbf{z}\|^2}{\sigma^2}.$$

Therefore

$$\rho_{\sqrt{\mathbf{\Sigma}_1}}(\mathbf{x})\rho_{\sqrt{\mathbf{\Sigma}_2}}(\mathbf{z}-\mathbf{F}^\mathsf{T}\mathbf{x}) = \rho_\sigma(\mathbf{z})\rho_{\sqrt{\mathbf{\Sigma}_3}}(\mathbf{x}-\mathbf{c}).$$

Summing over $\mathbf{x} \in \mathbb{Z}^n$, we obtain

$$\Pr\left[\mathbf{F}^\mathsf{T}\mathbf{x} + \mathbf{y} = \mathbf{z}\right] \propto \rho_\sigma(\mathbf{z})\rho_{\sqrt{\mathbf{\Sigma}_3},\mathbf{c}}(\mathbb{Z}^n).$$

It remains to show that the last factor barely depends on $\mathbf{z}$. By Lemma 3.7, it suffices to show $\mathbf{\Sigma}_3 \geq \eta_\varepsilon(\mathbb{Z}^n)^2\mathbf{I}_n$, which follows from $\mathbf{\Sigma}_3 \geq \mathbf{\Sigma}_1/2$ and the assumption on $\mathbf{\Sigma}_1$. Indeed, for every $\mathbf{v} \in \mathbb{R}^n$, since $\sqrt{\mathbf{\Sigma}_1}$ is symmetric,

$$\mathbf{v}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{v} = \|\mathbf{F}^\mathsf{T}\sqrt{\mathbf{\Sigma}_1}\sqrt{\mathbf{\Sigma}_1}\mathbf{v}\|^2 \leq \|\mathbf{F}^\mathsf{T}\sqrt{\mathbf{\Sigma}_1}\|^2\|\sqrt{\mathbf{\Sigma}_1}\mathbf{v}\|^2 = \|\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F}\| \cdot \mathbf{v}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{v} \leq \frac{\sigma^2}{2}\,\mathbf{v}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{v}.$$

By Eq. (5), this gives $\mathbf{v}^\mathsf{T}\mathbf{\Sigma}_3\mathbf{v} \geq \mathbf{v}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{v}/2$ for every $\mathbf{v} \in \mathbb{R}^n$, that is, $\mathbf{\Sigma}_3 \geq \mathbf{\Sigma}_1/2$. Therefore Lemma 3.7 with $\Lambda = \mathbb{Z}^n$ gives

$$\rho_{\sqrt{\mathbf{\Sigma}_3},\mathbf{c}}(\mathbb{Z}^n) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot \rho_{\sqrt{\mathbf{\Sigma}_3}}(\mathbb{Z}^n) \qquad \text{for every } \mathbf{z} \in \mathbb{Z}^m.$$

Consequently, after normalization,

$$\Pr\left[\mathbf{F}^\mathsf{T}\mathbf{x} + \mathbf{y} = \mathbf{z}\right] \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \mathcal{D}^m_{\mathbb{Z},\sigma}(\mathbf{z}).$$

Every ratio in this interval differs from 1 by at most $2\varepsilon/(1-\varepsilon)$, so the statistical distance is at most $\varepsilon/(1-\varepsilon)$. □

The discrete Gaussians in our reductions can be sampled exactly: a sample from $\mathcal{D}_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}}}$ is $\sqrt{\mathbf{\Sigma}}$ times a sample from the discrete Gaussian with parameter 1 over the lattice with basis $\sqrt{\mathbf{\Sigma}}^{-1}$, and the algorithm of [BLP+13, Lemma 2.3] samples the latter in polynomial time whenever the least eigenvalue of $\mathbf{\Sigma}$ is at least $\ln(2m+4)/\pi$, as is the case in our reductions.

[8] To verify Eq. (5), note that $\mathbf{I}_m - \sigma^{-2}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F} = \sigma^{-2}\mathbf{\Sigma}_2$ by the definition of $\mathbf{\Sigma}_2$, so

$$\begin{aligned}(\mathbf{\Sigma}_1^{-1} + \mathbf{F}\mathbf{\Sigma}_2^{-1}\mathbf{F}^\mathsf{T})(\mathbf{\Sigma}_1 - \sigma^{-2}\mathbf{\Sigma}_1\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1) &= \mathbf{I}_n - \sigma^{-2}\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1 + \mathbf{F}\mathbf{\Sigma}_2^{-1}(\mathbf{I}_m - \sigma^{-2}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F})\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1 \\ &= \mathbf{I}_n - \sigma^{-2}\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1 + \sigma^{-2}\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1 = \mathbf{I}_n.\end{aligned}$$

3.4 Learning with Errors

The Learning with Errors (LWE) problem was first defined in [Reg05]. It asks one to recover a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ given a sequence of "approximate" random linear equations on $\mathbf{s}$. Formally, we define the search and decisional LWE problems as follows.

Definition 3.3 (Learning with Errors)**.** *Let λ be a security parameter. Let $n = n(\lambda)$, $m = m(\lambda)$ and $q = q(\lambda)$ be positive integers and χ be an error distribution on $\mathbb{Z}$. We say that the (search) LWE problem $\mathsf{LWE}_{n,m,q,\chi}$ is hard, if for every P.P.T. adversary $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q) = \mathbf{s}\right] \leq \mathsf{negl}(\lambda),$$

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

Likewise, we say that the decision LWE problem $\mathsf{dLWE}_{n,m,q,\chi}$ is hard, if

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}),$$

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

We also consider the hardness of solving the LWE problems for any $m = \mathsf{poly}(n \log q)$ and denote it by $\mathsf{LWE}_{n,q,\chi}$ and $\mathsf{dLWE}_{n,q,\chi}$ [9].

Abundant work has been done on basing the hardness of LWE problem on long-studied worst-case lattice problems. For the discrete Gaussian error distribution $\chi = \mathcal{D}_{\mathbb{Z},\sigma}$ with $\sigma = \alpha q \geq 2\sqrt{n} + \omega(\sqrt{\log n})$, there is a quantum reduction from the approximation variant of the Shortest Independent Vectors problem (SIVP) for approximation factor $\gamma = \tilde{O}(n/\alpha)$ in worst-case n-dimensional lattices to the average-case problem $\mathsf{LWE}_{n,q,\chi}$ [Reg05, Pei10]. Classical reductions from worst-case lattice problems also exist ([Pei09]) but require the modulus q to be exponentially large or reduce from a slightly non-standard worst-case lattice problem. A search-to-decision reduction for the LWE problem was provided in [Reg05] for a polynomial-sized prime modulus q, and generalized reductions for a wider class of modulus were proposed in [Pei09] and [BLP+13]. In this work, all the LWE assumptions and all the distributions in our proofs use discrete Gaussian errors.

Remark 3.2. We use Lemma 3.8 to increase the error width of an LWE instance. If $s = \omega(\sqrt{\log \lambda})$, $t \geq \sqrt{2}s$, and m is polynomially bounded, then independent $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},s}^m$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z},\sqrt{t^2-s^2}}^m$ satisfy $\mathbf{x} + \mathbf{y} \approx_{st} \mathcal{D}_{\mathbb{Z},t}^m$. This follows from Lemma 3.8 with $\mathbf{F} = \mathbf{I}_m$, since Lemma 3.6 gives $\eta_\varepsilon(\mathbb{Z}^m) \leq s/\sqrt{2}$ for some negligible ε. Since increasing the secret dimension also preserves hardness, the hardness of $\mathsf{LWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ for $\gamma = \omega(\sqrt{\log \lambda})$ implies that of $\mathsf{LWE}_{n,q,\mathcal{D}_{\mathbb{Z},\sigma}}$ for every $n \geq n'$ and $\sigma \geq \sqrt{2}\gamma$, and likewise for dLWE.

3.5 Trapdoor Sampling

Definition 3.4 (Lattice trapdoors)**.** *For $q \in \mathbb{N}^*$ and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we say $\mathbf{T} \in \mathbb{Z}^{m\times m}$ is a trapdoor for $\mathbf{A}$ if $\mathbf{T}$ is full rank over $\mathbb{R}$ and $\mathbf{AT} \bmod q = \mathbf{0}$, i.e., a set of m linearly-independent vectors in $\mathcal{L}_q^\perp(\mathbf{A})$.*

[9] It is reasonable to do so since it turns out that the number of equations m actually does not really matter, which can be explained by the fact that there exists an algorithm that enables one to generate an arbitrary number of LWE samples that are "as good as new" given a small fixed polynomial number of LWE samples ([GPV08, security proof for the IBE system] and [ACPS09, Construction 1]).

We say a random variable $\mathbf{T} \in \mathbb{Z}^{m \times m}$ *is an* r*-Gaussian random trapdoor for* $\mathbf{A}$ *if* $\mathbf{T}$ *is distributed as a random variable sampled by the following procedure:*

1. $i \leftarrow 1$.
2. *While* $i \leq m$*:*
   (a) *Independently sample* $\mathbf{t}_i \leftarrow \mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{A}),r}$.
   (b) *If* $\mathbf{t}_i \notin \mathsf{span}(\mathbf{t}_1, \ldots, \mathbf{t}_{i-1})$, $i \leftarrow i + 1$.
3. *Output* $\mathbf{T} := [\mathbf{t}_1 \ \ldots \ \mathbf{t}_m]$.

Such a trapdoor $\mathbf{T}$, if satisfying $\|\mathbf{T}\|$ is small enough, enables efficient LWE inversion for $\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T}$ with short $\mathbf{e}$ and preimage sampling for function $f_\mathbf{A} : \mathbf{x} \mapsto \mathbf{A}\mathbf{x}$.

In [MP12], a new trapdoor notion was introduced (for distinction we call it an "inhomogeneous trapdoor" in this work), which is a small-norm $\mathbf{R} \in \mathbb{Z}^{(m-w)\times w}$ satisfying $\mathbf{A}\begin{bmatrix} -\mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{G}_\tau$ for a certain special gadget matrix $\mathbf{G}_\tau$. This trapdoor has even stronger functionalities than a homogeneous one. Especially, one can produce a high-quality homogeneous trapdoor $\mathbf{T}$ with $\mathbf{R}$. We will use the trapdoor sampling scheme described in Fig. 1 in our proofs. It samples $\mathbf{A}$ that is statistically close to random along with an inhomogeneous trapdoor $\mathbf{R}$.

Algorithm TdSamp

TdSamp:

1. Input: 1^n, 1^m, q, and τ where $m > n\lceil \log_\tau q \rceil$.
2. Let $c = \lceil \log_\tau q \rceil$, $w = n \cdot c$ and $m' = m - w$.
3. Let $\mathbf{g}_\tau = [1 \ \tau \ \ldots \ \tau^{c-1}]^\mathsf{T}$ and set gadget matrix $\mathbf{G}_\tau = \mathbf{I}_n \otimes \mathbf{g}_\tau^\mathsf{T} \in \mathbb{Z}_q^{n \times w}$ ($\otimes$ denotes the tensor product).
4. Sample $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n \times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m' \times w}$. Set $\mathbf{A} = [\mathbf{B} \| \mathbf{B}\mathbf{R} + \mathbf{G}_\tau] \in \mathbb{Z}_q^{n \times m}$.
5. Output $(\mathbf{A}, \mathbf{R})$.

Figure 1: TdSamp

Theorem 3.2 (Trapdoor Sampling, [MP12])**.** *Define algorithm* TdSamp *as in Figure 1. Let* $\tau \geq 2$ *be a positive integer and* q *be a prime modulus. Let* $n = \mathsf{poly}(\lambda)$, $m \geq 2n \log q$, *and* $q \geq \sqrt{m}$. *Let* $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^m, q, \tau)$, *then*

1. $\mathbf{A}$ *is statistically indistinguishable from uniform random and thus has full row rank with overwhelming probability.*
2. *There exists an efficient algorithm* TdInvert *such that with overwhelming probability over* $(\mathbf{A}, \mathbf{R})$, *it holds that*

$$\mathsf{TdInvert}(\mathbf{A}, \mathbf{R}, t, \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q) = \begin{cases} (\mathbf{s}, \mathbf{e}) & \text{if } \|\mathbf{e}\| \leq t \\ \perp & \text{if } \|\mathbf{e}\| > t \end{cases}$$

for any $\mathbf{s} \in \mathbb{Z}_q^n$, $t \leq q/(4m\tau)$, *and* $\mathbf{e} \in \mathbb{Z}_q^m$ *satisfying* $\|\mathbf{e}\| < q - t$.

Remark 3.3. The inversion algorithm of [MP12] only states a one-sided guarantee: on input $\mathbf{b} = \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q$ with $\|\mathbf{e}\| \le q/(4m\tau)$ it outputs $(\mathbf{s}, \mathbf{e})$, with no guarantee for larger $\mathbf{e}$. The algorithm $\mathsf{TdInvert}$ in Theorem 3.2 simply runs it on $\mathbf{b}$, and outputs its result $(\mathbf{s}', \mathbf{e}')$ if $\mathbf{A}^\mathsf{T}\mathbf{s}' + \mathbf{e}' \equiv \mathbf{b} \bmod q$ and $\|\mathbf{e}'\| \le t$, and outputs $\perp$ otherwise. The extra rejection guarantee comes from Lemma 3.2. With overwhelming probability, $\mathbf{A}$ has full row rank, and $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) = q$ by Lemma 3.2, since $\mathbf{A}$ is statistically close to uniform and $(77/m)^{m/2} \le 2^{-m/2}$ for $m \ge 154$. Condition on such an $\mathbf{A}$. If $\|\mathbf{e}\| \le t$, then $\|\mathbf{e}\| \le q/(4m\tau)$, so the inversion algorithm outputs $(\mathbf{s}, \mathbf{e})$, which passes both checks. Conversely, suppose $\mathsf{TdInvert}$ outputs some $(\mathbf{s}', \mathbf{e}')$. Then $\mathbf{e} - \mathbf{e}' \equiv \mathbf{A}^\mathsf{T}(\mathbf{s}' - \mathbf{s}) \bmod q$, so $\mathbf{e} - \mathbf{e}' \in \mathcal{L}_q(\mathbf{A}^\mathsf{T})$ and $\|\mathbf{e} - \mathbf{e}'\| \le \|\mathbf{e}\| + \|\mathbf{e}'\| < (q - t) + t = q$. Hence $\mathbf{e}' = \mathbf{e}$ and then $\mathbf{s}' = \mathbf{s}$. In particular $\|\mathbf{e}\| = \|\mathbf{e}'\| \le t$, so $\mathsf{TdInvert}$ outputs $\perp$ when $\|\mathbf{e}\| > t$.

The base-τ gadget matrix $\mathbf{G}_\tau$ will play an important role in this work. Here we prove a bound on the smoothing parameter of the lattice $\mathcal{L}_q(\mathbf{G}_\tau^\mathsf{T})$ i.e. the q-ary lattice generated by $\mathbf{G}_\tau$'s rows.

Lemma 3.9. *Let $\tau \ge 2$ be a positive integer and $q > \tau$ be a prime modulus. Let $c = \lceil \log_\tau q \rceil$ and dimensions $n > 0$, $w := n \cdot c$. For any function $h(w) = \omega\left(\sqrt{\log w}\right)$, there exists a negligible $\varepsilon = \mathsf{negl}(w)$ s.t.*

$$\eta_\varepsilon(\mathcal{L}_q(\mathbf{G}_\tau^\mathsf{T})) \le \tau^{c-1} h(w) \,.$$

Proof. Let $\Lambda^* = \frac{1}{q}\mathcal{L}_q^\perp(\mathbf{G}_\tau)$ be the dual lattice of $\Lambda = \mathcal{L}_q(\mathbf{G}_\tau^\mathsf{T})$. By Lemma 3.6, it is sufficient to prove $\lambda_1^\infty(\Lambda^*) \ge 1/(2\tau^{c-1})$ i.e. $\lambda_1^\infty(\mathcal{L}_q^\perp(\mathbf{G}_\tau)) \ge q/(2\tau^{c-1})$. Note that $\mathbf{G}_\tau = \mathbf{I}_n \otimes \mathbf{g}_\tau^\mathsf{T}$ and

$$\mathbf{g}_\tau = \begin{bmatrix} 1 & \tau & \dots & \tau^{c-1} \end{bmatrix}^\mathsf{T}$$

If some $\mathbf{x} \in \mathbb{Z}^c \backslash \{\mathbf{0}\}$ satisfies $\langle \mathbf{x}, \mathbf{g}_\tau \rangle \equiv 0 \bmod q$, we have either

(i) $|\langle \mathbf{x}, \mathbf{g}_\tau \rangle| \ge q$; or

(ii) $\langle \mathbf{x}, \mathbf{g}_\tau \rangle = 0$.

In case (i), since

$$\left|\langle \mathbf{x}, \mathbf{g}_\tau \rangle\right| \le \|\mathbf{x}\|_\infty \cdot \sum_{i=0}^{c-1} \tau^i = \frac{\tau^c - 1}{\tau - 1} \cdot \|\mathbf{x}\|_\infty,$$

we have

$$\|\mathbf{x}\|_\infty \ge q \cdot \frac{\tau - 1}{\tau^c - 1} \ge \frac{q}{2\tau^{c-1}}.$$

In case (ii), suppose the last non-zero entry of $\mathbf{x} = (x_1, x_2, \dots, x_c)$ is x_{k+1} ($k > 0$), then we have

$$|x_{k+1}\tau^k| = |x_k\tau^{k-1} + \cdots + x_2\tau + x_1| \,.$$

This implies,

$$\tau^k \le |x_{k+1}\tau^k| = |x_k\tau^{k-1} + \cdots + x_2\tau + x_1| \le \|\mathbf{x}\|_\infty \cdot \sum_{i=0}^{k-1} \tau^i = \frac{\tau^k - 1}{\tau - 1} \cdot \|\mathbf{x}\|_\infty \,,$$

which implies

$$\|\mathbf{x}\|_\infty \ge \frac{\tau^k(\tau - 1)}{\tau^k - 1} \ge \tau/2 \ge \frac{q}{2\tau^{c-1}} \,.$$

Therefore, if $\mathbf{y} \in \mathcal{L}_q^\perp(\mathbf{G}_\tau)\backslash\{\mathbf{0}\}$, we have $\|\mathbf{y}\|_\infty \ge q/(2\tau^{c-1})$. So $\lambda_1^\infty(\mathcal{L}_q^\perp(\mathbf{G}_\tau)) \ge q/(2\tau^{c-1})$. □

In the rest of this paper, we will omit the base parameter τ in $\mathbf{G}_\tau$ and just denote it by $\mathbf{G}$ for simplicity of notation. If not explicitly specified, we will always work with the gadget matrix $\mathbf{G}$ with base τ.

For simplicity, we abuse the notation and reuse $\mathsf{TdInvert}$ to also represent the inversion algorithm for a homogeneous lattice trapdoor $\mathbf{T}$, with the same interface as in Theorem 3.2. The following lemma gives the decoding radius of a homogeneous trapdoor. As in Remark 3.3, the inversion algorithm of [GPV08, MP12] only gives the first case of the lemma, and the second case follows by checking its output in the same way, with $q/(4\|\mathbf{T}\|)$ in place of $q/(4m\tau)$.

Lemma 3.10 (LWE Inversion [GPV08, MP12])**.** *Let q be a prime modulus and $n, m = \mathsf{poly}(\lambda)$. There exists an efficient algorithm $\mathsf{TdInvert}$ such that for any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that has full row rank and satisfies $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) = q$, and any homogeneous trapdoor $\mathbf{T}$ of $\mathbf{A}$, it holds that*

$$\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q) = \begin{cases} (\mathbf{s}, \mathbf{e}) & \text{if } \|\mathbf{e}\| \le t \\ \perp & \text{if } \|\mathbf{e}\| > t \end{cases}$$

for any $\mathbf{s} \in \mathbb{Z}_q^n$, $t \le q/(4\|\mathbf{T}\|)$, and $\mathbf{e} \in \mathbb{Z}_q^m$ satisfying $\|\mathbf{e}\| < q - t$.

The next lemma states the fact that with an inhomogeneous trapdoor $\mathbf{R}$, one can sample a *random Gaussian* homogeneous trapdoor $\mathbf{T}$ for $\mathbf{A}$ by utilizing the preimage sampling functionality of $\mathbf{R}$. Note that by [Reg05, Corollary 3.16], we can obtain a full-rank $\mathbf{T}$ with overwhelming probability through at most m^2 independent samples from $\mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{A}),r}$.

Lemma 3.11 (Derived from [MP12, BLP+13, Reg05])**.** *Let $\tau \ge 2$ be a positive integer and $q > \tau$ be a prime modulus. Let $n, m = \mathsf{poly}(\lambda)$ such that $m \ge 2n \log q$. There exists an efficient algorithm RndTd such that for any $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^m, q, \tau)$ and $r \ge m\tau\omega(\sqrt{\log m})$, $\mathsf{RndTd}(q, \tau, \mathbf{A}, \mathbf{R}, r)$ outputs an r-Gaussian trapdoor $\mathbf{T}$ of $\mathbf{A}$ with overwhelming probability.*

We use $\mathsf{HTdSamp}$ to denote the algorithm that samples $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is statistically close to uniform random together with a Gaussian random homogeneous trapdoor $\mathbf{T} \in \mathbb{Z}^{m\times m}$ through the following procedure: on input 1^n, 1^m, q, r, set $\tau := r/(m \log m)$, sample

$$\begin{aligned} (\mathbf{A}, \mathbf{R}) &\leftarrow \mathsf{TdSamp}(1^n, 1^m, q, \tau), \\ \mathbf{T} &\leftarrow \mathsf{RndTd}(q, \tau, \mathbf{A}, \mathbf{R}, r), \end{aligned}$$

and output $(\mathbf{A}, \mathbf{T})$.

Furthermore, combining the random Gaussian trapdoor sampler above and trapdoor extension techniques [CHKP10, MP12], one can sample a random trapdoor for $[\mathbf{A}_1\|\mathbf{A}_2]$ (or $[\mathbf{A}_2\|\mathbf{A}_1]$) using a trapdoor of $\mathbf{A}_1$.

Lemma 3.12. *Let $\tau \ge 2$ be a positive integer and $q > \tau$ be a prime modulus. Let n, $m_1 \ge 2n \log q$, m_2, and $m = m_1 + m_2$ be polynomially bounded positive integers. There exists an efficient algorithm ExtTd such that for any $(\mathbf{A}_1, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_1}, q, \tau)$, $r \ge m_1\tau\omega(\sqrt{\log m_1})$, and $\mathbf{A}_2 \in \mathbb{Z}_q^{n\times m_2}$, $\mathsf{ExtTd}(\text{“}R\text{”}, q, \tau, \mathbf{A}_1, \mathbf{R}, \mathbf{A}_2, r)$ outputs an r-Gaussian trapdoor $\mathbf{T} \in \mathbb{Z}^{m\times m}$ for $[\mathbf{A}_1\|\mathbf{A}_2]$ with overwhelming probability; similarly, $\mathsf{ExtTd}(\text{“}L\text{”}, q, \tau, \mathbf{A}_1, \mathbf{R}, \mathbf{A}_2, r)$ outputs an r-Gaussian trapdoor $\mathbf{T} \in \mathbb{Z}^{m\times m}$ for $[\mathbf{A}_2\|\mathbf{A}_1]$ with overwhelming probability.*

Finally, we have the following upper bound on the spectral norm of a random Gaussian trapdoor sampled by RndTd and ExtTd, by applying results from non-asymptotic theory of random matrices [Ver11].

Lemma 3.13. *For the r-Gaussian trapdoor $\mathbf{T}$ sampled in Lemma 3.11 and Lemma 3.12, we have $\|\mathbf{T}\| \leq O(r \cdot (m + \omega(\sqrt{\log \lambda})))$ with overwhelming probability.*

3.6 Entropic LWE

Now we recall a variant of the LWE problem considered in [BD20] where the secret is drawn from a general entropic distribution rather than being uniformly random on $\mathbb{Z}_q^n$. Intuitively, their result shows that LWE can still remain secure even if some of the information in the secret $\mathbf{s}$ is leaked, as long as the entropy remaining in $\mathbf{s}$ is still large enough.

Definition 3.5 (Entropic LWE)**.** *Let λ be the security parameter, $q = q(\lambda)$ be a modulus, $n, m = \mathsf{poly}(\lambda)$ be positive integers, $\mathcal{S}$ be a secret distribution on $\mathbb{Z}_q^n$, and χ be an error distribution on $\mathbb{Z}$. We say that the decisional entropic LWE problem $\mathsf{ent\text{-}dLWE}_{n,m,q,\chi,\mathcal{S}}$ is hard, if*

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}),$$

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

Theorem 3.3. *Let λ be the security parameter, $q = q(\lambda)$ be a prime modulus, $n, n', m = \mathsf{poly}(\lambda)$ be positive integers with $m \geq n$ and $m = \omega(\log \lambda)$. Let $\mathcal{S}$ be an efficiently samplable distribution on $\mathbb{Z}_q^n$ s.t.*

$$\widetilde{H}_\infty(\mathbf{s}|\mathbf{s} + \mathbf{e}_1) \geq n' \log q + \omega(\log \lambda),$$

where $\mathbf{s} \leftarrow \mathcal{S}$ and $\mathbf{e}_1 \leftarrow \mathcal{D}^n_{\mathbb{Z},\sigma_1}$ for some Gaussian parameter $\sigma_1 = \omega(\sqrt{\log \lambda})$. Let $\sigma > C\sqrt{m}\gamma\sigma_1$ for some global constant C and assume $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some $\gamma = \omega(\sqrt{\log \lambda})$, then $\mathsf{ent\text{-}dLWE}_{n,m,q,\mathcal{D}_{\mathbb{Z},\sigma},\mathcal{S}}$ is hard.

Moreover, the same holds in the presence of auxiliary information $\mathbf{y}$ sampled jointly with $\mathbf{s}$, i.e., $(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathbf{y}) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}, \mathbf{y})$, if $(\mathbf{s}, \mathbf{y})$ is efficiently samplable and $\widetilde{H}_\infty(\mathbf{s}|\mathbf{s} + \mathbf{e}_1, \mathbf{y}) \geq n' \log q + \omega(\log \lambda)$.

[BD20, Theorem 4.1] proves this result with continuous Gaussian errors. We follow their proof, using Lemma 3.8 to split and merge the discrete Gaussian errors. The idea is to make $\mathbf{A}$ lossy: once $\mathbf{A} = \mathbf{BC} + \mathbf{F}$, where $\mathbf{B}$ has only n' columns and $\mathbf{F}$ is short, the sample depends on $\mathbf{s}$ only through $\mathbf{s}^\mathsf{T}\mathbf{B}$, which the leftover hash lemma makes uniform, and through $\mathbf{s}^\mathsf{T}\mathbf{F}$, which is absorbed by splitting off a component $\mathbf{F}^\mathsf{T}\mathbf{e}_1$ of the error.

Proof. Let $\mathbf{s} \leftarrow \mathcal{S}$ and $\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$. By the hardness of $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$, applied to each row, we may replace the uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ by

$$\mathbf{A} = \mathbf{BC} + \mathbf{F} \bmod q,$$

where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times n'}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n'\times m}$, and $\mathbf{F} \leftarrow \mathcal{D}^{n\times m}_{\mathbb{Z},\gamma}$ are independent. Since $m \geq n$ and $m = \omega(\log \lambda)$, the spectral norm bound for matrices with independent sub-Gaussian entries [MP12, Lemmas 2.8 and 2.9] gives $\|\mathbf{F}\| \leq C_0\gamma\sqrt{m}$ with overwhelming probability, for a global constant C_0. We condition on this event, at the cost of negligible statistical distance.

Take $C \geq 2C_0$, so that $\sigma^2 \geq 2\sigma_1^2\|\mathbf{F}\|^2$. Then Lemma 3.8, applied with $\mathbf{F}$ and $\mathbf{\Sigma}_1 = \sigma_1^2\mathbf{I}_n$, allows us to replace $\mathbf{e}$ by $\mathbf{F}^\mathsf{T}\mathbf{e}_1 + \mathbf{e}_2$, where $\mathbf{e}_1 \leftarrow \mathcal{D}^n_{\mathbb{Z},\sigma_1}$ and $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sqrt{\sigma^2\mathbf{I}_m - \sigma_1^2\mathbf{F}^\mathsf{T}\mathbf{F}}}$ are independent of each other and of $\mathbf{s}, \mathbf{B}, \mathbf{C}$. The sample becomes

$$\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} = (\mathbf{s}^\mathsf{T}\mathbf{B})\mathbf{C} + (\mathbf{s} + \mathbf{e}_1)^\mathsf{T}\mathbf{F} + \mathbf{e}_2^\mathsf{T} \bmod q.$$

Since q is prime, the maps $\mathbf{s} \mapsto \mathbf{s}^\mathsf{T}\mathbf{B}$ form a universal family of hash functions, so by the entropy bound and the leftover hash lemma [DORS08, Lemma 2.4], $(\mathbf{B}, \mathbf{s}^\mathsf{T}\mathbf{B}, \mathbf{s} + \mathbf{e}_1) \approx_{st} (\mathbf{B}, \mathbf{u}_0^\mathsf{T}, \mathbf{s} + \mathbf{e}_1)$ for $\mathbf{u}_0 \xleftarrow{\$} \mathbb{Z}_q^{n'}$. We therefore replace $\mathbf{s}^\mathsf{T}\mathbf{B}$ by $\mathbf{u}_0^\mathsf{T}$, and then merge the errors back by Lemma 3.8: the sample becomes $\mathbf{u}_0^\mathsf{T}\mathbf{C} + \mathbf{s}^\mathsf{T}\mathbf{F} + \mathbf{e}^\mathsf{T}$ for a fresh $\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$.

Now $(\mathbf{C}, \mathbf{u}_0^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T})$ is an LWE sample of dimension n' and width $\sigma \geq \sqrt{2}\gamma$, so we can replace $\mathbf{u}_0^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T}$ by a uniform vector, by Remark 3.2. Adding $\mathbf{s}^\mathsf{T}\mathbf{F}$ preserves uniformity. Finally, replacing $\mathbf{BC} + \mathbf{F}$ by a uniform matrix gives $(\mathbf{A}, \mathbf{u}^\mathsf{T})$.

With auxiliary information $\mathbf{y}$, every reduction above samples $(\mathbf{s}, \mathbf{y})$ jointly and outputs $\mathbf{y}$ along with the sample, and the leftover hash lemma is applied with $(\mathbf{s} + \mathbf{e}_1, \mathbf{y})$ in place of $\mathbf{s} + \mathbf{e}_1$. □

4 Hardness Results of the LWE with Trapdoor Hints Problem

In this section, we give our theoretical results related to the LWE with Trapdoor Hints problem, which form the basis for one of our TDRE constructions. We will first present and prove a vanilla LWE with Trapdoor Hints theorem, and then extend it to the setting where some leakage on the error vector is also given to the adversary. Note that our results not only present the traditional "distinguishing hardness" (which also implies the "search hardness"), but furthermore provide a stronger guarantee that the LWE secret vector can still be used for extra safe LWE samples, which will later enable us to construct Euclidean ball obfuscation and TDRE with message payloads. We believe these results are of independent theoretical interest and may find more potential application.

4.1 Vanilla LWE with Trapdoor Hints

The following theorem gives our central result stating that the LWE problem remains hard in the presence of a lattice trapdoor, as long as the Gaussian noise width exceeds the decoding radius of the trapdoor by a small factor.

Theorem 4.1. *Let $\tau > 2$ be a positive integer, q be a prime modulus, and $c := \lceil \log_\tau q \rceil$ i.e., $\tau^{c-1} < q < \tau^c$. Let $n, m' \geq n \log q$, and l be polynomially bounded positive integers. Let $w := n \cdot c = n\lceil \log_\tau q \rceil$ and $m := w + m'$. Let $\gamma = \omega(\sqrt{\log \lambda})$ satisfy $\gamma < \frac{\tau^{c-1}}{\sqrt{m'w(m'+l)}}$, $\sigma \geq \tau^{c-1} \cdot \omega(\sqrt{\log \lambda})$, and $\sigma' \geq \gamma\sqrt{m' + l} \cdot \omega(\sqrt{\log \lambda})$.*

Suppose $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some n' satisfying $n' \log q \leq n - \omega(\log \lambda)$; then

$$\begin{aligned}&(\mathbf{A}, \mathbf{R}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)\\ &\approx_c (\mathbf{A}, \mathbf{R}, \mathbf{u}^\mathsf{T}, \mathbf{C}, (\mathbf{u}')^\mathsf{T}),\end{aligned}$$

where $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^m, q, \tau)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, $\mathbf{e}' \leftarrow \mathcal{D}^l_{\mathbb{Z},\sigma'}$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

From Theorem 4.1, one can see that the noise magnitude needed for the hardness of LWE with Trapdoor Hints problem is essentially $\sigma > q/\tau \cdot \omega(\sqrt{\log \lambda})$, as long as we choose a prime q such that $\Omega(\tau^c) \le q < \tau^c$, which is a mild requirement.

Now we head for the proof of Theorem 4.1.

To begin with, we present two lemmas that serve as stepping stones to our final proof. We first show the following lemma that states the lossiness property of $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}^\mathsf{T}$.

Lemma 4.1. *Let $\tau > 2$ be a positive integer, q be a prime modulus, and $c := \lceil \log_\tau q \rceil$. Let $n = \mathsf{poly}(\lambda)$ and $w := n \cdot c$. For any $\delta(\lambda) \in \omega(\sqrt{\log \lambda})$ function, let $\sigma_1 \ge 40(1+\delta)$ and $\sigma \ge 40\tau^{c-1}(1+\delta)$. Then*

$$\widetilde{H}_\infty(\mathbf{s} | \mathbf{G}^\mathsf{T}\mathbf{s} + \mathbf{e}, \mathbf{s} + \mathbf{e}_1) \ge n - \mathsf{negl}(\lambda),$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^w$, and $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^n$.

Proof. Let $\mathbf{s} = [s_1 \ \dots \ s_n]^\mathsf{T}$. Note that $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\mathsf{T}$, and thus $\mathbf{G}^\mathsf{T}\mathbf{s} = [s_1\mathbf{g}^\mathsf{T} \ \dots \ s_n\mathbf{g}^\mathsf{T}]^\mathsf{T}$. It is easy to observe that the triples formed by each coordinate of $\mathbf{s}$ and the corresponding coordinates of $\mathbf{G}^\mathsf{T}\mathbf{s}+\mathbf{e}$ and $\mathbf{s}+\mathbf{e}_1$ are independent, so $\widetilde{H}_\infty(\mathbf{s}|\mathbf{G}^\mathsf{T}\mathbf{s}+\mathbf{e}, \mathbf{s}+\mathbf{e}_1)$ is the sum of the n corresponding average conditional min-entropies of the coordinates. Therefore, it is sufficient to show the following claim instead:

Claim 4.1. *We have*

$$\widetilde{H}_\infty(s | s\mathbf{g} + \mathbf{e}, s + e_1) \ge 1 - \mathsf{negl}(\lambda),$$

where $s \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^c$, and $e_1 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}$.

To prove the claim above, consider sampling $\mathbf{y} := s'\mathbf{g} + \mathbf{e}'$ and $z := s' + e_1'$ together, where $s' \xleftarrow{\$} \mathbb{Z}_q$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^c$, and $e_1' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}$. Let $a := \|\mathbf{g}\|^2/\sigma^2 + 1/\sigma_1^2$. Since $\langle \mathbf{e}', \mathbf{g}\rangle/\sigma^2 + e_1'/\sigma_1^2$ is a linear combination of the independent discrete Gaussians $\mathbf{e}'$ and e_1', with coefficients $\mathbf{g}/\sigma^2$ and $1/\sigma_1^2$, Lemma 3.4 with $\delta/\sqrt{2\pi}$ in place of t gives

$$\Pr\left[\left|\frac{\langle \mathbf{e}', \mathbf{g}\rangle}{\sigma^2} + \frac{e_1'}{\sigma_1^2}\right| > \frac{\delta\sqrt{a}}{\sqrt{2\pi}}\right] \le 2e^{-\delta^2/2}.$$

Let $\mathcal{T}$ denote the set of pairs $(\mathbf{y}, z)$ for which there exist $s' \in \mathbb{Z}_q$, $\mathbf{e}' \in \mathbb{Z}^c$, and $e_1' \in \mathbb{Z}$ such that $\mathbf{y} = s'\mathbf{g} + \mathbf{e}'$, $z = s' + e_1'$, and $|\langle \mathbf{e}', \mathbf{g}\rangle/\sigma^2 + e_1'/\sigma_1^2| \le \delta\sqrt{a}/\sqrt{2\pi}$. Then $\varepsilon := \Pr[(\mathbf{y}, z) \notin \mathcal{T}] \le 2e^{-\delta^2/2}$, which is negligible since $\delta = \omega(\sqrt{\log \lambda})$.

Consider a pair $(\mathbf{y}, z) \in \mathcal{T}$, and fix such s', $\mathbf{e}'$, and e_1'. Define

$$f_{\mathbf{y},z}(s^*) := \Pr[s = s^* | s\mathbf{g} + \mathbf{e} = \mathbf{y}, s + e_1 = z] \ .$$

Then

$$\begin{aligned} f_{\mathbf{y},z}(s^*) &= \frac{\Pr[s = s^*] \cdot \rho_\sigma(\mathbf{y} - s^*\mathbf{g}) \cdot \rho_{\sigma_1}(z - s^*)}{\sum_t \Pr[s = t] \cdot \rho_\sigma(\mathbf{y} - t\mathbf{g}) \cdot \rho_{\sigma_1}(z - t)} \\ &= \frac{\rho_\sigma(\mathbf{y} - s^*\mathbf{g}) \cdot \rho_{\sigma_1}(z - s^*)}{\sum_t \rho_\sigma(\mathbf{y} - t\mathbf{g}) \cdot \rho_{\sigma_1}(z - t)}. \end{aligned}$$

Let $\hat{s} \in \mathsf{argmax}_{s^* \in \mathbb{Z}_q} f_{\mathbf{y},z}(s^*)$ be a most likely secret conditioned on the fixed observations $(\mathbf{y}, z)$. We will show that every secret within distance k of $\hat{s}$ has probability at least a constant fraction of

$f_{\mathbf{y},z}(\hat{s})$. Since these probabilities sum to at most one, this will give an upper bound on the largest conditional probability. First, we relate $\hat{s}$ to s'. Define

$$\mu := \frac{1}{a}\left(\frac{\langle \mathbf{y}, \mathbf{g}\rangle}{\sigma^2} + \frac{z}{\sigma_1^2}\right).$$

Completing the square gives $f_{\mathbf{y},z}(s^*) \propto \exp(-\pi a(s^* - \mu)^2)$. Thus $\hat{s}$ is a point of $\mathbb{Z}_q$ closest to μ. In particular, we have $|\hat{s} - \mu| \le |s' - \mu|$. Substituting $\mathbf{y} = s'\mathbf{g} + \mathbf{e}'$ and $z = s' + e_1'$, and then using the inequality in the definition of $\mathcal{T}$ for the fixed s', $\mathbf{e}'$, and e_1', yields

$$\begin{aligned} a|\hat{s} - \mu| \le a|s' - \mu| &= \left|\frac{\langle \mathbf{e}', \mathbf{g}\rangle}{\sigma^2} + \frac{e_1'}{\sigma_1^2}\right| \\ &\le \frac{\delta\sqrt{a}}{\sqrt{2\pi}}. \end{aligned}$$

To avoid messing up the proof with concrete values, we first use $\beta > 0$ and $k \in \mathbb{N}^*$ to denote two constants (we will later set $\beta = 10$ and $k = 4$). Take

$$\sigma_1 \ge \beta k(1 + \delta), \quad \sigma \ge \beta k \tau^{c-1}(1 + \delta). \tag{6}$$

Define $\mathcal{I}_{\hat{s},k} := \mathbb{Z}_q \cap [\hat{s} - k, \hat{s} + k]$. For any $s^* \in \mathcal{I}_{\hat{s},k}$, write $d := s^* - \hat{s}$, so $|d| \le k$. The quadratic expression above gives

$$\begin{aligned} \frac{f_{\mathbf{y},z}(s^*)}{f_{\mathbf{y},z}(\hat{s})} &= \exp\left(-\pi a\left(2d(\hat{s} - \mu) + d^2\right)\right) \\ &\ge \exp\left(-2\pi a k|\hat{s} - \mu| - \pi a k^2\right) \\ &\ge \exp\left(-\frac{\sqrt{10\pi}}{\beta} - \frac{5\pi}{\beta^2}\right). \end{aligned}$$

For the last inequality, we used $\|\mathbf{g}\| \le 2\tau^{c-1}$ and Eq. (6), which imply $a \le 5/(\beta k(1 + \delta))^2$, and hence $ak|\hat{s} - \mu| \le k\delta\sqrt{a}/\sqrt{2\pi} \le \sqrt{5}/(\sqrt{2\pi}\beta)$ and $ak^2 \le 5/\beta^2$. Setting $\beta = 10$ gives $f_{\mathbf{y},z}(s^*) > 0.4 \cdot f_{\mathbf{y},z}(\hat{s})$ for every $s^* \in \mathcal{I}_{\hat{s},k}$.

We assume $q > k$, which holds for any modulus of interest, and then we have $\left|\mathcal{I}_{\hat{s},k}\right| \ge k + 1$ for any $\hat{s} \in \mathbb{Z}_q$. Since $\sum_{s^* \in \mathcal{I}_{\hat{s},k}} f_{\mathbf{y},z}(s^*) \le 1$, we have $1 > \sum_{s^* \in \mathcal{I}_{\hat{s},k}} 0.4 \cdot f_{\mathbf{y},z}(\hat{s}) \ge 0.4(k + 1) \cdot f_{\mathbf{y},z}(\hat{s})$. Setting $k = 4$ gives $f_{\mathbf{y},z}(\hat{s}) < 1/2$.

Now we go back to the joint sampling of $(\mathbf{y}, z)$, which has the same distribution as $(s\mathbf{g} + \mathbf{e}, s + e_1)$. Since $\mathsf{Pr}[(\mathbf{y}, z) \notin \mathcal{T}] = \varepsilon$, we have

$$\begin{aligned} \widetilde{H}_\infty(s|s\mathbf{g} + \mathbf{e}, s + e_1) &= -\log\left(\mathop{\mathbb{E}}_{(\mathbf{y},z)}\left[\max_{s^*} f_{\mathbf{y},z}(s^*)\right]\right) \\ &> -\log\left(\varepsilon \cdot 1 + (1 - \varepsilon) \cdot \frac{1}{2}\right) \\ &\ge 1 - \log(1 + \varepsilon) \ge 1 - 2\varepsilon, \end{aligned}$$

which finishes the proof. □

Next, we present the following lemma, which basically states that $\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}^\mathsf{T} \bmod q$ is statistically close to the uniform distribution on $\mathbb{Z}_q^w$ when $\mathbf{e}$ is a Gaussian noise with parameter greater than the smoothing parameter of $\mathcal{L}_q(\mathbf{G}^\mathsf{T})$.

Lemma 4.2. *Let $\tau > 2$ be a positive integer, q be a prime modulus, and $c := \lceil \log_\tau q \rceil$. Let $n = \mathsf{poly}(\lambda)$ and $w := n \cdot c$. Let $\sigma \geq \tau^{c-1} \cdot \omega\left(\sqrt{\log w}\right)$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^w$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, then the distribution of $\mathbf{G}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q$ is statistically close to the uniform distribution on $\mathbb{Z}_q^w$.*

Proof. Let $f(\cdot)$ denote the probability mass function of $\mathbf{G}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q$, then for any $\mathbf{y} \in \mathbb{Z}_q^w$,

$$\begin{aligned} f(\mathbf{y}) &= \sum_{\mathbf{s}' \in \mathbb{Z}_q^n} \mathsf{Pr}\left[\mathbf{s} = \mathbf{s}'\right] \cdot \frac{1}{\rho_\sigma(\mathbb{Z}^w)} \rho_\sigma(\mathbf{y} - \mathbf{G}^\mathsf{T}\mathbf{s}' + q\mathbb{Z}^w) \\ &= \frac{1}{q^n \rho_\sigma(\mathbb{Z}^w)} \sum_{\mathbf{s}' \in \mathbb{Z}_q^n} \rho_\sigma(\mathbf{y} - \mathbf{G}^\mathsf{T}\mathbf{s}' + q\mathbb{Z}^w) \\ &= \frac{1}{q^n \rho_\sigma(\mathbb{Z}^w)} \rho_{\sigma,\mathbf{y}}(\mathcal{L}_q(\mathbf{G}^\mathsf{T})). \end{aligned}$$

By Lemma 3.9, there exists a negligible ε such that $\eta_\varepsilon(\mathcal{L}_q(\mathbf{G}^\mathsf{T})) \leq \tau^{c-1} \cdot \omega\left(\sqrt{\log w}\right) \leq \sigma$. Then Lemma 3.7 with $\mathbf{\Sigma} = \sigma^2\mathbf{I}_w$ gives us

$$\rho_{\sigma,\mathbf{y}}(\mathcal{L}_q(\mathbf{G}^\mathsf{T})) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot \rho_\sigma(\mathcal{L}_q(\mathbf{G}^\mathsf{T})),$$

which implies that

$$f_{\min} \leq f(\mathbf{y}) \leq \frac{1+\varepsilon}{1-\varepsilon} f_{\min}, \forall \mathbf{y} \in \mathbb{Z}_q^w,$$

where $f_{\min}$ denotes the minimum of $f(\cdot)$. Note also that

$$f_{\min} \leq q^{-w} \leq \frac{1+\varepsilon}{1-\varepsilon} f_{\min}.$$

Therefore,

$$\begin{aligned} \Delta(f, U_{\mathbb{Z}_q^w}) &\leq \frac{1}{2} \sum_{\mathbf{y} \in \mathbb{Z}_q^w} \left| f(\mathbf{y}) - q^{-w} \right| \\ &\leq \frac{1}{2} \sum_{\mathbf{y} \in \mathbb{Z}_q^w} \left| \frac{1+\varepsilon}{1-\varepsilon} f_{\min} - f_{\min} \right| \\ &= \frac{1}{2} q^w \cdot \left(\frac{1+\varepsilon}{1-\varepsilon} - 1 \right) f_{\min} \\ &\leq q^w \cdot \frac{\varepsilon}{1-\varepsilon} \cdot q^{-w} = \frac{\varepsilon}{1-\varepsilon} \end{aligned}$$

which is negligible. □

Now we present the proof for Theorem 4.1.

Proof for Theorem 4.1. Recall the definition of TdSamp and, in particular, that $\mathbf{A} = [\mathbf{B} \| \mathbf{BR} + \mathbf{G}_\tau]$. So, it suffices to prove:

$$\begin{aligned}&(\mathbf{B}, \mathbf{R}, \mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T} \bmod q, \mathbf{s}^\mathsf{T}\mathbf{BR} + \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)\\ &\approx_c (\mathbf{B}, \mathbf{R}, \mathbf{u}_1^\mathsf{T}, \mathbf{u}_2^\mathsf{T}, \mathbf{C}, (\mathbf{u}')^\mathsf{T})\end{aligned}$$

for $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\left[\begin{smallmatrix}\mathbf{e}_1\\ \mathbf{e}_2\end{smallmatrix}\right] \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, $\mathbf{e}' \leftarrow \mathcal{D}^l_{\mathbb{Z},\sigma'}$, $\left[\begin{smallmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{smallmatrix}\right] \xleftarrow{\$} \mathbb{Z}_q^m$ and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

We first fix the Gaussian width parameters: σ_1 for the error of the entropic LWE samples, σ_2 for the error of the gadget leakage, and σ_5 for the noise in the lossiness bound of Lemma 4.1. Let C be the constant in Theorem 3.3, and let

$$\sigma_5 := 40(1+\delta), \qquad \sigma_1 := 4C\gamma\sqrt{m'+l}\cdot\sigma_5, \qquad \sigma_2 := \sigma/2$$

for some $\delta = \omega(\sqrt{\log\lambda})$ such that

$$\sigma_1 \le \sigma', \qquad \sigma_1(1+\sqrt{m'w}) \le \sigma/2, \qquad \sigma_2 \ge 40\tau^{c-1}(1+\delta).$$

It is easy to verify that such a δ exists due to the parameter range of γ: we have $\gamma\sqrt{m'+l}\,(1+\sqrt{m'w}) < 2\tau^{c-1}$, while both σ/τ^{c-1} and $\sigma'/(\gamma\sqrt{m'+l})$ are $\omega(\sqrt{\log\lambda})$. In particular, $\sigma_1^2(1+\sqrt{m'w})^2 + \sigma_2^2 \le \sigma^2/2$. For each $\mathbf{R}$, define

$$\mathbf{\Sigma} := \begin{bmatrix}\sigma_1^2\mathbf{I} & \sigma_1^2\mathbf{R}\\ \sigma_1^2\mathbf{R}^\mathsf{T} & \sigma_1^2\mathbf{R}^\mathsf{T}\mathbf{R} + \sigma_2^2\mathbf{I}\end{bmatrix}.$$

We will prove the required computational indistinguishability via a hybrid argument.

- Hyb_0: $(\mathbf{B}, \mathbf{R}, \mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_1^\mathsf{T} \bmod q, \mathbf{s}^\mathsf{T}\mathbf{BR} + \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_2^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)$

  where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\end{bmatrix} \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}^l_{\mathbb{Z},\sigma'}$.

- Hyb_1: $(\mathbf{B}, \mathbf{R}, \mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_3^\mathsf{T} + (\mathbf{e}_1')^\mathsf{T} \bmod q, \mathbf{s}^\mathsf{T}\mathbf{BR} + \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_3^\mathsf{T}\mathbf{R} + \mathbf{e}_4^\mathsf{T} + (\mathbf{e}_2')^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)$

  where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\mathbf{e}_3 \leftarrow \mathcal{D}^{m'}_{\mathbb{Z},\sigma_1}$, $\mathbf{e}_4 \leftarrow \mathcal{D}^{w}_{\mathbb{Z},\sigma_2}$, $\begin{bmatrix}\mathbf{e}_1'\\ \mathbf{e}_2'\end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sqrt{\sigma^2\mathbf{I}-\mathbf{\Sigma}}}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}^l_{\mathbb{Z},\sigma'}$.

  (In this step, we do the "error massaging", switching $\mathbf{e}_1$ to $\mathbf{e}_3 + \mathbf{e}_1'$ and $\mathbf{e}_2$ to $\mathbf{R}^\mathsf{T}\mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_2'$. This allows us to decompose the information given and separately consider $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_3^\mathsf{T}$ and $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T}$.)

- Hyb_2: $(\mathbf{B}, \mathbf{R}, (\mathbf{u}_1')^\mathsf{T} + (\mathbf{e}_1')^\mathsf{T} \bmod q, (\mathbf{u}_1')^\mathsf{T}\mathbf{R} + \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T} + (\mathbf{e}_2')^\mathsf{T} \bmod q, \mathbf{C}, (\mathbf{u}')^\mathsf{T})$

  where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\mathbf{u}_1' \xleftarrow{\$} \mathbb{Z}_q^{m'}$, $\mathbf{e}_4 \leftarrow \mathcal{D}^{w}_{\mathbb{Z},\sigma_2}$, $\begin{bmatrix}\mathbf{e}_1'\\ \mathbf{e}_2'\end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sqrt{\sigma^2\mathbf{I}-\mathbf{\Sigma}}}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

  (In this step, we switch $\mathbf{s}^\mathsf{T}\mathbf{B} + \mathbf{e}_3^\mathsf{T}$ and $\mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T}$ to uniform, which is supported by the lossiness of $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T}$ on $\mathbf{s}$ and the hardness results of entropic LWE.)

- Hyb_3: $(\mathbf{B}, \mathbf{R}, (\mathbf{u}_1')^\mathsf{T} + (\mathbf{e}_1')^\mathsf{T} \bmod q, (\mathbf{u}_1')^\mathsf{T}\mathbf{R} + (\mathbf{u}_2')^\mathsf{T} + (\mathbf{e}_2')^\mathsf{T} \bmod q, \mathbf{C}, (\mathbf{u}')^\mathsf{T})$

  where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\mathbf{u}_1' \xleftarrow{\$} \mathbb{Z}_q^{m'}$, $\mathbf{u}_2' \xleftarrow{\$} \mathbb{Z}_q^{w}$, $\begin{bmatrix}\mathbf{e}_1'\\ \mathbf{e}_2'\end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{\sigma^2\mathbf{I}-\mathbf{\Sigma}'}}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^{l}$.

  (In this step, we switch $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T}$ itself to uniform random.)

- Hyb_4: $(\mathbf{B}, \mathbf{R}, \mathbf{u}_1^\mathsf{T}, \mathbf{u}_2^\mathsf{T}, \mathbf{C}, (\mathbf{u}')^\mathsf{T})$

  where $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{m'\times w}$, $\mathbf{u}_1 \xleftarrow{\$} \mathbb{Z}_q^{m'}$, $\mathbf{u}_2 \xleftarrow{\$} \mathbb{Z}_q^{w}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^{l}$.

And we prove the following claims one by one:

(1) $\mathsf{Hyb}_0 \approx_{st} \mathsf{Hyb}_1$. Fix any $\mathbf{R} \in \mathbb{Z}_2^{m'\times w}$, and apply Lemma 3.8 with

$$\mathbf{F} = \begin{bmatrix}\mathbf{I} & \mathbf{R}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}, \qquad \mathbf{\Sigma}_1 = \begin{bmatrix}\sigma_1^2\mathbf{I} & 0\\ 0 & \sigma_2^2\mathbf{I}\end{bmatrix},$$

so that $[\mathbf{e}_3^\mathsf{T}\|\mathbf{e}_4^\mathsf{T}]\mathbf{F} = [\mathbf{e}_3^\mathsf{T}\|\mathbf{e}_3^\mathsf{T}\mathbf{R} + \mathbf{e}_4^\mathsf{T}]$ and $\mathbf{F}^\mathsf{T}\mathbf{\Sigma}_1\mathbf{F} = \mathbf{\Sigma}$. For any $\mathbf{x} = [\mathbf{u}^\mathsf{T}\|\mathbf{v}^\mathsf{T}]^\mathsf{T}$, we have $\mathbf{x}^\mathsf{T}\mathbf{\Sigma}\mathbf{x} = \sigma_1^2\|\mathbf{u} + \mathbf{R}\mathbf{v}\|^2 + \sigma_2^2\|\mathbf{v}\|^2$. Since $\mathbf{R} \in \mathbb{Z}_2^{m'\times w}$ implies $\|\mathbf{R}\| \le \sqrt{m'w}$, taking the maximum over unit vectors $\mathbf{x}$ gives

$$\|\mathbf{\Sigma}\| \le \sigma_1^2(1 + \sqrt{m'w})^2 + \sigma_2^2 \le \sigma^2/2.$$

Moreover, $\sigma_1, \sigma_2 = \omega(\sqrt{\log\lambda})$, so $\mathbf{\Sigma}_1 \ge 2\eta_\varepsilon(\mathbb{Z}^m)^2\mathbf{I}$ for some negligible ε. Thus Lemma 3.8 shows that the error $[\mathbf{e}_3^\mathsf{T} + (\mathbf{e}_1')^\mathsf{T}\|\mathbf{e}_3^\mathsf{T}\mathbf{R} + \mathbf{e}_4^\mathsf{T} + (\mathbf{e}_2')^\mathsf{T}]$ in Hyb_1 is statistically close to $\mathcal{D}_{\mathbb{Z},\sigma}^m$ for every $\mathbf{R}$, which proves the claim.

(2) $\mathsf{Hyb}_1 \approx_c \mathsf{Hyb}_2$. To show this, it is sufficient to prove

$$\begin{aligned}&([\mathbf{B}\|\mathbf{C}], \mathbf{s}^\mathsf{T}[\mathbf{B}\|\mathbf{C}] + [\mathbf{e}_3^\mathsf{T}\|(\mathbf{e}')^\mathsf{T}] \bmod q, \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T})\\ &\approx_c ([\mathbf{B}\|\mathbf{C}], [(\mathbf{u}_1')^\mathsf{T}\|(\mathbf{u}')^\mathsf{T}], \mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T}).\end{aligned}$$

Originally, $\mathbf{e}_3$ is sampled from $\mathcal{D}_{\mathbb{Z},\sigma_1}^{m'}$ and $\mathbf{e}'$ from $\mathcal{D}_{\mathbb{Z},\sigma'}^{l}$. We first prove the claim when both $\mathbf{e}_3$ and $\mathbf{e}'$ are sampled with the smaller Gaussian parameter $\sigma_1/2$, which exceeds $C\sqrt{m'+l}\cdot\gamma\sigma_5$. By Lemma 4.1 and our choice of σ_2 and σ_5, the gadget leakage $\mathbf{y} := \mathbf{G}^\mathsf{T}\mathbf{s} + \mathbf{e}_4$ satisfies

$$\begin{aligned}\widetilde{H}_\infty(\mathbf{s} \mid \mathbf{s} + \mathbf{e}_5, \mathbf{y}) &\ge n - \mathsf{negl}(\lambda)\\ &\ge n'\log q + \omega(\log\lambda),\end{aligned}$$

where $\mathbf{e}_5 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_5}^n$. Since $(\mathbf{s}, \mathbf{y})$ is efficiently samplable, Theorem 3.3 with auxiliary information $\mathbf{y}$ and $m' + l$ samples of error width $\sigma_1/2$ proves the claim for this width, assuming $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard. To obtain the original widths, add independent errors from $\mathcal{D}_{\mathbb{Z},\sqrt{\sigma_1^2-\sigma_1^2/4}}^{m'}$ and $\mathcal{D}_{\mathbb{Z},\sqrt{(\sigma')^2-\sigma_1^2/4}}^{l}$ on both sides. Since $\sigma' \ge \sigma_1$, Remark 3.2 shows that this yields the original errors up to negligible statistical distance, and it keeps the uniform side uniform.

(3) $\mathsf{Hyb}_2 \approx_{st} \mathsf{Hyb}_3$. It is sufficient to show $\mathbf{s}^\mathsf{T}\mathbf{G} + \mathbf{e}_4^\mathsf{T} \bmod q$ is statistically indistinguishable from $(\mathbf{u}_2')^\mathsf{T}$, which is exactly given by Lemma 4.2 since $\sigma_2 > \tau^{c-1}\omega\left(\sqrt{\log w}\right)$.

(4) $\mathsf{Hyb}_3 \cong \mathsf{Hyb}_4$. This is a trivial step since $\mathbf{u}_1'$ and $\mathbf{u}_2'$ are independently uniformly random.

Therefore, we conclude that $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_4$, which finishes the proof. □

4.2 LWE with Error Leakage

On top of the vanilla LWE with Trapdoor Hints result we have just proved, we show the hardness of LWE with Trapdoor Hints even in the presence of some arbitrary error leakage as long as the bit-length of the leakage is logarithmically bounded.

We first show a sample-preserving search-to-decision reduction for LWE in the presence of some auxiliary information on the secret $\mathbf{s}$ and the error vector $\mathbf{e}$ by utilizing the Goldreich-Levin theorem for $\mathbb{Z}_q$ introduced as Theorem 3.1.

Lemma 4.3. *Let $n, m = \mathsf{poly}(\lambda)$, $q \leq \mathsf{poly}(\lambda)$ be a prime modulus, χ be an error distribution on $\mathbb{Z}$. Let* aux *be an arbitrary (possibly randomized) auxiliary information function. If for any P.P.T. adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\begin{array}{l|l} \mathcal{A}(\mathbf{A}, \mathbf{b}, \mathsf{aux}(\mathbf{s}, \mathbf{e})) & \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \\ = \mathbf{s} & \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m \\ & \mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q \end{array}\right] \leq \mathsf{negl}(\lambda),$$

then we have

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e})) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}, \mathsf{aux}(\mathbf{s}, \mathbf{e})),$$

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

Proof. We construct the following hybrid arguments: for $0 \leq k \leq m$, define Hyb_k to be

$$\left(\{\mathbf{a}_i\}_{i=1}^m, \{u_i\}_{i=1}^k, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right),$$

where $(\mathbf{a}_1, \ldots, \mathbf{a}_m) \xleftarrow{\$} (\mathbb{Z}_q^n)^m$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} := (e_1, \ldots, e_m) \leftarrow \chi^m$ and $(u_1, \ldots, u_k) \xleftarrow{\$} \mathbb{Z}_q^k$.

For any $1 \leq k \leq m$, to prove $\mathsf{Hyb}_{k-1} \approx_c \mathsf{Hyb}_k$, it is sufficient to prove the following two distributions are computationally indistinguishable:

- D_0: $\left(\{\mathbf{a}_i\}_{i=1}^m, \{u_i\}_{i=1}^{k-1}, \mathbf{s}^\mathsf{T}\mathbf{a}_k \bmod q, e_k \bmod q, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right)$
- D_1: $\left(\{\mathbf{a}_i\}_{i=1}^m, \{u_i\}_{i=1}^{k-1}, u_k, e_k \bmod q, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right)$

Indeed, apply the efficient post-processing map that replaces the pair $(z, e_k \bmod q)$ by $z + e_k \bmod q$ and discards $e_k \bmod q$. On input D_0, it produces $\mathbf{s}^\mathsf{T}\mathbf{a}_k + e_k \bmod q$, and thus the output is exactly Hyb_{k-1}; on input D_1, its output is identically distributed to Hyb_k, since $u_k + e_k \bmod q$ is uniform and independent of the remaining tuple. Suppose a P.P.T. adversary $\mathcal{A}_1$ can distinguish D_0 and D_1 for some $1 \leq k \leq m$ with advantage ε. Define a randomized function

$$f(\mathbf{s}) = \left(\{\mathbf{a}_i\}_{i=k+1}^m, e_k \bmod q, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right),$$

where $(\mathbf{a}_{k+1}, \ldots, \mathbf{a}_m) \xleftarrow{\$} (\mathbb{Z}_q^n)^{m-k}$ and $\mathbf{e} := (e_1, \ldots, e_m) \leftarrow \chi^m$. We construct a P.P.T. adversary $\mathcal{A}_2$ as follows: on input $f(\mathbf{s}) = (\{\mathbf{a}_i\}_{i=k+1}^m, e_k \bmod q, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e}))$, $\mathbf{r} \in \mathbb{Z}_q^n$ and $z \in \mathbb{Z}_q$, it samples $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n$ and $u_i \xleftarrow{\$} \mathbb{Z}_q$ for $1 \leq i \leq k-1$ and set $\mathbf{a}_k := \mathbf{r}$; then it runs $\mathcal{A}_1$ on input

$$\left(\{\mathbf{a}_i\}_{i=1}^m, \{u_i\}_{i=1}^{k-1}, z, e_k \bmod q, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right)$$

and outputs what $\mathcal{A}_1$ outputs. It is easy to see that $\mathcal{A}_2$ can distinguish

$$\left\{(y, \mathbf{r}, \langle \mathbf{r}, \mathbf{s}\rangle) : \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, y \leftarrow f(\mathbf{s}), \mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n\right\}$$

from

$$\left\{(y, \mathbf{r}, u) : \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, y \leftarrow f(\mathbf{s}), \mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n, u \xleftarrow{\$} \mathbb{Z}_q\right\}$$

with advantage ε. By Theorem 3.1, this gives us a P.P.T. adversary $\mathcal{A}_3$ such that

$$\Pr\left[\mathcal{A}_3(y) = \mathbf{s} \;\middle|\; \begin{array}{l}\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n \\ y \leftarrow f(\mathbf{s})\end{array}\right] \geq \Omega\left(\frac{\varepsilon^3}{n \cdot q^2}\right).$$

Therefore, we can construct a P.P.T. adversary $\mathcal{A}$ that tries to find $\mathbf{s} \in \mathbb{Z}_q^n$ given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{b} := \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{e} \bmod q$, and $\mathsf{aux}(\mathbf{s}, \mathbf{e})$: On input $(\mathbf{A}, \mathbf{b}, \mathsf{aux}(\mathbf{s}, \mathbf{e}))$, which can be parsed as

$$\left(\{\mathbf{a}_i\}_{i=1}^m, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right),$$

$\mathcal{A}$ guesses $t \xleftarrow{\$} \mathbb{Z}_q$, runs $\mathcal{A}_3$ on input

$$\left(\{\mathbf{a}_i\}_{i=k+1}^m, t, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=k+1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})\right)$$

and returns its output. Basically $\mathcal{A}$ strips the unused LWE samples and randomly guesses a value for $e_k \bmod q$ to run $\mathcal{A}_3$. Since the guess is correct with $1/q$ probability, we have

$$\Pr\left[\begin{array}{l}\mathcal{A}(\mathbf{A}, \mathbf{b}, \mathsf{aux}(\mathbf{s}, \mathbf{e})) \\ = \mathbf{s}\end{array}\;\middle|\; \begin{array}{l}\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \\ \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m \\ \mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q\end{array}\right] \geq \Omega\left(\frac{\varepsilon^3}{n \cdot q^3}\right).$$

Then according to the assumption in the lemma statement, we obtain $\varepsilon^3/(nq^3) \leq \mathsf{negl}(\lambda)$. As $n, q \leq \mathsf{poly}(\lambda)$, ε must be negligible and we conclude that $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_m$, i.e.,

$$\begin{aligned}&(\{\mathbf{a}_i\}_{i=1}^m, \{\mathbf{s}^\mathsf{T}\mathbf{a}_i + e_i \bmod q\}_{i=1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})) \\ &\approx_c (\{\mathbf{a}_i\}_{i=1}^m, \{u_i\}_{i=1}^m, \mathsf{aux}(\mathbf{s}, \mathbf{e})).\end{aligned}$$

□

Now we present the following theorem stating the hardness of LWE with Trapdoor Hints with a random homogeneous trapdoor and some arbitrary error leakage.

Theorem 4.2. *Let $\tau > 2$ be a positive integer, q be a polynomially bounded prime modulus, and $c := \lceil \log_\tau q \rceil$. Let $n, m \geq 4n \log q + 1$, and l be polynomially bounded integers. Let $r \geq m\tau \cdot \omega(\sqrt{\log m})$. Let $\gamma = \omega(\sqrt{\log \lambda})$ satisfy $\gamma < \frac{\tau^{c-1}}{m\sqrt{m+l}}$, $\sigma \geq \tau^{c-1} \cdot \omega(\sqrt{\log \lambda})$, and $\sigma' \geq \gamma\sqrt{m+l} \cdot \omega(\sqrt{\log \lambda})$.*

Suppose $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some n' satisfying $n' \log q \leq n - \omega(\log \lambda)$; then for any efficiently computable (possibly randomized) function $h : \mathbb{Z}^m \to \{0,1\}^k$ with $k \leq O(\log \lambda)$, we have

$$\begin{aligned}&(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, h(\mathbf{e}), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q) \\ &\approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, h(\mathbf{e}), \mathbf{C}, (\mathbf{u}')^\mathsf{T}),\end{aligned}$$

where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

Proof. Let $m_L := \lfloor m/2 \rfloor$ and $m_R := m - m_L$, so that $m_R \geq m_L \geq (m-1)/2 \geq 2n \log q$. We construct the following hybrid arguments:

- Hyb_0: $(\mathbf{A}, \mathbf{T}, \mathbf{s}^\intercal \mathbf{A} + \mathbf{e}^\intercal \bmod q, h(\mathbf{e}), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

- Hyb_1: $([\mathbf{A}_L \| \mathbf{A}_R], \mathbf{T}_L, [\mathbf{s}^\intercal \mathbf{A}_L + \mathbf{e}_L^\intercal \| \mathbf{s}^\intercal \mathbf{A}_R + \mathbf{e}_R^\intercal] \bmod q, h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $(\mathbf{A}_L, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_L}, q, \tau)$, $\mathbf{A}_R \xleftarrow{\$} \mathbb{Z}_q^{n \times m_R}$, $\mathbf{T}_L \leftarrow \mathsf{ExtTd}(\text{“R”}, q, \tau, \mathbf{A}_L, \mathbf{R}, \mathbf{A}_R, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$, $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

  (In this step, we simulate $\mathbf{A}$ by sampling its left hand side $\mathbf{A}_L$ with an inhomogeneous trapdoor $\mathbf{R}$ and a uniform random right hand side $\mathbf{A}_R$. Then simulate $\mathbf{T}$ with $\mathbf{R}$ using the trapdoor extension algorithm.)

- Hyb_2: $([\mathbf{A}_L \| \mathbf{A}_R], \mathbf{T}_L, [\mathbf{s}^\intercal \mathbf{A}_L + \mathbf{e}_L^\intercal \bmod q \| \mathbf{u}_R^\intercal], h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $(\mathbf{A}_L, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_L}, q, \tau)$, $\mathbf{A}_R \xleftarrow{\$} \mathbb{Z}_q^{n \times m_R}$, $\mathbf{T}_L \leftarrow \mathsf{ExtTd}(\text{“R”}, q, \tau, \mathbf{A}_L, \mathbf{R}, \mathbf{A}_R, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$, $\mathbf{u}_R \xleftarrow{\$} \mathbb{Z}_q^{m_R}$, $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

- Hyb_3: $([\mathbf{A}_L \| \mathbf{A}_R], \mathbf{T}_R, [\mathbf{s}^\intercal \mathbf{A}_L + \mathbf{e}_L^\intercal \bmod q \| \mathbf{u}_R^\intercal], h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $\mathbf{A}_L \xleftarrow{\$} \mathbb{Z}_q^{n \times m_L}$, $(\mathbf{A}_R, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_R}, q, \tau)$, $\mathbf{T}_R \leftarrow \mathsf{ExtTd}(\text{“L”}, q, \tau, \mathbf{A}_R, \mathbf{R}, \mathbf{A}_L, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$, $\mathbf{u}_R \xleftarrow{\$} \mathbb{Z}_q^{m_R}$, $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

  (In this step, we simulate $\mathbf{A}$ by sampling its right hand side $\mathbf{A}_R$ with an inhomogeneous trapdoor $\mathbf{R}$ and a uniform random left hand side $\mathbf{A}_L$. Then simulate $\mathbf{T}$ with $\mathbf{R}$ using the trapdoor extension algorithm.)

- Hyb_4: $([\mathbf{A}_L \| \mathbf{A}_R], \mathbf{T}_R, [\mathbf{u}_L^\intercal \| \mathbf{u}_R^\intercal], h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $\mathbf{A}_L \xleftarrow{\$} \mathbb{Z}_q^{n \times m_L}$, $(\mathbf{A}_R, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_R}, q, \tau)$, $\mathbf{T}_R \leftarrow \mathsf{ExtTd}(\text{“L”}, q, \tau, \mathbf{A}_R, \mathbf{R}, \mathbf{A}_L, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{u}_L \xleftarrow{\$} \mathbb{Z}_q^{m_L}$, $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$, $\mathbf{u}_R \xleftarrow{\$} \mathbb{Z}_q^{m_R}$, $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

- Hyb_5: $(\mathbf{A}, \mathbf{T}, \mathbf{u}^\intercal, h(\mathbf{e}), \mathbf{C}, \mathbf{s}^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q)$

  where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

- Hyb_6: $(\mathbf{A}, \mathbf{T}, \mathbf{u}^\intercal, h(\mathbf{e}), \mathbf{C}, (\mathbf{u}')^\intercal)$

  where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

For $\mathsf{Hyb}_0 \to \mathsf{Hyb}_1$, $\mathsf{Hyb}_2 \to \mathsf{Hyb}_3$, and $\mathsf{Hyb}_4 \to \mathsf{Hyb}_5$, we are switching the way we sample an $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ that is statistically close to uniform random as well as a Gaussian homogeneous trapdoor $\mathbf{T}$ of $\mathbf{A}$. Due to the properties of the algorithms TdSamp, $\mathsf{HTdSamp}$ and ExtTd (Theorem 3.2, Lemma 3.11, and Lemma 3.12), we have $\mathsf{Hyb}_0 \approx_{st} \mathsf{Hyb}_1$, $\mathsf{Hyb}_2 \approx_{st} \mathsf{Hyb}_3$, and $\mathsf{Hyb}_4 \approx_{st} \mathsf{Hyb}_5$.

Next, we prove the following statements:

(1) $\mathsf{Hyb}_1 \approx_c \mathsf{Hyb}_2$: Define

$$\mathsf{aux}(\mathbf{s}, \mathbf{e}_R) := (\mathbf{A}_L, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T} \bmod q, \mathbf{R}, h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q),$$

where $(\mathbf{A}_L, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_L}, q, \tau)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^{l}$. Then it's sufficient to prove

$$(\mathbf{A}_R, \mathbf{s}^\mathsf{T}\mathbf{A}_R + \mathbf{e}_R^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e}_R)) \approx_c (\mathbf{A}_R, \mathbf{u}_R^\mathsf{T}, \mathsf{aux}(\mathbf{s}, \mathbf{e}_R)),$$

where $\mathbf{A}_R \xleftarrow{\$} \mathbb{Z}_q^{n\times m_R}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$. According to Lemma 4.3, we only need to prove for any P.P.T. adversary $\mathcal{A}$ it holds that

$$\Pr\left[\begin{array}{l|l} \mathcal{A}(\mathbf{A}_R, \mathbf{b}, \mathsf{aux}(\mathbf{s}, \mathbf{e}_R)) & \mathbf{A}_R \xleftarrow{\$} \mathbb{Z}_q^{n\times m_R} \\ = \mathbf{s} & \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R} \\ & \mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A}_R + \mathbf{e}_R^\mathsf{T} \bmod q \end{array}\right] \le \mathsf{negl}(\lambda). \tag{7}$$

We apply Theorem 4.1 with m_L in place of m and with $[\mathbf{A}_R \| \mathbf{C}]$ as the extra samples, whose errors have a common width $\gamma\sqrt{m+l} \cdot \omega(\sqrt{\log\lambda}) \le \min\{\sigma, \sigma'\}/\sqrt{2}$; a reduction can then increase these widths to σ and σ' by Remark 3.2. Thus, for any P.P.T. adversary $\mathcal{A}$ it holds that

$$\Pr\left[\mathcal{A}\begin{pmatrix}\mathbf{A}_L, \mathbf{R}, [\mathbf{A}_R \| \mathbf{C}], \\ \mathbf{s}^\top\mathbf{A}_L + \mathbf{e}_L^\top \bmod q, \\ \mathbf{s}^\top[\mathbf{A}_R \| \mathbf{C}] + [\mathbf{e}_R^\top \| (\mathbf{e}')^\top] \bmod q\end{pmatrix} = \mathbf{s} \;\middle|\; \begin{array}{l} (\mathbf{A}_L, \mathbf{R}) \leftarrow \mathsf{TdSamp}(1^n, 1^{m_L}, q, \tau), \\ \mathbf{A}_R \xleftarrow{\$} \mathbb{Z}_q^{n\times m_R}, \mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l} \\ \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}, \\ \mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}, \mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^{l} \end{array}\right] \le \mathsf{negl}(\lambda). \tag{8}$$

In Eq. (7), the only extra information provided by $\mathsf{aux}(\mathbf{s}, \mathbf{e}_R)$ compared to Eq. (8) is $h(\mathbf{e}_L \| \mathbf{e}_R)$, which has at most $O(\log\lambda)$ bits according to the definition of h. By enumerating $h(\mathbf{e}_L \| \mathbf{e}_R)$, one can reduce the experiment of Eq. (8) to that of Eq. (7). Therefore, Eq. (7) is implied by Eq. (8).

(2) $\mathsf{Hyb}_3 \approx_c \mathsf{Hyb}_4$: It is sufficient to prove

$$\begin{aligned}&(\mathbf{A}_L, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T} \bmod q, h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q) \\ &\approx_c (\mathbf{A}_L, \mathbf{u}_L^\mathsf{T}, h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q)\end{aligned}$$

Define

$$\mathsf{aux}(\mathbf{s}, \mathbf{e}_L) := (h(\mathbf{e}_L \| \mathbf{e}_R), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q),$$

where $\mathbf{e}_R \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_R}$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^{l}$. Then it is sufficient to prove

$$(\mathbf{A}_L, \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{s}, \mathbf{e}_L)) \approx_c (\mathbf{A}_L, \mathbf{u}_L^\mathsf{T}, \mathsf{aux}(\mathbf{s}, \mathbf{e}_L)),$$

where $\mathbf{A}_L \xleftarrow{\$} \mathbb{Z}_q^{n \times m_L}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L}$. Similar to the previous proof, by Lemma 4.3 we only need to prove

$$\Pr\left[\begin{array}{l|l} \mathcal{A}(\mathbf{A}_L, \mathbf{b}, \mathsf{aux}(\mathbf{s}, \mathbf{e}_L)) & \mathbf{A}_L \xleftarrow{\$} \mathbb{Z}_q^{n \times m_L} \\ = \mathbf{s} & \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m_L} \\ & \mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A}_L + \mathbf{e}_L^\mathsf{T} \bmod q \end{array}\right] \leq \mathsf{negl}(\lambda),$$

which is implied by the hardness of the search LWE problem since the output length of h is at most $O(\log \lambda)$ bits.

(3) $\mathsf{Hyb}_5 \approx_c \mathsf{Hyb}_6$: Directly implied by the hardness of the decision LWE problem.

□

A caveat in Lemma 4.3 and Theorem 4.2 is that the modulus q needs to be polynomially bounded. However, we note that there is an alternative proof approach that can deal with a super-polynomial modulus q while only requiring the hardness of LWE against polynomial-time adversaries but with super-polynomial modulus-to-noise ratio. This route of proof works by adapting the sample-preserving reduction in [MM11] and the entropic LWE result in [BD20] to the settings with a short error leakage, and integrating them into the proof of Theorem 4.1 with similar trapdoor dual-mode switching techniques. We present the formal statement below but omit the full proof, as the underlying technical arguments are the same as those already shown in the proofs of Theorem 4.1 and Theorem 4.2. We defer the reader to Section A for a proof sketch.

Theorem 4.3. *Let $\tau = \mathsf{super\text{-}poly}(\lambda)$ be a positive integer, $q > \tau$ be a prime modulus, and $c := \lceil \log_\tau q \rceil$. Let $n, m > 4n \log q$, and l be polynomially bounded integers. Let $r \geq m\tau \cdot \omega(\sqrt{\log m})$. Let $\gamma = \omega(\sqrt{\log \lambda})$ be polynomially bounded, $\sigma \geq \tau^{c-1} \cdot \omega(\sqrt{\log \lambda})$, and $\sigma' \geq \gamma\sqrt{m+l} \cdot \omega(\sqrt{\log \lambda})$ be Gaussian width parameters.*

Suppose $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some n' satisfying $n' \log q \leq n - \omega(\log \lambda)$; then for any efficiently computable (possibly randomized) function $h : \mathbb{Z}^m \to \{0,1\}^k$ with $k \leq O(\log \lambda)$, we have

$$\begin{aligned} &(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, h(\mathbf{e}), \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q) \\ &\approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, h(\mathbf{e}), \mathbf{C}, (\mathbf{u}')^\mathsf{T}), \end{aligned}$$

where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^l$.

Remark 4.1. As a comparison, our vanilla LWE with Trapdoor Hints hardness result, Theorem 4.1, works with [MP12]'s specific trapdoor sampling scheme and actually has tighter parameters than Theorems 4.2 and 4.3, which works with random Gaussian homogeneous trapdoors. Theorem 4.1 already gives insights on the hardness of LWE in the presence of trapdoors and can enable potential applications, but publishing the inhomogeneous trapdoor $\mathbf{R}$ results in more leakage on the LWE sample's structure, blocking us from obtaining the strengthened version with error leakage. The reason we present Theorems 4.2 and 4.3 is to reach a more generic form of our results and to achieve stronger security for our applications.

5 Euclidean Ball Obfuscation with Distributional VBB security

As an application of our LWE with Trapdoor Hints theorem, we give an obfuscation scheme for Euclidean ball functions with distributional virtual black-box security.

5.1 Definitions

Euclidean ball functions are a fuzzy version of point functions in the following form

$$\mathsf{EB}_{t,\mathbf{x}}(\mathbf{x}') = \begin{cases} 1 & \text{if } \|\mathbf{x}-\mathbf{x}'\| \le t \\ 0 & \text{if } \|\mathbf{x}-\mathbf{x}'\| > t \end{cases},$$

i.e., membership functions for ℓ_2 balls in m-dimensional vector space. For simplicity, we directly consider the obfuscation scheme syntax specific for this Euclidean ball function family.

Definition 5.1 (Distributional-VBB Euclidean ball obfuscator)**.** *Let $t > 0$ be a distance threshold. Let X be a space of points and $\mathcal{X}$ be a distribution on X. A $(t, X, \mathcal{X})$-distributional VBB Euclidean ball obfuscator consists of two P.P.T. algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *that satisfy the following properties:*

- *Functional preserving: for any $\mathbf{x} \in X$, it holds that*

$$1 - \Pr_{\mathcal{O} \leftarrow \mathsf{Obf}(\mathbf{x})} [\forall \mathbf{x}' \in X, \mathsf{Eval}(\mathcal{O}, \mathbf{x}') = \mathsf{EB}_{t,\mathbf{x}}(\mathbf{x}')] \le \mathsf{negl}(\lambda)$$

- *Distributional virtual black-box: for any P.P.T. adversary $\mathcal{A}$, there exists a P.P.T. simulator* Sim *with oracle access to* EB *such that for any efficiently computable predicate $\phi : X \to \{0,1\}$, it holds that*

$$\left| \Pr_{\mathbf{x} \leftarrow \mathcal{X}} \left[\mathcal{A}(\mathsf{Obf}(\mathbf{x})) = \phi(\mathbf{x})\right] - \Pr_{\mathbf{x} \leftarrow \mathcal{X}} \left[\mathsf{Sim}^{\mathsf{EB}_{t,\mathbf{x}}}(\lambda) = \phi(\mathbf{x})\right] \right| \le \mathsf{negl}(\lambda)$$

In practice, we often instead use the following notion of distributional indistinguishability (with augmented distribution) for obfuscating evasive programs, which implies distributional VBB security [WZ17].

- *Distributional indistinguishability: there exists a P.P.T. simulator* Sim *such that for any efficiently computable predicate ϕ, it holds that*

$$(\mathsf{Obf}(\mathbf{x}), \phi(\mathbf{x})) \approx_c (\mathsf{Sim}(\lambda), \phi(\mathbf{x})),$$

where $\mathbf{x} \leftarrow \mathcal{X}$.

5.2 Our Construction

Now we give a construction for Euclidean ball obfuscation scheme in Fig. 2, utilizing our previous result on the hardness of LWE with Trapdoor Hints.

Theorem 5.1. *Let $\tau > 2$ be a positive integer, q be a polynomially bounded prime modulus, and $c := \lceil \log_\tau q \rceil$. Let n and $m \ge 4n \log q + 1$ be polynomially bounded integers. Let $r = m\tau \cdot \omega(\sqrt{\log m})$. Let $t \ll q/(4rm)$, $\sigma \ge \tau^{c-1} \cdot \omega(\sqrt{\log \lambda})$, and $\sigma\left(\sqrt{m} + \omega(\sqrt{\log \lambda})\right) < B < (q-t)/2$ (see Remark 5.1).*

Let $X := \{\mathbf{x} \in \mathbb{Z}^m : \|\mathbf{x}\| \le B\}$, and $\mathcal{X}$ be the renormalized distribution of $\mathcal{D}^m_{\mathbb{Z},\sigma}$ restricted to X. Suppose $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some n' satisfying $n' \log q \le n - \omega(\log \lambda)$ and some $\gamma = \omega(\sqrt{\log \lambda})$ with $\gamma < \frac{\tau^{c-1}}{m^{1.5}}$, then the construction in Fig. 2 gives a $(t, X, \mathcal{X})$-distributional VBB Euclidean ball obfuscation.

Euclidean ball obfuscation based on LWE with Trapdoor Hints

Public parameters: n, m, q, t, r.

$\mathsf{Obf}(\mathbf{x} \in \mathbb{Z}^m)$:

1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$.
2. Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. Output $O := (\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q)$.

$\mathsf{Eval}(O, \mathbf{x}' \in \mathbb{Z}^m)$:

1. Parse O as $(\mathbf{A}, \mathbf{T}, \mathbf{b}^\mathsf{T})$.
2. Run $\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{b} - \mathbf{x}' \bmod q)$; if it outputs $\perp$, output 0 and abort.
3. Output 1.

Figure 2: Euclidean ball obfuscation based on LWE with Trapdoor Hints

Proof. It is easy to verify the functional preserving property. Note that $\mathbf{b} - \mathbf{x}' = \mathbf{A}^\mathsf{T}\mathbf{s} + (\mathbf{x} - \mathbf{x}')$. With overwhelming probability $\mathsf{HTdSamp}$ will output a full-row-rank $\mathbf{A}$ with $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) = q$ and a full-rank $\mathbf{T}$ with $\|\mathbf{T}\| \le O(rm)$ (Theorem 3.2, Lemma 3.2, Lemma 3.11, and Lemma 3.13; note that $q > \tau^{c-1} > \gamma m^{1.5} \ge \sqrt{m}$). In that case $t \ll q/(4\|\mathbf{T}\|)$, and for any $\mathbf{x}' \in X$ we have $\|\mathbf{x} - \mathbf{x}'\| \le 2B < q - t$. So by Lemma 3.10, $\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{b} - \mathbf{x}' \bmod q)$ outputs $(\mathbf{s}, \mathbf{x} - \mathbf{x}')$ if $\|\mathbf{x} - \mathbf{x}'\| \le t$ and $\perp$ otherwise. Hence Eval outputs 1 if and only if $\|\mathbf{x} - \mathbf{x}'\| \le t$.

For distributional VBB security/Distributional indistinguishability, let $\mathsf{Sim}(\lambda)$ be a simulator that samples $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and outputs $(\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T})$. We need to show

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \phi(\mathbf{x})) \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, \phi(\mathbf{x})),$$

where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \mathcal{X}$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$. Note that $B > \sigma\big(\sqrt{m} + \omega(\sqrt{\log\lambda})\big)$. Thus, by Lemma 3.5 $\mathcal{X}$ is statistically close to $\mathcal{D}_{\mathbb{Z},\sigma}^m$ and we can replace $\mathcal{X}$ with $\mathcal{D}_{\mathbb{Z},\sigma}^m$ in the above statement. Then this indistinguishability is directly implied by Theorem 4.2, setting h to be the predicate ϕ and without even considering extra LWE samples. □

Remark 5.1. The upper bound $B < (q - t)/2$ cannot be relaxed beyond $B < (q + 1)/2$. Otherwise, X contains the vectors $\mathbf{x}$ and $\mathbf{x}'$ whose first entries are $(q + 1)/2$ and $-(q - 1)/2$ respectively and whose other entries are 0. Then $\mathbf{b} - \mathbf{x}' \equiv \mathbf{b} - \mathbf{x} \bmod q$, so Eval gives the same output on $\mathbf{x}'$ as on $\mathbf{x}$, although $\|\mathbf{x} - \mathbf{x}'\| = q > t$.

Together with the lower bound on B, the upper bound requires $\sigma\big(\sqrt{m} + \omega(\sqrt{\log\lambda})\big) < (q - t)/2$. Since $\sigma \ge \tau^{c-1} \cdot \omega(\sqrt{\log\lambda})$, this means $q/\tau^{c-1} \ge \sqrt{m} \cdot \omega(\sqrt{\log\lambda})$. It holds, for example, when $q > \tau^c/2$ and $\tau \ge 11\sqrt{m} \cdot \omega(\sqrt{\log\lambda})$: then $\sigma := \tau^c/(11\sqrt{m}) \ge \tau^{c-1} \cdot \omega(\sqrt{\log\lambda})$, and any B with $2\sigma\sqrt{m} \le B < (q - t)/2$ satisfies both bounds, where such B exists because $2\sigma\sqrt{m} = 2\tau^c/11 < 4q/11 < 3q/8 < (q - t)/2$ as $t < q/4$.

Remark 5.2. As an analogy to reusable fuzzy extractors [Boy04, CFP+16, ACEK17] and same-point composable obfuscated point functions [FF20], we can define a reusability notion for obfuscated

fuzzy point functions/Euclidean Ball Obfuscation. Namely, in the security game the adversary will not only obtain $\mathsf{Obf}(\mathbf{x})$ and $\phi(\mathbf{x})$, but can also adaptively submit adversarial shifts $\boldsymbol{\delta}_i$ satisfying $\|\boldsymbol{\delta}_i\| \leq t$ and obtain a sample $\mathsf{Obf}(\mathbf{x}+\boldsymbol{\delta}_i)$. The reusable security requires that $\mathsf{Obf}(\mathbf{x})$ and $\{\mathsf{Obf}(\mathbf{x}+\boldsymbol{\delta}_i)\}_i$ can all be simulated in the presence of $\phi(\mathbf{x})$. Here we point out that our scheme can be easily adapted to satisfy this reusability notion in the CRS model using a simple technique from [ACEK17]: instead of sampling fresh $(\mathbf{A}, \mathbf{T})$ for every obfuscation process, we sample and set the public parameters/common reference string to be $(\mathbf{A}, \mathbf{T})$ and let $\mathsf{Obf}(\mathbf{x})$ just contain $\mathbf{s}^\mathsf{T}\mathbf{A}+\mathbf{x}^\mathsf{T} \bmod q$. Based on Theorem 4.2, an easy simulation strategy is given by the following indistinguishability result:

$$\begin{aligned}&(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \phi(\mathbf{x}), \{\mathbf{s}_i^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} + \boldsymbol{\delta}_i^\mathsf{T} \bmod q\}_i)\\ \approx_c\ &(\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, \phi(\mathbf{x}), \{\mathbf{u}^\mathsf{T} + \Delta\mathbf{s}_i^\mathsf{T}\mathbf{A} + \boldsymbol{\delta}_i^\mathsf{T} \bmod q\}_i),\end{aligned}$$

where $\Delta\mathbf{s}_i$ are uniform random vectors for refreshing the LWE secret. We omit the formal definitions and proof here, as this reusable security notion is also formally considered in TDRE's security definitions and the corresponding proofs. We refer the readers to Section 6 for more details.

Remark 5.3. We can also extend our scheme to obfuscate Euclidean-ball functions with multi-bit output, i.e., $\mathsf{EB}_{t,\mathbf{x},\mathbf{y}}(\mathbf{x}')$ outputs $\mathbf{y} \in \{0,1\}^l$ if $\|\mathbf{x} - \mathbf{x}'\| \leq t$ and $\perp$ otherwise, by utilizing the extra LWE samples. Now $\mathsf{Obf}(\mathbf{x}, \mathbf{y})$ will sample and output

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}^\mathsf{T} \bmod q).$$

On input $\mathbf{x}'$ that is close to $\mathbf{x}$, Eval can recover $\mathbf{s}$ and then use it to decrypt and compute $\mathbf{y}$. The distributional indistinguishability still directly follows from Theorem 4.2 and we omit the proof. Actually, this is also the template we use to build our TDRE scheme in the following section.

6 Threshold Difference Revealing Encryption

In this section, we formally define our new primitive Threshold Difference Revealing Encryption (TDRE) and then give a construction based on our results on the hardness of LWE with Trapdoor Hints.

6.1 Definitions

Definition 6.1 (Threshold difference revealing encryption)**.** *Let X and Y be the space of input and message payload, respectively (assume X is the subset of some linear space). An (X, Y)-threshold difference revealing encryption scheme (TDRE) is a tuple of P.P.T. algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Rev})$ *with the following interfaces:*

- $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$*: a key-generation algorithm that takes as input the security parameter 1^λ and outputs a public key* pk*. Since there is no secret key in this definition, the public key can be treated as a common reference string. However, an optional secret key can also be defined to support unconditional decryption, as we will remark later.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x, y)$ *where $x \in X$ and $y \in Y$: an encryption algorithm that takes the public key* pk*, an input $x \in X$ and a message payload $y \in Y$ and outputs a ciphertext* ct*.*

- $(x^*, y^*)/\perp \leftarrow \mathsf{Rev}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$: *a revelation algorithm that takes the public key* pk *and two ciphertexts* ct_1, ct_2 *and conditionally outputs a revelation or rejects.*

Remark 6.1. A minor point about which readers might be confused in Definition 6.1 is that there is no secret key corresponding to the public key pk and no decryption interfaces. We note that a notion of secret key sk, which supports unconditionally decrypting $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x, y)$ to x and y, can indeed be introduced. This unconditional secret-key decryption functionality is easily achievable by combining any correct TDRE scheme with a public-key encryption scheme. However, we omit the interface and correctness definition of secret key decryption for simplicity, as the revelation functionality Rev is usually sufficient for intended applications.

6.1.1 Correctness

The correctness property of a TDRE scheme is defined below. The requirements are two-sided—the TDRE scheme should correctly compute the revelation function f if and only if the distance of two underlying inputs is below the threshold and reject otherwise.

Definition 6.2 (Correctness of TDRE)**.** *Let* dis *be a metric on* X *and* f *be a function on* $Y \times Y$*. Let* $t > 0$ *be a distance threshold. An* (X, Y)*-TDRE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Rev})$ *is* (dis, t, f)*-correct if it satisfies the following property: with overwhelming probability over* $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$*, it holds that for any* $x_1, x_2 \in X$ *and any* $y_1, y_2 \in Y$*,*

$$\mathsf{Rev}\begin{pmatrix}\mathsf{pk},\\ \mathsf{Enc}(\mathsf{pk}, x_1, y_1),\\ \mathsf{Enc}(\mathsf{pk}, x_2, y_2)\end{pmatrix} = \begin{cases}(x_1 - x_2, f(y_1, y_2)) & \text{if } \mathsf{dis}(x_1, x_2) \le t\\ \perp & \text{if } \mathsf{dis}(x_1, x_2) > t\end{cases} \tag{9}$$

6.1.2 Security

To model the security properties of a TDRE scheme, we introduce the following notion of *distributional indistinguishability*. This notion introduces a pair of real-world and ideal-world experiments in Fig. 3 and requires them to be computationally indistinguishable.

The experiments proceed in two phases. In the first phase, the simulator does not have the message, which models the semantic security of the message payload; furthermore, the simulator only obtains a leakage $\mathsf{lk}(x_i)$ instead of x_i itself, which intuitively argues that the ciphertext does not tell $\mathcal{A}$ anything more about $\mathsf{aux}(x_i)$ than $\mathsf{lk}(x_i)$ does (to make it a non-trivial requirement, $\mathsf{aux}(x)$ should be undetermined given $\mathsf{lk}(x)$). On the other hand, the second phase models the requirement that the revelation process only leaks the difference between inputs and the message payload function evaluation.

Definition 6.3 (Distributional indistinguishability)**.** *An* (X, Y)*-TDRE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Rev})$ *satisfies* $(\mathcal{X}, \mathsf{lk}, \mathsf{aux})$-distributional indistinguishability *if there exists a (possibly stateful) P.P.T. simulator* Sim *such that for any P.P.T. adversary* $\mathcal{A}$*, it holds that*

$$\left|\Pr\left[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Real}}(1^\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Ideal}}(1^\lambda) = 1\right]\right| \le \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Real}}$ *and* $\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Ideal}}$ *are defined in Fig. 3.*

$\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Real}}(1^\lambda)$:

1. Sample $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$ and give it to $\mathcal{A}$;
2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
   (a) $\mathcal{A}$ outputs (i, y_i) where $y_i \in Y$;
   (b) Sample $x_i \leftarrow \mathcal{X}$,
   run $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, x_i, y_i)$,
   and send $(\mathsf{ct}_i, \mathsf{aux}(x_i))$ to $\mathcal{A}$;
3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
   (a) $\mathcal{A}$ outputs $(i, j, \delta_{i,j}, y_{i,j})$ where $\delta_{i,j} \in X$ and $y_{i,j} \in Y$ such that $\mathsf{dis}(x_i, x_i + \delta_{i,j}) \le t$;
   (b) Generate $\mathsf{ct}_{i,j} \leftarrow \mathsf{Enc}(\mathsf{pk}, x_i + \delta_{i,j}, y_{i,j})$ and send $\mathsf{ct}_{i,j}$ to $\mathcal{A}$;
4. $\mathcal{A}$ outputs a bit b as the output of the experiment.

$\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-DistrIND-Ideal}}(1^\lambda)$:

1. Sample $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$ and give it to $\mathcal{A}$;
2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
   (a) $\mathcal{A}$ outputs (i, y_i) where $y_i \in Y$;
   (b) Sample $x_i \leftarrow \mathcal{X}$,
   run $\widetilde{\mathsf{ct}}_i \leftarrow \mathsf{Sim}(\mathsf{pk}, i, \mathsf{lk}(x_i))$,
   and send $(\widetilde{\mathsf{ct}}_i, \mathsf{aux}(x_i))$ to $\mathcal{A}$;
3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
   (a) $\mathcal{A}$ outputs $(i, j, \delta_{i,j}, y_{i,j})$ where $\delta_{i,j} \in X$ and $y_{i,j} \in Y$ such that $\mathsf{dis}(x_i, x_i + \delta_{i,j}) \le t$;
   (b) Generate $\widetilde{\mathsf{ct}}_{i,j} \leftarrow \mathsf{Sim}(\mathsf{pk}, i, j, \delta_{i,j}, f(y_{i,j}, y_i))$ and send $\widetilde{\mathsf{ct}}_{i,j}$ to $\mathcal{A}$;
4. $\mathcal{A}$ outputs a bit b as the output of the experiment.

Figure 3: TDRE Distributional Indistinguishability Experiments

Another security notion that we would want a TDRE scheme to satisfy is *fuzzy second-preimage resistance*. It states that no efficient adversary, seeing $\mathsf{Enc}(\mathsf{pk}, x, y)$, can come up with (x', y') and utilize the revelation algorithm to learn the message y. This can also be understood as an input hiding property in the sense that if an adversary given $\mathsf{Enc}(\mathsf{pk}, x, y)$ can find x' close enough to x, then the scheme clearly does not satisfy fuzzy second-preimage resistance.

Definition 6.4 (Fuzzy second-preimage resistance)**.** *An (X, Y)-TDRE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Rev})$ *satisfies* weak fuzzy second-preimage resistance *for input distribution $\mathcal{X}$, if for any P.P.T. adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Weak}}(1^\lambda) = 1\right] \le \mathsf{negl}(\lambda).$$

Similarly, we say it satisfies strong fuzzy second-preimage resistance *for input distribution $\mathcal{X}$, if for any P.P.T. adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Strong}}(1^\lambda) = 1\right] \le \mathsf{negl}(\lambda).$$

$\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Weak}}$ *and* $\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Strong}}$ *are defined in Fig. 4.*

6.2 Our Construction

Construction 6.1. Set parameters as follows: let $\tau > 2$ be a positive integer, $q \ge \mathsf{super\text{-}poly}(\lambda)$ be a prime modulus, and $c := \lceil \log_\tau q \rceil$; let n, $m > 4n \log q$, and l be polynomially bounded integers; let $r = m\tau \cdot \omega(\sqrt{\log m})$; let $\gamma = \mathsf{poly}(\lambda)$ and $\sigma' \ge \gamma \cdot \mathsf{super\text{-}poly}(\lambda)$; let $t \ll q/(4rm) = q/(4m^2\tau \cdot \omega(\sqrt{\log m}))$.

$\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Weak}}(1^\lambda)$:

1. Sample $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$ and give it to $\mathcal{A}$;
2. $\mathcal{A}$ may adaptively make queries of the following form:
   (a) $\mathcal{A}$ outputs (i, y_i) where $y_i \in Y$;
   (b) Sample $x_i \leftarrow \mathcal{X}$, run $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, x_i, y_i)$;
   (c) Send ct_i to $\mathcal{A}$;
3. $\mathcal{A}$ outputs an index i, $\widetilde{x} \in X$ and $\widetilde{y} \in Y$; the output of the experiment is 1 if $\mathsf{Rev}(\mathsf{pk}, \mathsf{ct}_i, \mathsf{Enc}(\mathsf{pk}, \widetilde{x}, \widetilde{y})) = (x_i - \widetilde{x}, f(y_i, \widetilde{y}))$ and 0 otherwise.

$\mathsf{Exp}^{\mathcal{A}}_{\mathsf{TDRE-PreRst-Strong}}(1^\lambda)$:

1. Sample $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$ and give it to $\mathcal{A}$;
2. $\mathcal{A}$ may adaptively make queries of the following form:
   (a) $\mathcal{A}$ outputs (i, y_i) where $y_i \in Y$;
   (b) Sample $x_i \leftarrow \mathcal{X}$, run $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, x_i, y_i)$;
   (c) Send ct_i to $\mathcal{A}$;
3. $\mathcal{A}$ outputs an index i, $\widetilde{x} \in X$ and $\widetilde{y} \in Y$; the output of the experiment is 1 if $\mathsf{Rev}(\mathsf{pk}, \mathsf{ct}_i, \mathsf{Enc}(\mathsf{pk}, \widetilde{x}, \widetilde{y})) \neq \perp$ and 0 otherwise.

Figure 4: TDRE Fuzzy Second-Preimage Resistance Experiments

Let $B < (q-t)/2$, $X := \{\mathbf{x} \in \mathbb{Z}^m : \|\mathbf{x}\| \le B\}$, and $Y := \{0,1\}^l$. Construct an (X, Y)-TDRE scheme as in Fig. 5. The security theorems below additionally require $B > \sigma\left(\sqrt{m} + \omega(\sqrt{\log\lambda})\right)$ for some $\sigma \ge \tau^{c-1} \cdot \omega(\sqrt{\log\lambda})$, which is indeed satisfiable (see Remark 5.1).

We first prove the *two-sided correctness* of Construction 6.1.

Theorem 6.1. *Let* $\mathsf{dis}(\mathbf{x}_1, \mathbf{x}_2) := \|\mathbf{x}_1 - \mathbf{x}_2\|$ *and* $f(\mathbf{y}_1, \mathbf{y}_2) := \mathbf{y}_1 \oplus \mathbf{y}_2$. *Then Construction 6.1 is* (dis, t, f)*-correct as per Definition 6.2*

Proof. The verification is straightforward. Note that $\mathbf{b}_1 - \mathbf{b}_2 = \mathbf{A}^\mathsf{T}(\mathbf{s}_1 - \mathbf{s}_2) + (\mathbf{x}_1 - \mathbf{x}_2)$. Assume $\mathsf{HTdSamp}$ outputs a full-row-rank $\mathbf{A}$ with $\lambda_1(\mathcal{L}_q(\mathbf{A}^\mathsf{T})) = q$ and a full-rank $\mathbf{T}$ satisfying $\|\mathbf{T}\| \le O(rm)$, which happens with overwhelming probability according to Theorem 3.2, Lemma 3.2, Lemma 3.11, and Lemma 3.13. In that case $t \ll q/(4\|\mathbf{T}\|)$, and for any $\mathbf{x}_1, \mathbf{x}_2 \in X$ we have $\|\mathbf{x}_1 - \mathbf{x}_2\| \le 2B < q - t$ by the definition of X. So by Lemma 3.10, $\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{b}_1 - \mathbf{b}_2 \bmod q)$ outputs $(\mathbf{s}_1 - \mathbf{s}_2, \mathbf{x}_1 - \mathbf{x}_2)$ if $\|\mathbf{x}_1 - \mathbf{x}_2\| \le t$ and $\perp$ otherwise. Thus, when $\|\mathbf{x}_1 - \mathbf{x}_2\| > t$, Rev outputs $\perp$. When $\|\mathbf{x}_1 - \mathbf{x}_2\| \le t$, we have $\hat{\mathbf{s}} = \mathbf{s}_1 - \mathbf{s}_2$, and therefore $\hat{\mathbf{y}} = \mathsf{Round}(\mathbf{e}_1 - \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot (\mathbf{y}_1 - \mathbf{y}_2) \bmod q) = \mathbf{y}_1 \oplus \mathbf{y}_2$. □

Next we analyze the security of Construction 6.1. We first present the distributional indistinguishability property of this construction.

Theorem 6.2. *Let* $\sigma \ge \tau^{c-1} \cdot \omega(\sqrt{\log\lambda})$, $B > \sigma\left(\sqrt{m} + \omega(\sqrt{\log\lambda})\right)$, $\mathcal{X}$ *be the renormalized distribution of* $\mathcal{D}^m_{\mathbb{Z},\sigma}$ *restricted to* X, aux *be any efficiently computable function whose output length is at most* $O(\log\lambda)$ *bits, and* lk *be empty (i.e., the function that always outputs* $\perp$*).*

Suppose $\gamma = \omega(\sqrt{\log\lambda})$ *and* $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ *is hard for some* n' *satisfying* $n' \log q \le n - \omega(\log\lambda)$*; then Construction 6.1 satisfies* $(\mathcal{X}, \mathsf{lk}, \mathsf{aux})$*-distributional indistinguishability defined in Definition 6.3.*

Proof. Note that $B > \sigma\left(\sqrt{m} + \omega(\sqrt{\log\lambda})\right)$ and thus by Lemma 3.5 $\mathcal{X}$ is statistically close to $\mathcal{D}^m_{\mathbb{Z},\sigma}$. Therefore, for simplicity we directly replace $\mathcal{X}$ with $\mathcal{D}^m_{\mathbb{Z},\sigma}$ in the proof. Let $\sigma^* := \gamma\sqrt{m+l} \cdot \omega(\sqrt{\log\lambda})$, then $\sigma'/\sigma^* \ge \mathsf{super\text{-}poly}(\lambda)$. Let $\mathcal{A}$ be a P.P.T. adversary for the security experiments defined in Fig. 3

Threshold difference revealing encryption based on LWE with Trapdoor Hints

Public parameters: $n, m, q, t, r, l, \sigma'$ (all functions of λ).

$\mathsf{Gen}(1^\lambda)$:

1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$.
2. Sample $\mathbf{C} \leftarrow \mathbb{Z}_q^{n \times l}$.
3. Output $\mathsf{pk} := (\mathbf{A}, \mathbf{T}, \mathbf{C})$.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x} \in \mathbb{Z}_q^m, \mathbf{y} \in \{0,1\}^l)$:

1. Parse pk as $(\mathbf{A}, \mathbf{T}, \mathbf{C})$;
2. Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$;
3. Set $\mathsf{ct} := (\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{x}^\mathsf{T} \bmod q, \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}^\mathsf{T} \bmod q)$;
4. Output ct.

$\mathsf{Rev}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$:

1. Parse pk as $(\mathbf{A}, \mathbf{T}, \mathbf{C})$, ct_1 as $(\mathbf{b}_1^\mathsf{T}, (\mathbf{b}_1')^\mathsf{T})$ and ct_2 as $(\mathbf{b}_2^\mathsf{T}, (\mathbf{b}_2')^\mathsf{T})$;
2. Run $\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{b}_1 - \mathbf{b}_2 \bmod q)$; if it outputs $\perp$, output $\perp$ and abort, otherwise let its output be $(\hat{\mathbf{s}}, \hat{\mathbf{e}})$;
3. Compute $\hat{\mathbf{y}} := \mathsf{Round}(\mathbf{b}_1' - \mathbf{b}_2' - \mathbf{C}^\mathsf{T}\hat{\mathbf{s}} \bmod q)$, where $\mathsf{Round}(\mathbf{v})$ means applying the following function RD on every entry of $\mathbf{v}$:

$$\mathsf{RD}(v) := \begin{cases} 0, & 0 \le v < \frac{q}{4} \text{ or } \frac{3q}{4} \le v < q \\ 1, & \frac{q}{4} \le v < \frac{3q}{4} \end{cases}.$$

4. Output $(\hat{\mathbf{e}}, \hat{\mathbf{y}})$.

Figure 5: Threshold difference revealing encryption based on LWE with Trapdoor Hints

and $N = \mathsf{poly}(\lambda)$ be the number of phase-1 queries it makes. We construct the following hybrid experiments:

- Hyb_0:

  1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$. Send $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ to $\mathcal{A}$.
  2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
     (a) $\mathcal{A}$ outputs $(i, \mathbf{y}_i \in \{0,1\}^l)$;
     (b) Sample $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.
     (c) Send $(\mathbf{s}_i^\intercal \mathbf{A} + \mathbf{x}_i^\intercal \bmod q, \mathbf{s}_i^\intercal \mathbf{C} + \mathbf{e}_i^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_i^\intercal \bmod q, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
  3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
     (a) $\mathcal{A}$ outputs $(i, j, \boldsymbol{\delta}_{i,j} \in \mathbb{Z}_q^m, \mathbf{y}_{i,j} \in \{0,1\}^l)$ satisfying $\|\boldsymbol{\delta}_{i,j}\| \le t$.
     (b) Sample $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.
     (c) Send $(\mathbf{s}_{i,j}^\intercal \mathbf{A} + \mathbf{x}_i^\intercal + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, \mathbf{s}_{i,j}^\intercal \mathbf{C} + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_{i,j}^\intercal \bmod q)$ to $\mathcal{A}$.
  4. $\mathcal{A}$ outputs a bit b.

- Hyb_1:

  1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$. Send $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ to $\mathcal{A}$.
  2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
     (a) $\mathcal{A}$ outputs $(i, \mathbf{y}_i \in \{0,1\}^l)$;
     (b) Sample $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, and $\mathbf{e}'_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma^*}^l$.
     (c) Send $(\mathbf{s}_i^\intercal \mathbf{A} + \mathbf{x}_i^\intercal \bmod q, \mathbf{s}_i^\intercal \mathbf{C} + (\mathbf{e}'_i)^\intercal + \mathbf{e}_i^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_i^\intercal \bmod q, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
  3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
     (a) $\mathcal{A}$ outputs $(i, j, \boldsymbol{\delta}_{i,j} \in \mathbb{Z}_q^m, \mathbf{y}_{i,j} \in \{0,1\}^l)$ satisfying $\|\boldsymbol{\delta}_{i,j}\| \le t$.
     (b) Sample $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.
     (c) Send $((\mathbf{s}_i + \mathbf{s}_{i,j})^\intercal \mathbf{A} + \mathbf{x}_i^\intercal + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, (\mathbf{s}_i + \mathbf{s}_{i,j})^\intercal \mathbf{C} + (\mathbf{e}'_i)^\intercal + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_{i,j}^\intercal \bmod q)$ to $\mathcal{A}$.
  4. $\mathcal{A}$ outputs a bit b.

- Hyb_{1+k} for $k = 1, 2, \ldots, N$:

  1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$. Send $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ to $\mathcal{A}$.
  2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
     (a) $\mathcal{A}$ outputs $(i, \mathbf{y}_i \in \{0,1\}^l)$;
     (b) Sample $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{e}'_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma^*}^l$, $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}'_i \xleftarrow{\$} \mathbb{Z}_q^l$.
     (c) If $1 \le i \le k$, send $(\mathbf{u}_i^\intercal, (\mathbf{u}'_i)^\intercal + \mathbf{e}_i^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_i^\intercal \bmod q, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
     (d) Otherwise, send $(\mathbf{s}_i^\intercal \mathbf{A} + \mathbf{x}_i^\intercal \bmod q, \mathbf{s}_i^\intercal \mathbf{C} + (\mathbf{e}'_i)^\intercal + \mathbf{e}_i^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_i^\intercal \bmod q, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
  3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
     (a) $\mathcal{A}$ outputs $(i, j, \boldsymbol{\delta}_{i,j} \in \mathbb{Z}_q^m, \mathbf{y}_{i,j} \in \{0,1\}^l)$ satisfying $\|\boldsymbol{\delta}_{i,j}\| \le t$.
     (b) Sample $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.

(c) If $1 \le i \le k$, send $(\mathbf{u}_i^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{A} + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, (\mathbf{u}_i')^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{C} + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_{i,j}^\intercal \bmod q)$ to $\mathcal{A}$.

(d) Otherwise, send $((\mathbf{s}_i + \mathbf{s}_{i,j})^\intercal \mathbf{A} + \mathbf{x}_i^\intercal + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, (\mathbf{s}_i + \mathbf{s}_{i,j})^\intercal \mathbf{C} + (\mathbf{e}_i')^\intercal + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot \mathbf{y}_{i,j}^\intercal \bmod q)$ to $\mathcal{A}$.

4. $\mathcal{A}$ outputs a bit b.

- Hyb_{N+2}:

1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$. Send $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ to $\mathcal{A}$.
2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
   (a) $\mathcal{A}$ outputs $(i, \mathbf{y}_i \in \{0,1\}^l)$;
   (b) Sample $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}_i' \xleftarrow{\$} \mathbb{Z}_q^l$.
   (c) Send $(\mathbf{u}_i^\intercal, (\mathbf{u}_i')^\intercal, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
   (a) $\mathcal{A}$ outputs $(i, j, \boldsymbol{\delta}_{i,j} \in \mathbb{Z}_q^m, \mathbf{y}_{i,j} \in \{0,1\}^l)$ satisfying $\|\boldsymbol{\delta}_{i,j}\| \le t$.
   (b) Sample $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.
   (c) Send $(\mathbf{u}_i^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{A} + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, (\mathbf{u}_i')^\intercal - \mathbf{e}_i^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{C} + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot (\mathbf{y}_{i,j} - \mathbf{y}_i)^\intercal \bmod q)$ to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b.

- Hyb_{N+3}:

1. Sample $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$. Send $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ to $\mathcal{A}$.
2. $\mathcal{A}$ may adaptively make queries of the following form (phase-1 queries):
   (a) $\mathcal{A}$ outputs $(i, \mathbf{y}_i \in \{0,1\}^l)$;
   (b) Sample $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}_i' \xleftarrow{\$} \mathbb{Z}_q^l$.
   (c) Send $(\mathbf{u}_i^\intercal, (\mathbf{u}_i')^\intercal, \mathsf{aux}(\mathbf{x}_i))$ to $\mathcal{A}$.
3. $\mathcal{A}$ may adaptively make queries of the following form (phase-2 queries):
   (a) $\mathcal{A}$ outputs $(i, j, \boldsymbol{\delta}_{i,j} \in \mathbb{Z}_q^m, \mathbf{y}_{i,j} \in \{0,1\}^l)$ satisfying $\|\boldsymbol{\delta}_{i,j}\| \le t$.
   (b) Sample $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$.
   (c) Send $(\mathbf{u}_i^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{A} + \boldsymbol{\delta}_{i,j}^\intercal \bmod q, (\mathbf{u}_i')^\intercal - \mathbf{e}_i^\intercal + \mathbf{s}_{i,j}^\intercal \mathbf{C} + \mathbf{e}_{i,j}^\intercal + \lfloor q/2 \rfloor \cdot (\mathbf{y}_{i,j} \oplus \mathbf{y}_i)^\intercal \bmod q)$ to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b.

We prove the following claims on the hybrids above.

(1) $\mathsf{Hyb}_0 \approx_{st} \mathsf{Hyb}_1$: Note that $\mathbf{e}_i' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma^*}^l$ and $\mathbf{e}_i, \mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$. Since $\sigma'/\sigma^* \ge \mathsf{super\text{-}poly}(\lambda)$, by a standard error flooding argument we have $(\mathbf{e}_i', \mathbf{e}_i' + \mathbf{e}_i, \mathbf{e}_i' + \mathbf{e}_{i,j}) \approx_{st} (\mathbf{e}_i', \mathbf{e}_i, \mathbf{e}_{i,j})$. Also, since $\mathbf{s}_{i,j}$ is uniform, $(\mathbf{s}_i \bmod q, \mathbf{s}_{i,j} \bmod q) \cong (\mathbf{s}_i \bmod q, \mathbf{s}_i + \mathbf{s}_{i,j} \bmod q)$. Thus, $\mathsf{Hyb}_0 \approx_{st} \mathsf{Hyb}_1$.

(2) $\mathsf{Hyb}_k \approx_c \mathsf{Hyb}_{k+1}$ for $k = 1, 2, \ldots, N$: from Hyb_k to Hyb_{k+1}, we switch $\mathbf{s}_k^\intercal \mathbf{A} + \mathbf{x}_k \bmod q$ to $\mathbf{u}_k^\intercal$ and $\mathbf{s}_k^\intercal \mathbf{C} + (\mathbf{e}')^\intercal \bmod q$ to $(\mathbf{u}_k')^\intercal$, in the presence of $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ and $\mathsf{aux}(\mathbf{x}_k)$. Thus, it is sufficient to

show

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}_k^\mathsf{T}\mathbf{A} + \mathbf{x}_k^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{x}_k), \mathbf{C}, \mathbf{s}_k^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T} \bmod q) \\ \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}_k^\mathsf{T}, \mathsf{aux}(\mathbf{x}_k), \mathbf{C}, (\mathbf{u}_k')^\mathsf{T}),$$

which is exactly the result given in Theorem 4.3.

(3) $\mathsf{Hyb}_{N+1} \cong \mathsf{Hyb}_{N+2}$: It is apparent that these two sampling methods are equivalent since $\mathbf{u}_i'$ is uniformly random.

(4) $\mathsf{Hyb}_{N+2} \approx_{st} \mathsf{Hyb}_{N+3}$: For $y_1, y_2 \in \{0,1\}$, $\lfloor q/2 \rfloor \cdot (y_1 - y_2) \bmod q$ and $\lfloor q/2 \rfloor \cdot (y_1 \oplus y_2) \bmod q$ differ by at most 1. Therefore, since $\sigma' > \mathsf{super\text{-}poly}(\lambda)$, we have $\mathbf{e}_{i,j}^\mathsf{T} + \lfloor q/2 \rfloor \cdot (\mathbf{y}_{i,j} - \mathbf{y}_i)^\mathsf{T} \bmod q \approx_{st} \mathbf{e}_{i,j}^\mathsf{T} + \lfloor q/2 \rfloor \cdot (\mathbf{y}_{i,j} \oplus \mathbf{y}_i)^\mathsf{T} \bmod q$.

Therefore, we conclude that $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_{N+3}$, where Hyb_0 and Hyb_{N+3} corresponds to the real world and the ideal world in Fig. 3, respectively. We can construct a stateful simulator Sim that works as follows:

- For phase-1 queries, on input $\mathsf{pk} := (\mathbf{A}, \mathbf{T}, \mathbf{C})$ and i, Sim samples $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$, and $\mathbf{u}_i' \xleftarrow{\$} \mathbb{Z}_q^l$ and outputs $\widetilde{\mathsf{ct}}_i := (\mathbf{u}_i^\mathsf{T}, (\mathbf{u}_i')^\mathsf{T})$.
- For phase-2 queries, on input $\mathsf{pk} := (\mathbf{A}, \mathbf{T}, \mathbf{C})$, i, j, $\boldsymbol{\delta}_{i,j}$, and $\Delta\mathbf{y}_{i,j} := \mathbf{y}_{i,j} \oplus \mathbf{y}_i$, Sim samples $\mathbf{s}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^l$, and outputs

$$\widetilde{\mathsf{ct}}_{i,j} := (\mathbf{u}_i^\mathsf{T} + \mathbf{s}_{i,j}^\mathsf{T}\mathbf{A} + \boldsymbol{\delta}_{i,j}^\mathsf{T} \bmod q, (\mathbf{u}_i')^\mathsf{T} - \mathbf{e}_i^\mathsf{T} + \mathbf{s}_{i,j}^\mathsf{T}\mathbf{C} + \mathbf{e}_{i,j}^\mathsf{T} + \lfloor q/2 \rfloor \cdot \Delta\mathbf{y}_{i,j}^\mathsf{T} \bmod q).$$

It is straightforward to see that Sim indeed satisfies the requirements in Definition 6.3. □

Finally, we show that Construction 6.1 satisfies strong fuzzy second-preimage resistance.

Theorem 6.3. *Let $\sigma \geq \tau^{c-1} \cdot \omega(\sqrt{\log \lambda})$, $B > \sigma\left(\sqrt{m} + \omega(\sqrt{\log \lambda})\right)$, and $\mathcal{X}$ be the renormalized distribution of $\mathcal{D}_{\mathbb{Z},\sigma}^m$ restricted to X. Suppose $\gamma = \omega(\sqrt{\log \lambda})$ and $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some n' satisfying $n' \log q \leq n - \omega(\log \lambda)$; then Construction 6.1 satisfies strong fuzzy second-preimage resistance defined in Definition 6.4 for input distribution $\mathcal{X}$.*

Proof. Similar to the proof for Theorem 6.2, we directly replace the input distribution $\mathcal{X}$ with $\mathcal{D}_{\mathbb{Z},\sigma}^m$ due to the statistical indistinguishability. Let $\mathcal{A}$ be an adversary for the strong second-preimage resistance experiment in Fig. 4 that wins with probability $\varepsilon \geq \mathsf{non\text{-}negl}(\lambda)$ and $N = \mathsf{poly}(\lambda)$ be the number of its queries. We construct an adversary $\mathcal{A}'$ that on input $(\mathbf{A}, \mathbf{T}, \mathbf{b}^\mathsf{T}, \mathbf{C}, (\mathbf{b}')^\mathsf{T})$ tries to search for $\mathbf{s}$:

1. Sample $k \xleftarrow{\$} \{1, 2, ..., N\}$.
2. Invoke $\mathcal{A}$ on $\mathsf{pk} := (\mathbf{A}, \mathbf{T}, \mathbf{C})$.
3. For $\mathcal{A}$'s ith query $\mathbf{y}_i$:
   (a) If $i = k$, send $\mathsf{ct}_i := (\mathbf{b}^\mathsf{T}, (\mathbf{b}')^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}_i \bmod q)$ to $\mathcal{A}$.
   (b) Otherwise, sample $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{e}_i' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^l$, and send $\mathsf{ct}_i := (\mathbf{s}_i^\mathsf{T}\mathbf{A} + \mathbf{x}_i^\mathsf{T} \bmod q, \mathbf{s}_i^\mathsf{T}\mathbf{C} + (\mathbf{e}_i')^\mathsf{T} + \lfloor q/2 \rfloor \cdot \mathbf{y}_i \bmod q)$ to $\mathcal{A}$.

4. Let $(k', \widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ be the output of $\mathcal{A}$. If $k' \neq k$, output $\perp$ and abort.
5. Run $\mathsf{TdInvert}(\mathbf{A}, \mathbf{T}, t, \mathbf{b} - \widetilde{\mathbf{x}} \bmod q)$; if it outputs $\perp$, output $\perp$ and abort, otherwise let its output be $(\hat{\mathbf{s}}, \hat{\mathbf{e}})$.
6. Output $\hat{\mathbf{s}}$.

Suppose $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$,$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma'}^l$, and set $\mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T}$ and $(\mathbf{b}')^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{C} + (\mathbf{e}')^\mathsf{T}$. In Theorem 6.1 we have proven the two-sided correctness property defined in Definition 6.2 of this scheme. Then with overwhelming probability, $(\mathbf{A}, \mathbf{T}, \mathbf{C})$ forms a pk under which the two-sided correctness requirement in Eq. (9) holds. It is easy to see that $\mathcal{A}'$ simulates the view of $\mathcal{A}$ in the experiment in a statistically close way. When $k = k'$ and $\mathcal{A}$ succeeds in the experiment, we have $\|\widetilde{\mathbf{x}} - \mathbf{e}\| \leq t$ by definition, in which case $\mathcal{A}'$ will be able to find the correct $\hat{\mathbf{s}} = \mathbf{s}$ through $\mathsf{TdInvert}$. The above event happens with probability $\varepsilon/N \geq \mathsf{non\text{-}negl}(\lambda)$, which is contradictory to Theorem 4.1. □

A Proof sketch for Theorem 4.3

Here we sketch the proof ideas of Theorem 4.3, i.e., our hardness results on trapdoor LWE with error leakage with a possibly superpolynomial modulus q, only relying on the polynomial-time hardness of LWE. This route of proof works by adapting the sample-preserving reduction in [MM11] as well as the entropic LWE result in [BD20] to the settings with a short error leakage, and integrating them into the proof of Theorem 4.1 with similar trapdoor dual-mode switching techniques used in Theorem 4.2. We omit the full formal proof because its technical ideas are similar to the existing proofs.

Adapting [MM11] for standard LWE with error leakage First, by slightly augmenting the proof in [MM11], one can obtain the following sample-preserving search-to-decision reduction for LWE augmented with some auxiliary information on the error vector.

Lemma A.1. *Let $n, m = \mathsf{poly}(\lambda)$ be positive integers such that $m \geq n + \omega(\log n)$. Let aux be an arbitrary (possibly randomized) auxiliary information function. Let χ be an error distribution whose support is on $[-s, s] \cap \mathbb{Z}$. Let $q > s$ be any prime modulus.*

Assume there exists an efficient algorithm $\mathcal{A}$ that distinguishes with non-negligible advantage between the following two distributions:

- $(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \mathsf{aux}(\mathbf{e}))$
  where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, *and* $\mathbf{e} \leftarrow \chi^m$;
- $(\mathbf{A}, \mathbf{u}^\mathsf{T}, \mathsf{aux}(\mathbf{e}))$
  where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, *and* $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$.

Then there exists an efficient algorithm $\mathcal{A}'$ such that

$$\Pr\left[\begin{array}{l|l} \mathcal{A}'(\mathbf{A}, \mathbf{b}, \mathsf{aux}(\mathbf{e})) & \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m} \\ = \mathbf{s} & \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m \\ & \mathbf{b}^\mathsf{T} := \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q \end{array}\right] \geq \mathsf{non\text{-}negl}(\lambda).$$

If auxiliary information $\mathsf{aux}(\mathbf{e})$ has output bit length bounded by $O(\log \lambda)$, then $\mathsf{aux}(\mathbf{e})$ does not help solve search LWE as the adversary can enumerate over all possible values of $\mathsf{aux}(\mathbf{e})$ and try for each. Thus, for such aux, decision LWE will also remain hard even given $\mathsf{aux}(\mathbf{e})$, assuming the hardness of standard LWE. Moreover, for any efficiently computable (samplable) randomized $\mathsf{aux}(\mathbf{e})$ with $|\mathsf{aux}(\mathbf{e})| \leq O(\log \lambda)$, giving out its internal random coins should also not hurt the search hardness. Thus, by considering the above reduction w.r.t. $\mathsf{aux}'(\mathbf{e}) := (\theta, \mathsf{aux}(\mathbf{e}; \theta))$, we will furthermore have

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \theta, \mathsf{aux}(\mathbf{e}; \theta)) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}, \theta, \mathsf{aux}(\mathbf{e}; \theta)).$$

This will be useful in the proof of Theorem 4.3 below, where the leakage also depends on randomness that the reduction samples by itself and uses to construct the instance given to the adversary.

Adapting [BD20] for entropic LWE with error leakage Directly plugging the results above into the proof of Theorem 3.3 and augmenting the reduction with the auxiliary information $\mathsf{aux}(\mathbf{e})$, it is not hard to obtain the following version of *entropic LWE* hardness result in the presence of error leakage. The results above apply to the discrete Gaussian error $\mathbf{e}$ here: since σ is polynomially bounded, by Lemma 3.5 we can restrict $\mathbf{e}$ to a polynomially bounded support at the cost of a negligible statistical distance.

Lemma A.2 (Entropic LWE with short error leakage)**.** *Let q be a prime modulus, $n, n', m = \mathsf{poly}(\lambda)$ be positive integers with $m \geq n + \omega(\log \lambda)$. Let $\mathcal{S}$ be an efficiently samplable distribution on $\mathbb{Z}_q^n$ s.t.*

$$\widetilde{H}_\infty(\mathbf{s}|\mathbf{s} + \mathbf{e}_1) \geq n' \log q + \omega(\log \lambda),$$

where $\mathbf{s} \leftarrow \mathcal{S}$ and $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^n$ for some Gaussian parameter $\sigma_1 = \omega(\sqrt{\log \lambda})$. Let $\sigma > C\sqrt{m}\gamma\sigma_1$ be a polynomially bounded Gaussian parameter, where C is some global constant. Let aux be any efficiently computable (possibly randomized) function with output length up to $O(\log \lambda)$ bits. Assume $\mathsf{dLWE}_{n',q,\mathcal{D}_{\mathbb{Z},\gamma}}$ is hard for some $\gamma = \omega(\sqrt{\log \lambda})$, then

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \theta, \mathsf{aux}(\mathbf{e}; \theta)) \approx_c (\mathbf{A}, \mathbf{u}^\mathsf{T}, \theta, \mathsf{aux}(\mathbf{e}; \theta)),$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathcal{S}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, and $\theta \xleftarrow{\$} \{0,1\}^*$ is the random coins for aux. As in Theorem 3.3, the same holds in the presence of auxiliary information sampled jointly with $\mathbf{s}$.

LWE with Trapdoor Hints with error leakage for superpolynomial q As the final step, we plug in Lemma A.2 into the proof recipe for Theorem 4.1 and apply trapdoor simulation/extension techniques to obtain Theorem 4.3.

Proof sketch for Theorem 4.3. We follow the proof of Theorem 4.1. We also discard the extra LWE samples $(\mathbf{C}, \mathbf{s}^\mathsf{T}\mathbf{C} + \mathbf{e}^\mathsf{T})$ for simplicity, as it does not affect the proof idea. So our goal is to show for any efficient (possibly randomized) function h with $|h(\cdot)| \leq O(\log \lambda)$,

$$(\mathbf{A}, \mathbf{T}, \mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q, \theta, h(\mathbf{e}; \theta)) \approx_c (\mathbf{A}, \mathbf{T}, \mathbf{u}^\mathsf{T}, \theta, h(\mathbf{e}; \theta)), \tag{10}$$

where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{HTdSamp}(1^n, 1^m, q, r)$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, and $\theta \xleftarrow{\$} \{0,1\}^*$.

Starting from the left-hand side of Eq. (10), we consider using [MP12]'s trapdoor sampling and simulating $\mathbf{T}$ with $\mathbf{R}$:

$$([\mathbf{B}\|\mathbf{BR}+\mathbf{G}], \mathbf{T}, [\mathbf{s}^\mathsf{T}\mathbf{B}+\mathbf{e}_1^\mathsf{T} \bmod q \| \mathbf{s}^\mathsf{T}\mathbf{BR}+\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T} \bmod q], \theta, h(\mathbf{e}_1, \mathbf{e}_2; \theta)).$$

where $\mathbf{T}$ is generated using $\mathbf{R}$. We use error flooding arguments to smudge the errors as $(\mathbf{e}_1, \mathbf{e}_2) \approx_{st} (\mathbf{e}_3 + \mathbf{e}_1, \mathbf{R}^\mathsf{T}\mathbf{e}_3 + \mathbf{e}_2)$, where $\mathbf{e}_3$ is a discrete Gaussian error vector with width parameter $\sigma_3 := \gamma\sqrt{m} \cdot \omega(\sqrt{\log \lambda}) = \mathsf{poly}(\lambda)$ thus $\mathbf{e}_3$ and $\mathbf{R}^\mathsf{T}\mathbf{e}_3$ are all overwhelmed by the superpolynomially wide errors $\mathbf{e}_1$ and $\mathbf{e}_2$. Thus, the polynomially-bounded error $\mathbf{e}_3$ satisfies the requirement on the error width in Lemma A.2. By the same arguments except with substituting the standard entropic LWE theorem (Theorem 3.3) with Lemma A.2, we can obtain the following indistinguishability result[10]:

$$\begin{aligned}&(\mathbf{B}, \mathbf{s}^\mathsf{T}\mathbf{B}+\mathbf{e}_3^\mathsf{T}, \mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T}, \theta', h'_{\mathbf{e}_2}(\mathbf{e}_3; \theta'))\\ \approx_c\ &(\mathbf{B}, \mathbf{u}^\mathsf{T}, \mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T}, \theta', h'_{\mathbf{e}_2}(\mathbf{e}_3; \theta'))\end{aligned}$$

where $\theta' := (\theta, \mathbf{R}, \mathbf{e}_1)$ and $h'_{\mathbf{e}_2}(\mathbf{e}_3; \theta') := h(\mathbf{e}_3 + \mathbf{e}_1, \mathbf{R}^\mathsf{T}\mathbf{e}_3 + \mathbf{e}_2; \theta)$. This will imply

$$\begin{aligned}&([\mathbf{B}\|\mathbf{BR}+\mathbf{G}], \mathbf{T}, [\mathbf{s}^\mathsf{T}\mathbf{B}+\mathbf{e}_1^\mathsf{T} \bmod q \| \mathbf{s}^\mathsf{T}\mathbf{BR}+\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T} \bmod q], \theta, h(\mathbf{e}_1, \mathbf{e}_2; \theta))\\ \approx_{st}&([\mathbf{B}\|\mathbf{BR}+\mathbf{G}], \mathbf{T}, [\mathbf{s}^\mathsf{T}\mathbf{B}+\mathbf{e}_3^\mathsf{T}+\mathbf{e}_1^\mathsf{T} \bmod q \| \mathbf{s}^\mathsf{T}\mathbf{BR}+\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_3^\mathsf{T}\mathbf{R}+\mathbf{e}_2^\mathsf{T} \bmod q],\\ &\theta, h(\mathbf{e}_3 + \mathbf{e}_1, \mathbf{R}^\mathsf{T}\mathbf{e}_3 + \mathbf{e}_2; \theta))\\ \approx_c&([\mathbf{B}\|\mathbf{BR}+\mathbf{G}], \mathbf{T}, [\mathbf{u}^\mathsf{T}+\mathbf{e}_1^\mathsf{T} \bmod q \| \mathbf{u}^\mathsf{T}\mathbf{R}+\mathbf{s}^\mathsf{T}\mathbf{G}+\mathbf{e}_2^\mathsf{T} \bmod q], \theta, h(\mathbf{e}_3 + \mathbf{e}_1, \mathbf{R}^\mathsf{T}\mathbf{e}_3 + \mathbf{e}_2; \theta)).\end{aligned}$$

The next key step is to switch $\mathbf{B}$ to be sampled with a trapdoor $\mathbf{T_B}$, and use $\mathbf{T_B}$ to simulate the trapdoor for $\mathbf{A}$ using the trapdoor extension algorithm. Now, the only term related to $\mathbf{R}$ besides $\mathbf{u}^\mathsf{T}\mathbf{R}$ is $\mathbf{BR}$. Recall that $\mathbf{BR} \in \mathbb{Z}_q^{n \times w}$ and $\mathbf{R} \in \{0,1\}^{m' \times w}$ where $w := nc$ and $m' := m - nc > 3n \log q$. By a simple calculation, conditioned on $\mathbf{BR}$, $\mathbf{R}$ still has at least $m'w - nw \log q \geq 2nw \log q$ entropy. This is much more than the entropy needed to apply left over hashing on $\mathbf{u}^\mathsf{T}\mathbf{R}$, which will let us statistically switch $\mathbf{u}^\mathsf{T}\mathbf{R}$ to an independent uniform $(\mathbf{u}')^\mathsf{T}$ and finally obtain the right hand side of Eq. (10). □

[10] A possible subtlety here is that now $h'_{\mathbf{e}_2}(\mathbf{e}_3)$ also needs to be conditioned on when considering the conditional distribution of $\mathbf{s}$, but since $|h'(\cdot)|$ is bounded by $O(\log \lambda)$, it barely affects the entropy bounds.

References

[AA16] Jacob Alperin-Sheriff and Daniel Apon. Dimension-preserving reductions from LWE to LWR. Cryptology ePrint Archive, Report 2016/589, 2016. 10

[ACEK17] Daniel Apon, Chongwon Cho, Karim Eldefrawy, and Jonathan Katz. Efficient, reusable fuzzy extractors from lwe. In *Cyber Security Cryptography and Machine Learning*, volume 10332 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017. 8, 10, 16, 41, 42

[ACF+22] Daniel Apon, Chloé Cachet, Benjamin Fuller, Peter Hall, and Feng-Hao Liu. Non-malleable digital lockers and robust fuzzy extractors in the plain model. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 353–383. Springer, Cham, December 2022. 10

[ACL+22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 102–132. Springer, Cham, August 2022. 9

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, Berlin, Heidelberg, August 2009. 24

[ALP26] Martin R. Albrecht, Russell W. F. Lai, and Eamonn W. Postlethwaite. Hardness of hinted ISIS from the space-time hardness of lattice problems. Cryptology ePrint Archive, Paper 2026/187, 2026. 9

[AMR25] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Key-homomorphic computations for RAM: Fully succinct randomised encodings and more. In Kalai and Kamara [KK25], pages 236–268. 2, 9

[AMYY25] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Zeroizing attacks against evasive and circular evasive LWE. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025: 23rd Theory of Cryptography Conference, Part II*, volume 16269 of *Lecture Notes in Computer Science*, pages 259–290. Springer, Cham, December 2025. 2, 9

[AP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 334–352. Springer, Berlin, Heidelberg, May 2012. 10

[BA13] Marina Blanton and Mehrdad Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE Transactions on Information Forensics and Security*, 8(9):1433–1445, 2013. 6

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, December 1993. 19

[BCFL23] David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Chainable functional commitments for unbounded-depth circuits. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023: 21st Theory of Cryptography Conference, Part III*, volume 14371 of *Lecture Notes in Computer Science*, pages 363–393. Springer, Cham, November / December 2023. 9

[BCLO09] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 224–241. Springer, Berlin, Heidelberg, April 2009. 6

[BCO11] Alexandra Boldyreva, Nathan Chenette, and Adam O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In Rogaway [Rog11], pages 578–595. 6

[BD20] Zvika Brakerski and Nico Döttling. Hardness of LWE on general entropic distributions. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 551–575. Springer, Cham, May 2020. 14, 28, 39, 50, 51

[BDHV07] Ileana Buhan, Jeroen Doumen, Pieter H. Hartel, and Raymond N. J. Veldhuis. Fuzzy extractors for continuous distributions. In Feng Bao and Steven Miller, editors, *ASIACCS 07: 2nd ACM Symposium on Information, Computer and Communications Security*, pages 353–355. ACM Press, March 2007. 5

[BF11] Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16. Springer, Berlin, Heidelberg, March 2011. 2, 9

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer, Berlin, Heidelberg, May 2014. 2

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Berlin, Heidelberg, August 2001. 4

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology –*

ASIACRYPT 2015, Part I, volume 9452 of *Lecture Notes in Computer Science*, pages 470–491. Springer, Berlin, Heidelberg, November / December 2015. 11

[BJTW25] Katharina Boudgoust, Corentin Jeudy, Erkan Tairi, and Weiqiang Wen. Hardness of M-LWE with general distributions and applications to leaky variants. Cryptology ePrint Archive, Report 2025/1472, 2025. 10

[BKM+18] Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 731–752. Springer, Cham, August 2018. 4

[BLMZ19] James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New techniques for obfuscating conjunctions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 636–666. Springer, Cham, May 2019. 4

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584. ACM Press, June 2013. 10, 23, 24, 27

[BLR+15] Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 563–594. Springer, Berlin, Heidelberg, April 2015. 6, 11

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004: 11th Conference on Computer and Communications Security*, pages 82–91. ACM Press, October 2004. 4, 6, 8, 10, 16, 41

[BS16] Mihir Bellare and Igors Stepanovs. Point-function obfuscation: A framework and generic constructions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 565–594. Springer, Berlin, Heidelberg, January 2016. 4

[BÜW24] Chris Brzuska, Akin Ünal, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part IV*, volume 15487 of *Lecture Notes in Computer Science*, pages 418–449. Springer, Singapore, December 2024. 2, 9

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106. IEEE Computer Society Press, October 2011. 2

[BVWW16] Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In Madhu Sudan, editor, *ITCS 2016: 7th Conference on Innovations in Theoretical Computer Science*, pages 147–156. Association for Computing Machinery, January 2016. 4

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, Berlin, Heidelberg, February 2007. 9, 11

[BW19] Ward Beullens and Hoeteck Wee. Obfuscating simple functionalities from knowledge assumptions. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 254–283. Springer, Cham, April 2019. 4

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski, Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, Berlin, Heidelberg, August 1997. 4

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In Smart [Sma08], pages 489–508. 4

[CD18] Dario Catalano and Roberto De Prisco, editors. *SCN 18: 11th International Conference on Security in Communication Networks*, volume 11035 of *Lecture Notes in Computer Science*. Springer, Cham, September 2018. 59

[CFP+16] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam D. Smith. Reusable fuzzy extractors for low-entropy distributions. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 117–146. Springer, Berlin, Heidelberg, May 2016. 6, 8, 10, 16, 41

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, Berlin, Heidelberg, May / June 2010. 17, 27

[CHRF24] Chloé Cachet, Ariel Hamlin, Maryam Rezapour, and Benjamin Fuller. Upgrading fuzzy extractors. In Christina Pöpper and Lejla Batina, editors, *ACNS 2024: 22nd International Conference on Applied Cryptography and Network Security, Part I*, volume 14583 of *Lecture Notes in Computer Science*, pages 156–182. Springer, Cham, March 2024. 6, 10

[CHW25] Jeffrey Champion, Yao-Ching Hsieh, and David J. Wu. Registered ABE and adaptively-secure broadcast encryption from succinct LWE. In Kalai and Kamara [KK25], pages 3–34. 2

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In Handschuh and Lysyanskaya [HL23], pages 72–105. 9

[DFR21] Luke Demarest, Benjamin Fuller, and Alexander Russell. Code offset in the exponent. In Stefano Tessaro, editor, *ITC 2021: 2nd Conference on Information-Theoretic Cryptography*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:23. Schloss Dagstuhl, July 2021. 10

[DGK+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, Berlin, Heidelberg, February 2010. 17, 18

[DJM+25] Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, and Vinod Vaikuntanathan. Simple and general counterexamples for private-coin evasive LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part VII*, volume 16006 of *Lecture Notes in Computer Science*, pages 73–92. Springer, Cham, August 2025. 2, 9

[DKL+23] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In Hazay and Stam [HS23], pages 417–446. 10

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, Berlin, Heidelberg, August 2006. 10

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. 6, 15, 29

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, Berlin, Heidelberg, May 2004. 4, 6, 10

[DS05] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In Gabow and Fagin [GF05], pages 654–663. 6

[FF20] Peter Fenteany and Benjamin Fuller. Same point composable and nonmalleable obfuscated point functions. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 2020: 18th International Conference on Applied Cryptography and Network Security, Part II*, volume 12147 of *Lecture Notes in Computer Science*, pages 124–144. Springer, Cham, October 2020. 6, 16, 41

[FLV23] Ben Fisch, Zeyu Liu, and Psi Vesely. Orbweaver: Succinct linear functional commitments from lattices. In Handschuh and Lysyanskaya [HL23], pages 106–131. 9

[FMR13] Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. Computational fuzzy extractors. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 174–193. Springer, Berlin, Heidelberg, December 2013. 10, 15

[FP19] Benjamin Fuller and Lowen Peng. Continuous-source fuzzy extractors: Source uncertainty and insecurity. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2952–2956. IEEE, 2019. 5

[FRS16] Benjamin Fuller, Leonid Reyzin, and Adam D. Smith. When are fuzzy extractors possible? In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 277–306. Springer, Berlin, Heidelberg, December 2016. 10

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Mitzenmacher [Mit09], pages 169–178. 2

[GF05] Harold N. Gabow and Ronald Fagin, editors. *37th Annual ACM Symposium on Theory of Computing*. ACM Press, May 2005. 57, 61

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *ICS 2010: 1st Innovations in Computer Science*, pages 230–240. Tsinghua University Press, January 2010. 14

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Umans [Uma17], pages 612–621. 2, 4, 10

[GMPW20] Nicholas Genise, Daniele Micciancio, Chris Peikert, and Michael Walter. Improved discrete gaussian and subgaussian analysis for lattice cryptography. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 623–651. Springer, Cham, May 2020. 14, 22

[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 Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008. 24, 27

[GRD19] Babak Poorebrahim Gilkalaye, Ajita Rattani, and Reza Derakhshani. Euclidean-distance based fuzzy commitment scheme for biometric template security. In *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2019. 5

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances*

in Cryptology – CRYPTO 2015, Part II, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523. Springer, Berlin, Heidelberg, August 2015. 2, 11

[GZ19] Steven D. Galbraith and Lukas Zobernig. Obfuscated fuzzy hamming distance and conjunctions from subset product problems. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 81–110. Springer, Cham, December 2019. 5, 10

[HJL25] Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. Lattice-based post-quantum iO from circular security with random opening assumption (part II: zeroizing attacks against private-coin evasive LWE assumptions). Cryptology ePrint Archive, Report 2025/390, 2025. 2, 9

[HL23] Helena Handschuh and Anna Lysyanskaya, editors. *Advances in Cryptology – CRYPTO 2023, Part II*, volume 14082 of *Lecture Notes in Computer Science*. Springer, Cham, August 2023. 57, 58

[HLP11] Shai Halevi, Yehuda Lindell, and Benny Pinkas. Secure computation on the web: Computing without simultaneous interaction. In Rogaway [Rog11], pages 132–150. 7

[HS23] Carmit Hazay and Martijn Stam, editors. *Advances in Cryptology – EUROCRYPT 2023, Part III*, volume 14006 of *Lecture Notes in Computer Science*. Springer, Cham, April 2023. 57, 62

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd Annual ACM Symposium on Theory of Computing*, pages 60–73. ACM Press, June 2021. 4

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, Cham, May / June 2022. 4

[JP18] Marc Joye and Alain Passelègue. Function-revealing encryption - definitions and constructions. In Catalano and De Prisco [CD18], pages 527–543. 11

[KK25] Yael Tauman Kalai and Seny F. Kamara, editors. *Advances in Cryptology – CRYPTO 2025, Part III*, volume 16002 of *Lecture Notes in Computer Science*. Springer, Cham, August 2025. 53, 56

[KLM+18] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Catalano and De Prisco [CD18], pages 544–562. 11

[KLSS23] Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 549–580. Springer, Cham, August 2023. 10

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Smart [Sma08], pages 146–162. 9, 11

[LKK+18] Joohee Lee, Dongwoo Kim, Duhyeong Kim, Yongsoo Song, Junbum Shin, and Jung Hee Cheon. Instant privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive, Report 2018/1214, 2018. 10

[LPSS14] San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 315–334. Springer, Berlin, Heidelberg, August 2014. 2, 9

[LSW25] Russell W. F. Lai, Monisha Swarnakar, and Ivy K. Y. Woo. Leaky LWE: Learning with errors with semi-adaptive secret- and error-leakage. *IACR Communications in Cryptology*, 2(3), 2025. 10

[LW16] Kevin Lewi and David J. Wu. Order-revealing encryption: New constructions, applications, and lower bounds. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1167–1178. ACM Press, October 2016. 6

[Mit09] Michael Mitzenmacher, editor. *41st Annual ACM Symposium on Theory of Computing*. ACM Press, May / June 2009. 58, 60

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Rogaway [Rog11], pages 465–484. 39, 50

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [PJ12], pages 700–718. 3, 11, 12, 17, 21, 25, 26, 27, 28, 39, 52

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381. IEEE Computer Society Press, October 2004. 13, 22

[OPW11] Adam O'Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. In Rogaway [Rog11], pages 525–542. 10

[Pei07] Chris Peikert. Limits on the hardness of lattice problems in ell _p norms. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 333–346. IEEE Computer Society, 2007. 22

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher [Mit09], pages 333–342. 24

[Pei10] Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97. Springer, Berlin, Heidelberg, August 2010. 14, 22, 24

[PJ12] David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, April 2012. 60, 61

[PR12] Omkant Pandey and Yannis Rouselakis. Property preserving symmetric encryption. In Pointcheval and Johansson [PJ12], pages 375–391. 6, 11

[PXS25] Mallika Prabhakar, Louise Xu, and Prateek Saxena. Model inversion attacks meet cryptographic fuzzy extractors, 2025. 5

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Gabow and Fagin [GF05], pages 84–93. 2, 24, 27

[Rog11] Phillip Rogaway, editor. *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, August 2011. 54, 59, 60

[Sma08] Nigel P. Smart, editor. *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, April 2008. 56, 60

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Berlin, Heidelberg, March 2009. 9, 11

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 535–559. Springer, Cham, August 2022. 2, 9

[Uma17] Chris Umans, editor. *58th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, October 2017. 58, 62

[Ver11] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices, 2011. 28

[Ver26] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2 edition, 2026. 21

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 195–221. Springer, Cham, December 2022. 2, 9

[Wee05] Hoeteck Wee. On obfuscating point functions. In Gabow and Fagin [GF05], pages 523–532. 4

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 217–241. Springer, Cham, May / June 2022. 2, 9

[Wee24] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part III*, volume 14922 of *Lecture Notes in Computer Science*, pages 178–209. Springer, Cham, August 2024. 2, 9

[Wee25] Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part III*, volume 15603 of *Lecture Notes in Computer Science*, pages 34–62. Springer, Cham, May 2025. 2

[WL18] Yunhua Wen and Shengli Liu. Robustly reusable fuzzy extractor from standard assumptions. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 459–489. Springer, Cham, December 2018. 10

[WLH18] Yunhua Wen, Shengli Liu, and Shuai Han. Reusable fuzzy extractor from the decisional Diffie-Hellman assumption. *Designs, Codes and Cryptography*, 86(11):2495–2512, 2018. 10

[WW23a] Hoeteck Wee and David J. Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part V*, volume 14442 of *Lecture Notes in Computer Science*, pages 201–235. Springer, Singapore, December 2023. 9

[WW23b] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In Hazay and Stam [HS23], pages 385–416. 9

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Umans [Uma17], pages 600–611. 2, 4, 5, 6, 10, 40

[ZCY21] Kaiyi Zhang, Hongrui Cui, and Yu Yu. Facial template protection via lattice-based fuzzy extractors. Cryptology ePrint Archive, Report 2021/1559, 2021. 5