Tight Lattice-Based Signatures without Trapdoors from Search LWE

Rutchathon Chairattana-Apirom[1], Nico Döttling[2], Julian Loss[3], Stefano Tessaro[1], and Benedikt Wagner[4]

[1] Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
rchairat@cs.washington.edu, tessaro@cs.washington.edu

[2] CISPA Helmholtz Center for Information Security
doettling@cispa.de

[3] Ruhr University Bochum
lossjulian@gmail.com

[4] Ethereum Foundation
benedikt.wagner@ethereum.org

Abstract. The study of digital signatures with tight reductions has attracted considerable attention over the past two decades, as such schemes inherit essentially the same quantitative hardness as the underlying computational problem. In the context of lattice-based cryptography, the GPV approach (Gentry, Peikert, and Vaikuntanathan, STOC '08) admits a simple tight reduction from the SIS problem, but relies on preimage sampling via trapdoors, which often leads to complex and brittle implementations. By contrast, tight proofs for trapdoor-free constructions, following the Fiat–Shamir paradigm, necessarily rely on decisional assumptions, most notably the decisional LWE assumption. From the perspective of concrete security, however, reliance on a search assumption is preferable, as known search-to-decision reductions for LWE provide only weak quantitative guarantees.

This paper presents the first efficient lattice-based signature scheme with a tight reduction to a search assumption (namely, the hardness of the search LWE problem) that does not require any trapdoor in the scheme itself (but merely uses, instead, a trapdoor in the proof). Our construction follows the Fiat-Shamir paradigm and can be viewed as a lattice analogue of Chevallier–Mames signatures (CRYPTO '05). Establishing security in the lattice setting, however, requires overcoming significant technical obstacles: in particular, our proof develops several new techniques to cope with the inherently weak soundness guarantees of lattice-based interactive proofs.

Keywords. Lattice-based cryptography; digital signatures; tightness

1 Introduction

With the potential arrival of large-scale quantum computers on the horizon, we are currently undergoing a shift toward cryptographic mechanisms secure against quantum adversaries. The most prominent approach to achieve post-quantum security relies on *lattice-based cryptography*. Notably, two of the three signatures selected by NIST for post-quantum standardization rely on lattices [LDK+22, PFH+22]. In this work, we specifically study lattice-based digital signatures from the perspective of *tight security*. Concretely, we will propose a new efficient construction for which tight security is proved under the assumed hardness of the search LWE problem, in the random oracle model (ROM) [BR93]. Before turning to our contribution, we discuss the broader context in more detail.

©IACR 2026. This is the full version of an article that will be published in the proceedings of CRYPTO 2026.

Tight Security for Lattice-based Signatures. A security proof for a cryptographic scheme (e.g., a signature scheme) proceeds via a *reduction* to a hard problem Π. Given an adversary that forges a signature with probability ε in time t, we construct an algorithm that solves Π with probability ε' in time t'. Necessarily, since the reduction runs the original adversary at least once, we have $t' \geqslant t$. Typically, we have $\varepsilon' \leqslant \varepsilon$. The larger the gap between t and t' (and between ε and ε'), the worse the quality of our reduction—if the reduction is loose, while it can still guarantee that the security of the signature is implied by the hardness of Π *qualitatively*, it cannot rule out that it may be much easier to break the scheme than to solve Π. When $\varepsilon' \approx \varepsilon$ and $t' \approx t$, we call the security reduction *tight*. Intuitively, such a tight reduction shows that the scheme is at least as secure as the underlying problem Π, *without any degradation*.

Tightness has been the focus of an entire sub-area of cryptography, with several of these results applying specifically to signatures [KW03, HJ12, AFLT12, BKKP15, BJLS16, BL16, KMP16, DGJL21, PW22, CAHT26]. For the case of efficient lattice-based signatures in the ROM, two avenues have been shown to yield tight reductions. The first is to follow the GPV approach [GPV08], a hash-and-sign signature based on preimage sampling, for which the original security proof gives a tight reduction, in the ROM, to the hardness of the Short Integer Solution (SIS) problem (or its module variant, MSIS). However, the original security proof that underlies the alternative approach to build lattice-based signatures in the ROM via the Fiat-Shamir transform [Lyu09, Lyu12] is not as amenable to a tight reduction to SIS. It requires rewinding and forking, in the same spirit of the security proof of Schnorr signatures [PS00] assuming the hardness of the discrete logarithm problem, for which tight reductions are ruled out by several impossibility results [PV05, GBL08, Seu12]. Tight proofs for Fiat-Shamir lattice-based signatures are instead obtained by going through the intermediate notion of a *lossy identification* scheme [AFLT12, KLS18, ABB+17]. In contrast to GPV, whose security relies on a search problem, this is an approach that inherently relies on *decisional* assumptions, in particular the hardness of the decisional version of the Learning with Errors (dLWE) problem.

Decision vs. Search Assumptions. Many consider proving security from a search assumption a qualitatively stronger guarantee, as the hardness of a search problem is often viewed as a weaker assumption than the hardness of its decisional counterpart. Still, one may wonder whether this matters at all for LWE, due to the existence of *search-to-decision reductions* [Reg05, MM11, LPR10, LS15] showing that the hardness of *search* LWE (sLWE) implies the hardness of dLWE. However, these reductions, while *asymptotically* meaningful, only provide weaker *concrete* guarantees.[1] In practice, concrete parameter choices for schemes based on dLWE are typically guided primarily by direct cryptanalysis rather than by quantitatively instantiating these reductions. For example, consider that the running time t' of the constructed sLWE adversary grows *polynomially* in $1/\varepsilon$, where ε is the advantage of the original dLWE adversary.

This hence raises the following question, which is at the core of our paper:

> *Can we design a Fiat-Shamir-like signature scheme with a tight security reduction to a lattice-based search assumption in the ROM?*

We stress that the study of such constructions as an alternative to GPV is very worthwhile—indeed, NIST standardized both a GPV-like construction, Falcon [PFH+22], and a Fiat-Shamir style construction, Dilithium [LDK+22], as they each have their own merits. A challenge with GPV-like hash-then-sign constructions is that the sampling of preimages with help of a trapdoor is challenging from an implementation standpoint (Falcon, for instance, requires floating-point arithmetic). One should point out that computationally more lightweight hash-then-sign signatures have been developed [YJW23, EEN+24], but these inherently rely on the use of a super-polynomial modulus to simplify preimage sampling with smudging techniques, and their security proofs rely on dLWE.

Our Contributions. This paper affirmatively answers the above question by providing the first lattice-based Fiat-Shamir signature with a tight reduction to a search assumption, specifically the search variant of *module* LWE. Our scheme is proved in particular to achieve strong unforgeability, and our security proof works with polynomial modulus.

[1] The same applies to direct reductions from worst-case lattice problems to dLWE [PRS17, RSW18].

Our scheme can be thought of as applying the Fiat-Shamir transform to a *five*-move identification scheme. As we explain in our technical overview below, our signature scheme can be thought of as a lattice analogue of the Chevallier-Mames (CM) [Che05, KLP17] signature, which enjoys a tight reduction to CDH (this scheme can also be thought of as applying the Fiat-Shamir transform to a five-move identification scheme, as abstracted by [KLP17]). However, obtaining such a signature in the lattice regime requires overcoming several barriers and the introduction of new proof techniques. Indeed, at its core, the tight security proof of CM signatures relies on the soundness of a discrete-log equality proof. However, in the lattice setting, it is well known that the analogue proof that two vectors $\mathbf{t}, \bar{\mathbf{t}}$ have the format of $\mathbf{t} = \mathbf{s}\mathbf{A} + \mathbf{e}$ and $\bar{\mathbf{t}} = \mathbf{s}\mathbf{H} + \bar{\mathbf{e}}$ (for known matrices $\mathbf{A}, \mathbf{H}$, a secret vector $\mathbf{s}$, and short noise vectors $\mathbf{e}, \bar{\mathbf{e}}$) offers much weaker soundness guarantees, in that successfully convincing a verifier does not imply $\bar{\mathbf{t}}$ is of the required format, even if $\mathbf{t}$ is. Yet our security proof will need to be able, given $\mathbf{t}, \bar{\mathbf{t}}$ and a valid proof with such weak soundness guarantee, to extract $\mathbf{s}$ with the help of a trapdoor for $\mathbf{H}$, and *crucially*, to do so in a way that (roughly) preserves running time and success probability of the original adversary producing a forgery. (We stress here that our use of trapdoors is confined to the security *proof*.)

Overcoming the above issue requires the development of a new probabilistic framework that may be of independent interest and lead to new tight reductions. The technical overview below will give a detailed explanation of the key ideas. We see this as our main technical contribution.

What we do not do (or: future work). It is important to stress a few questions that remain open and that we do not address in our paper. First off, we do not optimize for concrete parameter choices. We do give asymptotic parameter estimates, in particular showing that we can instantiate our scheme with polynomial modulus. However, more care is needed to make our analysis concrete and give us smaller parameters.

Moreover, our techniques do not extend to the quantum random oracle model (QROM) [BDF+11], and obtaining a tight proof from a search assumption for a Fiat-Shamir-like signature in the QROM remains an important open question. For GPV, Boneh et al. [BDF+11] already give a tight QROM proof relying on SIS, although under the assumption that the reduction can use a sufficiently secure PRF. Katsumata et al. [KYY18] then give a better reduction that relies on dLWE, and not search LWE.

Another direction in which tightness has been considered is the *multi-user setting*, where one shows reductions that do not degrade with the number of users in the system. Here, tight lattice-based constructions have been considered, both in the setting without corruptions [ZLPG19] and with corruptions [PW22]. All these results are in the ROM and assume dLWE. The problem was also studied in the standard model [HLWG23], again using decisional assumptions. Extending our results to the multi-user setting is an important avenue for future work that we do not consider in this paper. We note here that even without lattices, only considering the group setting, the current state of the art relying on search assumptions [HOS25] relies on online-extractable proofs [Fis05], resulting in a construction with high cost.

We also note that we stick with ROM solutions here. Still, we note that tightly secure schemes in the standard model were further proposed by Boyen and Li [BL16] and Lai, Liu, Wang [LLW20], and also require dLWE.

Finally, our research is limited to efficient Fiat-Shamir signatures in the ROM. There are certainly other avenues towards tightness. For instance, disregarding efficiency, one could build in principle a tight signature based on SIS and an online-extractable NIZK [Fis05], but this is not our focus.

1.1 Technical Overview

Starting point: tight signatures from CDH. As stated above, our starting point is the signature scheme by Chevallier-Mames (CM) [Che05, KLP17] in prime-order group settings, which features a tight security reduction from the CDH problem. In this scheme, the keys have the form $\mathsf{pk} = X = g^x \in \mathbb{G}$ and $\mathsf{sk} = x \in \mathbb{Z}_p$. Signing a message μ proceeds by (1) computing $R = g^r$ for $r \leftarrow\!\!\$\, \mathbb{Z}_p$, (2) hash $h \leftarrow \mathsf{H}_1(R, \mu)$, (3) compute $Z = h^x, S = h^r$, (4) hash $c \leftarrow \mathsf{H}_2(R, h, Z, S, \mu)$, and (5) output $\sigma = (R, Z, S, z = r + c \cdot x)$. Verification checks that for $h = \mathsf{H}_1(R, \mu)$ and $c = \mathsf{H}_2(R, h, Z, S, \mu)$ it holds that $R \cdot X^c = g^z$ and $S \cdot Y^c = h^z$. Observe that the elements (R, S, c, z) are simply a transcript of a Σ-protocol for DLOG equality, i.e. we prove that X and Z have the same discrete logarithm with respect to the bases g and h.

In the security proof, the hash functions H_1 and H_2 are modeled as random oracles. Unforgeability of this scheme can be established in two steps [KLP17]: (a) we can simulate the signing oracles without knowledge of the secret key x by programming H_1 and H_2, and (b) we can embed a rerandomized component of a CDH instance into H_1 and extract the corresponding CDH solution from the forgery.

In a bit more detail, in Step (a) one proceeds by simulating (R, S, c, z) using an honest-verifier zero-knowledge (HVZK) simulator for the DLOG equality proof and sets $Z = X^a$ while programming $\mathsf{H}_1(R, \mu) = g^a$ and $\mathsf{H}_2(R, h, Z, S, \mu) = c$ for random $a, c \leftarrow\!\!\$\, \mathbb{Z}_p$.

In Step (b), observe that by the statistical soundness of the DLOG equality proof a forgery $(\mu^*, (R^*, Z^*, S^*, z^*))$ must satisfy $Z^* = \mathsf{H}_1(R^*, \mu^*)^x$. Hence, for a CDH instance (g, X, Y) the reduction sets $\mathsf{pk} = X$ and programs H_1 to output Y^t for freshly sampled $t \leftarrow\!\!\$\, \mathbb{Z}_p$ for each query. Thus, if $\mathsf{H}_1(R^*, \mu^*) = Y^t$ and $(\mu^*, (R^*, Z^*, S^*, z^*))$ is a valid forgery then $(Z^*)^{1/t} = Y^x$ is the desired CDH solution.

Attempt 1: A lattice analogue of CM. Consider the following *translation* of the CM scheme into the (Module-)LWE setting. The public key $\mathsf{pk} = (\mathbf{A}, \mathbf{t})$ consists of an LWE matrix $\mathbf{A}$ and a vector $\mathbf{t} = \mathbf{sA} + \mathbf{e}$, and the secret key $\mathsf{sk} = (\mathbf{s}, \mathbf{e})$ consists of a uniformly random LWE secret $\mathbf{s}$ and a (discrete Gaussian) error term $\mathbf{e}$.[2] In analogy to the CM scheme, the matrix $\mathbf{A}$ will take the role of the generator $g \in \mathbb{G}$, whereas the LWE secret $\mathbf{s}$ takes the role of the secret $x \in \mathbb{Z}_p$, whereas $\mathbf{t}$ takes the role of the public key $X \in \mathbb{G}$. Signing a message μ proceeds as follows:

- Compute $\mathbf{u} \leftarrow \mathbf{rA} + \mathbf{d}$ with $\mathbf{r}$ sampled uniformly and an error term $\mathbf{d}$.
- Hash $\mathbf{H} \leftarrow \mathsf{H}_1(\mathbf{u}, \mu)$ where $\mathbf{H}$ is a uniformly random matrix with the same number of rows as $\mathbf{A}$.
- Compute $\overline{\mathbf{t}} \leftarrow \mathbf{sH} + \overline{\mathbf{e}}$, i.e., an LWE sample with respect to the matrix $\mathbf{H}$ with error term $\overline{\mathbf{e}}$.
- Compute $\overline{\mathbf{u}} \leftarrow \mathbf{rH} + \overline{\mathbf{d}}$ with error term $\overline{\mathbf{d}}$.
- Hash $c \leftarrow \mathsf{H}_2(\mathbf{u}, \mathbf{H}, \overline{\mathbf{t}}, \overline{\mathbf{u}}, \mu)$ where c comes from a large challenge space with short elements.
- Return $(\mathbf{u}, \overline{\mathbf{t}}, \overline{\mathbf{u}}, \mathbf{z} = \mathbf{r} + c\mathbf{s})$

The verification recomputes $\mathbf{H}$ and c, and checks if the following vectors

$$\mathbf{u} + c \cdot \mathbf{t} - \mathbf{zA} \quad \text{and} \quad \overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}} - \mathbf{zH} \text{ are short.} \tag{1}$$

Conceptually, this signature scheme is built around (the attempt of) a lattice analogue of a DLOG-equality proof system. Specifically, the underlying Σ-protocol tries to establish that $(\mathbf{A}, \mathbf{t})$ and $(\mathbf{H}, \overline{\mathbf{t}})$ are related by the same LWE secret $\mathbf{s}$, i.e. $\mathbf{t} \approx \mathbf{sA}$ and $\overline{\mathbf{t}} \approx \mathbf{sH}$. If this proof system was indeed sound, we would obtain a simple and tight reduction to search LWE, i.e. given $\mathbf{A}$ and $\mathbf{t} = \mathbf{sA} + \mathbf{e}$ we want to find $\mathbf{s}$. Staying within the two-step reduction framework laid out for the CM scheme above, we can achieve Step (a), that is simulating signatures without knowledge of the secret key by relying on *rejection sampling* [Lyu09, Lyu12]. More precisely, a routine modification of the protocol above, which *rejects* transcripts with "bad" c, enables us to simulate $(\mathbf{u}, \overline{\mathbf{u}}, c, \mathbf{z})$ without knowledge of sk, whereas we obtain $(\mathbf{H}, \overline{\mathbf{t}})$ by rerandomizing $(\mathbf{A}, \mathbf{t})$ (e.g. [GMPW20]).

Now, in Step (b) our goal is to recover the LWE secret $\mathbf{s}$. And indeed, a valid forgery provided by the adversary includes $\overline{\mathbf{t}}$ and a (supposed) proof that $\mathbf{t} \approx \mathbf{sA}$ and $\overline{\mathbf{t}} \approx \mathbf{sH}$. So the obvious strategy here is to attempt to extract $\mathbf{s}$ from $\overline{\mathbf{t}} \approx \mathbf{sH}$, and since $\mathbf{H}$ is generated by H_1, our reduction can generate $\mathbf{H}$ together with a lattice trapdoor which allows it to efficiently *decode* $\mathbf{s}$ from $\overline{\mathbf{t}}$.

Indeed, the above protocol would achieve tight security from search LWE if not for one fundamental obstruction: The *simultaneous proximity* proof for $\mathbf{t} \approx \mathbf{sA}$ and $\overline{\mathbf{t}} \approx \mathbf{sH}$ naively adapted from the DLOG equivalence proof in CM *is not sound*.

Specifically, while we do get the guarantee that $\mathbf{t} = \mathbf{sA} + \mathbf{e}$ is well-formed as it is part of pk, meaning that $\mathbf{e}$ is short, the same cannot be said about $\overline{\mathbf{t}} \leftarrow \mathbf{sH} + \overline{\mathbf{e}}$ as it is adversarially chosen. Specifically, an adversary can find $\overline{\mathbf{e}}$ with the following properties: (1) $\overline{\mathbf{e}}$ itself is non-short, and (2) for a random short c the probability ε that $c \cdot \overline{\mathbf{e}}$ is short is inversely polynomial. Property (1) causes the trapdoor extraction mechanism described above to fail, while property (2) makes the resulting signature verify with probability

[2] We assume vectors are, by default, row vectors.

ε. In other words, an adversary might produce valid forgeries from which we cannot extract an LWE solution. While this discussion only scratches the surface, the problem is compounded when we consider that $\mathbf{u}$ and $\overline{\mathbf{u}}$ are also adversarially generated with potentially non-short error terms.

Thus, trying to extract a search LWE solution directly using the trapdoor of $\mathbf{H}$ seems doomed. However, observe that we can use the trapdoor of $\mathbf{H}$ for a different purpose. Note that as soon as c has been sampled, the adversary is *committed* to $\mathbf{z}$, that is $\mathbf{u} + c \cdot \mathbf{t}$ and $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}}$ statistically bind to $\mathbf{z}$.[3] Hence, there is at most one $\mathbf{z}$ such that $\mathbf{u} + c \cdot \mathbf{t} - \mathbf{z}\mathbf{A}$ and $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}} - \mathbf{z}\mathbf{H}$ are both short. Consequently, a reduction can use the trapdoor for $\mathbf{H}$ to extract $\mathbf{z}$ from $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}}$ *as soon as c has been sampled*, i.e. even before the adversary sees the output of H_2, and importantly does not have to wait for the adversary to provide $\mathbf{z}$. Stating this differently, the reduction can instantly decide whether a transcript can lead to a forgery as soon as c has been sampled. This *shortcut* feature will be critical for our actual construction.

While we cannot prove tight security for our lattice-based CM analogue, we can in fact prove security under search LWE using standard forking ideas. However, we will not need to *fork all the way*: As we have just emphasized, for our specific scheme the reduction does not need to wait for the adversary to provide the response $\mathbf{z}$, but can extract $\mathbf{z}$ on its own from $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}}$ using the trapdoor for $\mathbf{H}$. Hence, consider the following alternative strategy to complete part (b) of the reduction: Every time the adversary queries H_2 with an input $\mathbf{u}, \mathbf{H}, \overline{\mathbf{t}}, \overline{\mathbf{u}}, \mu$, the reduction samples c and uses the trapdoor of $\mathbf{H}$ (recall that $\mathbf{H}$ has been generated by H_1) to extract $\mathbf{z}$ from $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}}$. If this extraction fails, then by the above discussion no $\mathbf{z}$ exists such that $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}} - \mathbf{z}\mathbf{H}$ and $\mathbf{u} + c \cdot \mathbf{t} - \mathbf{z}\mathbf{A}$ are short. Thus, the reduction can just output c as a response of H_2 to the adversary and continue the simulation, as it can be sure that this transcript cannot lead to a forgery. However, if on the other hand it holds for the extracted $\mathbf{z}$ that $\overline{\mathbf{u}} + c \cdot \overline{\mathbf{t}} - \mathbf{z}\mathbf{H}$ and $\mathbf{u} + c \cdot \mathbf{t} - \mathbf{z}\mathbf{A}$ are short then our reduction tries to sample a second c' with this property. I.e. it keeps resampling c' and tries to extract a $\mathbf{z}'$ from $\overline{\mathbf{u}} + c' \cdot \overline{\mathbf{t}}$ using the trapdoor of $\mathbf{H}$ until both $\overline{\mathbf{u}} + c' \cdot \overline{\mathbf{t}} - \mathbf{z}'\mathbf{H}$ and $\mathbf{u} + c' \cdot \mathbf{t} - \mathbf{z}'\mathbf{A}$ are short. Once we have found such c, c' and $\mathbf{z}, \mathbf{z}'$ we are done: Note that since

$$\mathbf{u} + c \cdot \mathbf{t} - \mathbf{z}\mathbf{A} = \mathbf{f} \,, \qquad \text{and} \qquad \mathbf{u} + c' \cdot \mathbf{t} - \mathbf{z}'\mathbf{A} = \mathbf{f}' \,,$$

for short $\mathbf{f}$ and $\mathbf{f}'$, subtracting the two equations and rearranging yields

$$(c - c') \cdot \mathbf{t} = (\mathbf{z} - \mathbf{z}')\mathbf{A} + \mathbf{f} - \mathbf{f}'.$$

As it holds $\mathbf{t} = \mathbf{s}\mathbf{A} + \mathbf{e}$ by assumption, we obtain

$$((c - c')\mathbf{s} - \mathbf{z} + \mathbf{z}')\mathbf{A} = \mathbf{f} - \mathbf{f}' - (c - c')\mathbf{e}.$$

Again, since the right-hand side is short and there are no short vectors in the span of $\mathbf{A}$, we can conclude that both sides must be 0. Hence, as the right-hand side is short it also holds that $\mathbf{f} - \mathbf{f}' - (c - c')\mathbf{e} = 0$ over the ring of integers $\mathcal{R}$. Therefore, we can efficiently recover the error $\mathbf{e}$ by computing $\mathbf{e} = \frac{\mathbf{f} - \mathbf{f}'}{c - c'}$ in the algebraic number field $\mathcal{K}$.[4] Once we have $\mathbf{e}$, the reduction can just solve the linear equation system $\mathbf{s} \cdot \mathbf{A} = \mathbf{t} - \mathbf{e}$ for $\mathbf{s}$ and output $\mathbf{s}$.

The astute reader has observed what makes this reduction non-tight: Once we have found a challenge c that gives rise to an accepting transcript, we need to resample until we have found a second accepting c'. If, for a given transcript the probability that c gives rise to an accepting transcript is (say) $\varepsilon = 1/\mathsf{poly}(\lambda)$, then we need $1/\varepsilon = \mathsf{poly}(\lambda)$ attempts to find a valid c'. A very similar argument was used, for instance, in recent work [JTWZ26] proposing the first two-round adaptively secure threshold signature from decision LWE.

We do not know how to prove tight security for this protocol. However, we will now provide a simple modification of this protocol which only leads to a factor 2 increase in signature size and runtime, but allows us to prove tight security from search LWE.

[3] This follows from basic properties of random lattices, specifically there are no short vectors in the row-spans of $\mathbf{A}$ and $\mathbf{H}$, except with negligible probability.

[4] *Not* in the modular ring $\mathcal{R}_q$!

Attempt 2: Two-fold parallel repetition to the rescue. Our modification is conceptually simple: In essence, we run two sessions of the above protocol in parallel. The two sessions, denoted session 0 and session 1, share the same public key $(\mathbf{A}, \mathbf{t})$ as well as $(\mathbf{H}, \bar{\mathbf{t}})$ and have joint random oracle queries, with the twist that the query to H_2 outputs two *independent* challenges (c_0, c_1), one for each session. Looking ahead, this modification will allow us to *polarize* the adversary's advantage and sample a pair of accepting challenges (c_b, c_b') for one of the two sessions (i.e. for a $b \in \{0,1\}$) with the same probability as sampling a pair (c_0, c_1) of accepting challenges for both sessions in the actual protocol. As discussed above such a pair (c_b, c_b') then enables us to efficiently extract an LWE solution.

The modified protocol proceeds as follows. The signer first computes $\mathbf{u}_0, \mathbf{u}_1$ which are hashed along with μ to $\mathbf{H} \leftarrow \mathsf{H}_1(\mathbf{u}_0, \mathbf{u}_1, \mu)$. Then, it computes $\bar{\mathbf{t}}$ as before but computes two $\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1$. As explained above, the challenges provided by $\mathsf{H}_2((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$ are now c_0, c_1. Finally, the signer computes the two responses $\mathbf{z}_0, \mathbf{z}_1$ and outputs the signature containing $(\bar{\mathbf{t}}, (\mathbf{u}_b, \bar{\mathbf{u}}_b, \mathbf{z}_b)_{b\in\{0,1\}})$. The verification checks that for both $b \in \{0,1\}$, $\mathbf{u}_b + c_b \cdot \mathbf{t} - \mathbf{z}_b\mathbf{A}$ and $\bar{\mathbf{u}}_b + c_b \cdot \bar{\mathbf{t}} - \mathbf{z}_b\mathbf{H}$ are short. For the purpose of this overview, we omitted several details and refer to Section 4 for the full description of the signing algorithm.

To build some intuition in what we gain from this modification, let us consider an adversary that makes just a single query to H_2. Consequently, the entire advantage of this adversary is concentrated in this single query. Recall our discussion above that the trapdoor for the matrix $\mathbf{H}$ allows us to decide if the adversary wins just from the query $((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$ and the challenges c_0, c_1. In particular, we do not need to wait for the adversary to provide us with a forgery and extract the responses $\mathbf{z}_0$ and $\mathbf{z}_1$ from $\bar{\mathbf{u}}_0 + c_0 \cdot \bar{\mathbf{t}}$ and $\bar{\mathbf{u}}_1 + c_1 \cdot \bar{\mathbf{t}}$ using the trapdoor of $\mathbf{H}$.

Now fix a query $((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$ and let $P_b(c_b)$ be the event that there exists a $\mathbf{z}_b$ such that both $\mathbf{u}_b + c_b \cdot \mathbf{t} - \mathbf{z}_b\mathbf{A}$ and $\bar{\mathbf{u}}_b + c_b \cdot \bar{\mathbf{t}} - \mathbf{z}_b\mathbf{H}$ are short. Notice that *both* $P_0(c_0)$ and $P_1(c_1)$ must hold for the transcript to be accepting. Let ε_b be the probability over the choice of c_b that the event $P_b(c_b)$ occurs. Now recall that c_0 and c_1 are sampled *independently*, hence the probability that the event $A(c_0, c_1) = P_0(c_0) \wedge P_1(c_1)$ holds, that is both $P_0(c_0)$ and $P_1(c_1)$ hold, is $\varepsilon_0 \cdot \varepsilon_1$.

Recall that the reduction can compute $A(c_0, c_1)$ on its own, without the adversary ever noticing. Hence, the reduction could just as well evaluate a different predicate on different challenges! Let c_0, c_0', c_1, c_1' be four independently sampled challenges and define the predicate $B(c_0, c_0', c_1, c_1') = (P_0(c_0) \wedge P_0(c_0')) \vee (P_1(c_1) \wedge P_1(c_1'))$. Then, if $B(c_0, c_0', c_1, c_1')$ holds, then either both c_0 and c_0' provide accepting transcripts for session 0 or both c_1 and c_1' provide accepting transcripts for session 1. Since c_0, c_0', c_1, c_1' are sampled independently, the inclusion-exclusion principle yields that the probability (over the choice of c_0, c_0', c_1, c_1') that $B(c_0, c_0', c_1, c_1')$ holds is

$$\varepsilon_0^2 + \varepsilon_1^2 - \varepsilon_0^2\varepsilon_1^2 = \underbrace{(\varepsilon_0 - \varepsilon_1)^2}_{\geqslant 0} + \underbrace{\varepsilon_0\varepsilon_1}_{\geqslant 0}\underbrace{(1 - \varepsilon_0\varepsilon_1)}_{\geqslant 0} + \varepsilon_0\varepsilon_1 \geqslant \varepsilon_0\varepsilon_1.$$

In other words, the probability that we sample two accepting challenges (c_b, c_b') for one of the two sessions is greater than or equal to the adversary's probability of producing a forgery, which is $\varepsilon_0 \cdot \varepsilon_1$. In this sense, the predicate B polarizes the adversary's advantage into one of the two sessions.

What remains to show is how we can lift this polarization technique to the case of general multi-query adversaries without incurring additional losses.

Extension to multi-query adversaries. In the following, we will need a somewhat more precise terminology. Note that in the discussion above the predicates P_0 and P_1 implicitly depend on the adversary's query $q = ((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$ to H_2. As there are many queries now, we will make q an explicit input to both P_0 and P_1. Consequently, we will also make q an input to the predicate A which determines the local winning condition and the predicate B which determines the local extraction condition. That is, we define

$$\begin{aligned} A(c_0, c_1, q) &= P_0(c_0, q) \wedge P_1(c_1, q)\ , \\ B(c_0, c_0', c_1, c_1', q) &= (P_0(c_0, q) \wedge P_0(c_0', q)) \vee (P_1(c_1, q) \wedge P_1(c_1', q))\ . \end{aligned}$$

Moreover, we will refer to (c_0, c_1) as an *accepting pair* for a query q if $A(c_0, c_1, q) = 1$. Before we discuss our extraction strategy against multi-query adversaries, consider the following simplification. Instead of considering the real experiment where the adversary obtains the output $(c_0, c_1) = \mathsf{H}_2(q)$ irrespective of

whether (c_0, c_1) is an accepting pair for q, it suffices to consider a hybrid experiment which immediately aborts and declares a win once an accepting pair (c_0, c_1) for q is sampled. The important point here is that in this hybrid experiment the adversary *never* gets to see an accepting pair (c_0, c_1) produced by H_2, i.e. every single output of H_2 is a non-accepting pair for its corresponding query. Observe further that this hybrid modification only increases the winning probability of the adversary, as accepting pairs now trigger an instant win instead of letting the adversary decide whether it wants to cash in this win. Hence, for a query q we will call $\varepsilon_q = \mathsf{Pr}_{c_0,c_1}[A(c_0, c_1, q) = 1]$ the *local advantage* of the adversary.

Now, our high-level extraction strategy for the multi-query setting is fairly simple: Every time the adversary queries H_2 with a query q we will try to extract, and if extraction fails we will try to sample a non-accepting pair (c_0, c_1) for q. Hence, a naive approach of extracting a query could look as follows. Upon receiving a query q for H_2, proceed as follows.

1. Sample (c_0, c_0', c_1, c_1') and test if $B(c_0, c_0', c_1, c_1', q) = 1$, if so extract the LWE solution as described above.
2. If not, keep sampling (c_0, c_1) until $A(c_0, c_1, q) = 0$, i.e. until we have sampled a non-accepting pair (c_0, c_1) for q. Send (c_0, c_1) as output of H_2 to the adversary.

First observe that by the discussion in the last paragraph, for each query q the probability of finding an LWE solution is greater than or equal to the probability of sampling an accepting transcript (c_0, c_1) for q. So we are guaranteed that for each query individually our reduction actually *harnesses* the adversary's advantage. On the other hand, if we did not manage to extract an LWE solution, then the output (c_0, c_1) of $\mathsf{H}_2(q)$ is guaranteed to follow the correct distribution that the adversary would observe in the hybrid experiment. Consequently, from the adversary's view the hybrid experiment and the simulation by our reduction are identically distributed. However, there still is an issue with efficiency. Specifically, for a given query q the event $B(c_0, c_0', c_1, c_1', q) = 1$ might actually have very high probability $1-\varepsilon_q$, but we got unlucky and sampled (c_0, c_0', c_1, c_1') with $B(c_0, c_0', c_1, c_1', q) = 0$. Then step 2 of our simulation of H_2 might incur an excessively large number of sampling attempts $O(1/\varepsilon_q)$, which makes the *worst-case runtime* of our reduction once more non-tight. The cause of this issue is that our naive simulation of H_2 is highly unbalanced: We only attempt step 1 once, otherwise we repeat step 2 until an output is produced.

Concrete hardness assumptions are typically expressed in terms of worst-case running time, and Jaeger and Tessaro [JT20] showed that corresponding computational problems can become easier with respect to expected running time as a metric. For this reason, it is important to ensure tightness with respect to worst-case running time. To achieve this, we will resort to a *balanced strategy* to simulate H_2. Specifically, we proceed as follows. Repeat the following procedure until an output is produced: Flip an unbiased coin $b \leftarrow\!\!\$\ \{0, 1\}$.

- If $b = 0$ sample (c_0, c_0', c_1, c_1') and test if $B(c_0, c_0', c_1, c_1', q) = 1$, if so extract the LWE solution as described above and terminate. Otherwise, repeat the loop.
- If $b = 1$ sample (c_0, c_1) and test if $A(c_0, c_1, q) = 0$. If so send (c_0, c_1) as output of H_2 to the adversary. Otherwise, repeat the loop.

First notice that each iteration of this sampler terminates with probability $\geqslant 1/2$: Let $\mathsf{Pr}_{c_0,c_1}[A(c_0, c_1, q) = 1] = \varepsilon_q$. Our analysis in the last paragraph has established that $\mathsf{Pr}_{c_0,c_0',c_1,c_1'}[B(c_0, c_0', c_1, c_1', q) = 1] = \varepsilon_q' \geqslant \varepsilon_q$. Hence, for each round the probability the sampler terminates is

$$\begin{aligned}\frac{1}{2}\left(\mathsf{Pr}[B(c_0, c_0', c_1, c_1', q) = 1] + \mathsf{Pr}[A(c_0, c_1, q) = 0]\right) &= \frac{1}{2}(\varepsilon_q' + 1 - \varepsilon_q) \\ &\geqslant \frac{1}{2}(\varepsilon_q + 1 - \varepsilon_q) = \frac{1}{2}.\end{aligned}$$

That is, it terminates in 2 attempts in expectation. Furthermore, if we condition on a round of the sampler terminating, then by Bayes' law, the probability ε_q^* that this round produces an LWE solution is

$$\varepsilon_q^* = \frac{\varepsilon_q'}{1 + \varepsilon_q' - \varepsilon_q} = \frac{\varepsilon_q'\varepsilon_q}{\varepsilon_q + \underbrace{\varepsilon_q}_{\leqslant 1} \cdot \underbrace{(\varepsilon_q' - \varepsilon_q)}_{\geqslant 0}} \geqslant \frac{\varepsilon_q \cdot \varepsilon_q'}{\varepsilon_q + \varepsilon_q' - \varepsilon_q} = \varepsilon_q.$$

Consequently, the local advantage ε_q^* that we find an LWE solution from query q is still at least ε_q. That is, our reduction converts the local advantage ε_q of the adversary into local advantage ε_q^* against the search LWE problem.

From local to global advantage. As a final step, we need to show that the local advantage of our reduction translates into a global advantage which is greater then or equal to than the adversary's advantage in the unforgeability experiment.

Assume for a moment that it holds for every query q that $\varepsilon_q^* = \varepsilon_q$, i.e. the local advantage of the reduction is always identical to the local advantage of the adversary. Then, by the discussion in the last paragraph, the view of the adversary in the simulation of the reduction is identically distributed to the view of the adversary in the hybrid experiment, and furthermore the overall probability of the reduction finding an LWE solution is also identical to the adversary's winning probability in the hybrid experiment. Hence, all that remains to ensure is that a potential excess of local advantage of the reduction does not influence the global advantage of the reduction negatively. We achieve this via a monotone coupling argument: We can show that if, for every query q we artificially raise the local advantage of the adversary from ε_q to $\varepsilon_q^* \geqslant \varepsilon_q$ by giving the adversary an extra chance of winning, then the global advantage also increases. After this transformation we arrive at a hybrid experiment where all the local advantages of the adversary are identical to the local advantages of the reduction and the above argument applies. For details refer to Section 3.

Finally, observe that our reduction needs two sampling attempts in expectation to simulate the oracle H_2. By bookkeeping the number of resampling attempts and introducing an abort condition if too many attempts occurred, we can achieve strict polynomial runtime with an amortized number of less than 3 sampling attempts per query to H_2. By a careful application of a Hoeffding bound, we only pay for this with a negligible drop in success probability.

2 Preliminaries

Notation. We use κ as the security parameter. We denote $[n] = \{1, \ldots, n\}$ for any $n \in \mathbb{N}$. For an odd integer q, $a \in \mathbb{Z}_q$ is represented with an integer in the range $\{-\lfloor q/2 \rfloor, \ldots, \lfloor q/2 \rfloor\}$. We denote vectors (and matrices) using bold-faced (capital) letters (e.g., $\mathbf{v}$ and $\mathbf{H}$). We also use **row-vector** notation instead of column vectors (denoted with $\mathbf{v}^\top$). Denote $x \leftarrow a$ as assigning value a to a variable x, $a \leftarrow\!\!\$\ S$ as (uniformly) sampling a from a finite set or a distribution S. We write $y \leftarrow\!\!\$\ \mathsf{A}(x)$ as running a (probabilistic) algorithm A on input x with fresh randomness. We denote by Ber_p the Bernoulli distribution with parameter $p \in [0, 1]$. We denote the $N \times N$ identity matrix by $\mathbb{I}_N$ or $\mathbb{I}$ when the dimension is clear from the context.

2.1 Signature Schemes

A digital signature scheme SS consists of the following algorithms:

- The key generation algorithm $\mathsf{SS.KG}(1^\kappa)$ outputs the signing and verification keys $(\mathsf{sk}, \mathsf{pk})$.
- The randomized signing algorithm $\mathsf{SS.Sign}(\mathsf{sk}, \mu \in \{0,1\}^*)$ takes as inputs, the signing key sk and a message $\mu \in \{0,1\}^*$, and outputs a signature σ.
- The deterministic verification algorithm outputs a bit $\mathsf{SS.Ver}(\mathsf{pk}, m, \sigma)$.

We say that SS is correct with correctness error at most $\varepsilon = \varepsilon(\kappa)$ if for any $\mu \in \{0,1\}^*$, $\mathsf{Pr}[\mathsf{SS.Ver}(\mathsf{pk}, m, \sigma) = 0] \leqslant \varepsilon$ for $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{SS.KG}(1^\kappa), \sigma \leftarrow\!\!\$\ \mathsf{SS.Sign}(\mathsf{sk}, \mu \in \{0,1\}^*)$. For security, we consider *strong unforgeability under chosen message attack (SUF-CMA) and unforgeability under key-only attack (UF-KOA)* security, defined by the games SUF-CMA, UF-KOA (given in Figure 1). The difference between the two games is $\mathcal{A}$'s access to the signing oracle S. We denote the corresponding advantage of any adversary $\mathcal{A}$ as $\mathsf{Adv}^{\mathsf{game}}_{\mathsf{SS}}(\mathcal{A}, \kappa) := \mathsf{Pr}[\mathrm{GAME}^{\mathcal{A}}_{\mathsf{SS}}(\kappa) = 1]$ for $\mathsf{game} = \mathsf{sufcma}, \mathsf{ufkoa}$ and GAME = SUF-CMA, UF-KOA, respectively.

Game UF-KOA$^{\mathcal{A}}_{\mathsf{SS}}(\kappa)$: $\boxed{\text{SUF-CMA}^{\mathcal{A}}_{\mathsf{SS}}(\kappa)\text{:}}$	Oracle S(m):
$\mathsf{Sigs} \leftarrow \varnothing$; $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{SS.KG}(1^\kappa)$	$\sigma \leftarrow\!\!\$\ \mathsf{SS.Sign}(\mathsf{sk}, m)$
$(m^*, \sigma^*) \leftarrow\!\!\$\ \mathcal{A}^{\boxed{\mathrm{S}}}(\mathsf{pk})$	$\mathsf{Sigs} \leftarrow \mathsf{Sigs} \cup \{(m, \sigma)\}$
return $(m^*, \sigma^*) \notin \mathsf{Sigs} \ \wedge\ \mathsf{SS.Ver}(\mathsf{pk}, m^*, \sigma^*) = 1$	**return** σ

Fig. 1. Unforgeability under Key-Only Attack and Strong Unforgeability games.

2.2 Polynomial Rings

Let N be a power-of-two integer, $\mathcal{R} = \mathbb{Z}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$. For a (odd-prime) modulus q, we write $\mathbb{Z}_q$ as the ring of integers modulo q and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. We denote the cyclotomic field $\mathcal{K} := \mathbb{Q}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle \supset \mathcal{R}$ and $\mathcal{K}_\mathbb{R} := \mathbb{R} \otimes \mathcal{K} \cong \mathbb{R}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$. An element $a = \sum_{i=0}^{N-1} a_i \mathsf{X}^i \in \mathcal{K}_\mathbb{R}$ can be represented with its coefficient embedding $(a_i)_{i\in[0..N-1]} \in \mathbb{R}^N$ and this extends to vectors of $\mathcal{K}_\mathbb{R}$-elements. For $\mathbf{a} \in \mathbb{R}^n$, we denote its ℓ_p-norm as $\|\mathbf{a}\|_p := \left(\sum_{j=1}^n |a_j|^p\right)^{1/p}$. Accordingly, for $\mathbf{a} = (\sum_{i=0}^{N-1} a_{1,i}\mathsf{X}^i, \ldots, \sum_{i=0}^{N-1} a_{n,i}\mathsf{X}^i) \in K^n$, we write $\|\mathbf{a}\|_p := \left\|(a_{j,i})_{j\in[n],i\in[0..N-1]}\right\|_p$. We also recall the notation $\mathsf{Rot}(a)$ for the negacirculant matrix defined by $\sum_{i=0}^{N-1} a_i \mathsf{Rot}(\mathsf{X})$ with $\mathsf{Rot}(\mathsf{X}) = \begin{bmatrix} \mathbf{0} & 1 \\ \mathbb{I}_{N-1} & \mathbf{0}^\top \end{bmatrix} \in \mathbb{Z}^{N\times N}$. For a vector $\mathbf{e} \in \mathcal{R}^m$, we denote the block matrix $\mathsf{Rot}(\mathbf{e}) = (\mathsf{Rot}(e_i))_{i\in[m]} \in \mathbb{R}^{N\times mN}$. Recall also the matrix norm $\|\mathbf{B}\| = \sup_{\mathbf{v}\in\mathbb{R}^m:\|\mathbf{v}\|=1} \|\mathbf{B}\mathbf{v}^\top\|$ and that $\|\mathbf{B}^\top\mathbf{B}\| = \|\mathbf{B}\mathbf{B}^\top\| = \|\mathbf{B}\|^2$. We have that for $\mathbf{e} \in \mathcal{R}^n, \|\mathsf{Rot}(\mathbf{e})\| \leqslant \sqrt{N}\,\|\mathbf{e}\|$.

We recall the following results about the structure of $\mathcal{R}_q$ for specific primes q and the probability of a uniformly random $\mathbf{A}$ being primitive (i.e., $\mathbf{A}\mathcal{R}_q^m = \mathcal{R}_q^n$).

Lemma 2.1 (Corollary 1.2 of [LS18]). *Let $N \geqslant k > 1$ and $q \equiv 2k + 1 \pmod{4k}$ be a prime. Then, the polynomial $\mathsf{X}^N + 1$ factors as $\mathsf{X}^N + 1 \equiv \prod_{j=1}^k (\mathsf{X}^{N/k} + r_j) \pmod q$ for distinct $r_j \in \mathbb{Z}_q^*$ and each $(\mathsf{X}^{N/k} + r_j)$ is irreducible in $\mathbb{Z}_q[\mathsf{X}]$. Furthermore, any $a \in \mathcal{R}_q$ is invertible if $0 < \|a\|_\infty < \frac{1}{\sqrt{k}} q^{1/k}$ or $0 < \|a\| < q^{1/k}$.*

Lemma 2.2 (Lemma 2.6 of [BJRW23]). *Let q be an unramified prime such that the polynomial $\mathsf{X}^N + 1 \equiv \prod_{j=1}^k (\mathsf{X}^{N/k} + r_j) \pmod q$ for distinct $r_j \in \mathbb{Z}_q^*$ and each $(\mathsf{X}^{N/k} + r_j)$ are irreducible in $\mathbb{Z}_q[\mathsf{X}]$, and $n < m$. Then, $\mathsf{Pr}_{\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}}[\mathbf{A}\mathcal{R}_q^m = \mathcal{R}_q^n] \geqslant \prod_{i=0}^{n-1}(1 - q^{-N/k(m-i)})^k \geqslant 1 - nk \cdot q^{-N/k(m-n+1)}$.*

2.3 Lattices and Discrete Gaussian Distributions

LATTICES. A m-dimensional lattice $\Lambda \subseteq \mathbb{R}^m$ is a discrete additive subgroup of $\mathbb{R}^m$. A lattice is finitely generated by a set of linearly independent basis vectors $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^m$ for $n < m$ and $\Lambda(\mathbf{B}) = \{\sum x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$. We say that Λ is full-rank if $n = m$. The dual lattice of Λ is denoted as $\Lambda^* := \{\mathbf{x} \in \mathsf{Span}(\Lambda) : \forall \mathbf{y} \in \Lambda, \langle \mathbf{x}, \mathbf{y}\rangle \in \mathbb{Z}\}$. We can also consider lattices over $\mathcal{R}$ via the coefficient embeddings and $\Lambda \subseteq \mathcal{R}^m$ is a mN-dimensional lattice over $\mathbb{Z}$.

For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ (resp., $\mathcal{R}_q^{n\times m}$), denote $\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x}^\top = \mathbf{0}^\top \pmod q\}$ (for rings $\Lambda_q^\perp(\mathbf{A}) \subseteq \mathcal{R}^m$) and its dual lattice $(\Lambda_q^\perp(\mathbf{A}))^* = \frac{1}{q}\Lambda_q(\mathbf{A})$ where $\Lambda_q(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \exists \mathbf{y} \in \mathbb{Z}^n : \mathbf{x} = \mathbf{y}\mathbf{A} \pmod q\}$ (for rings $\Lambda_q(\mathbf{A}) \subseteq \mathcal{R}^m$). Both are full-rank lattices.

SYMMETRIC POSITIVE MATRICES. We say that a matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$ is positive (semi-)definite ($\mathbf{\Sigma} \succeq 0$) if for any $\mathbf{x} \in \mathbb{R}^m, \mathbf{x}\mathbf{\Sigma}\mathbf{x}^\top \geqslant 0$. We denote the largest and smallest singular value of a matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ as $s_1(\mathbf{B})$ and $s_m(\mathbf{B})$, respectively. For symmetric positive definite $\mathbf{\Sigma}, s_1(\mathbf{\Sigma}) = s_m(\mathbf{\Sigma}^{-1})^{-1}$ and $s_1(\mathbf{\Sigma}) = \|\mathbf{\Sigma}\|$.

DISCRETE GAUSSIANS AND SMOOTHING PARAMETERS. Let $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$ be a symmetric positive definite matrix and the discrete Gaussian distribution over $\Lambda \subseteq \mathbb{Z}^m$ centered at $\mathbf{c}$ with covariance $\mathbf{\Sigma}$, denoted $\mathcal{D}_{\Lambda,\mathbf{c},\mathbf{\Sigma}}$, is defined as

$$\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) := \exp\left(-\pi(\mathbf{x}-\mathbf{c})\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{c})^\top\right) \quad \text{and} \quad \mathcal{D}_{\Lambda,\mathbf{c},\sqrt{\mathbf{\Sigma}}}(\mathbf{x}) := \frac{\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x})}{\rho_{\sqrt{\mathbf{\Sigma}}}(\Lambda)}\ ,$$

where $\rho_{\sqrt{\mathbf{\Sigma}}}(\Lambda) := \sum_{\mathbf{x}\in\Lambda}\rho_{\sqrt{\mathbf{\Sigma}}}(\mathbf{x})$. We omit $\mathbf{c}$ when $\mathbf{c} = 0$. We also write $\mathcal{D}_{\Lambda,\mathbf{c},\sigma}$ if $\mathbf{\Sigma} = \sigma^2\mathbb{I}_m$ for $\sigma > 0$. The smoothing parameter $\eta_\varepsilon(\Lambda)$ of a lattice for $\varepsilon > 0$ is the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^*\backslash\{\mathbf{0}\}) \leqslant \varepsilon$. We will denote $\varepsilon = 2^{-2\kappa}$ for this paper. We then recall the following lemmata on discrete Gaussian tail-bound [AGHS13] and smoothing parameter bound for random matrices.

Lemma 2.3 (Lemma 3 of [AGHS13] adapted to rings). *For a full-rank lattice $\Lambda \subseteq \mathcal{R}^m$, $\mathbf{c} \in \mathcal{R}^m$, a constant $\varepsilon \in (0,1)$ and a symmetric positive definite $\mathbf{\Sigma}$ such that $\sqrt{s_m(\mathbf{\Sigma})} \geqslant \eta_\varepsilon(\Lambda)$, then*

$$\mathsf{Pr}_{\mathbf{x}\leftarrow\$\,\mathcal{D}_{\Lambda,\mathbf{c},\sqrt{\mathbf{\Sigma}}}}[\|\mathbf{x}-\mathbf{c}\| \geqslant \sqrt{s_1(\mathbf{\Sigma})mN}] \leqslant \frac{1+\varepsilon}{1-\varepsilon}2^{-mN}\ .$$

Lemma 2.4 (Lemma 2.6 of [GPV08]). *For any full-rank $\Lambda \subseteq \mathbb{R}^m$ and $\varepsilon > 0$, we have $\eta_\varepsilon(\Lambda) \leqslant \frac{\sqrt{\log(2m(1+1/\varepsilon))/\pi}}{\lambda_1^\infty(\Lambda^*)}$ where $\lambda_1^\infty(\Lambda^*)$ denotes the length of the shortest vector in ℓ_∞-norm in the dual lattice Λ^*.*

Lemma 2.5 (Lemma 2.5 of [BTT22]). *Let q be an odd prime and $\mathbf{A}$ a uniformly random matrix in $\mathcal{R}_q^{n\times m}$ with $m > n$. Then, except with probability at most 2^{-N} on the choice of $\mathbf{A}$, $\lambda_1^\infty(\Lambda_q(\mathbf{A})) \geqslant \frac{1}{8\sqrt{N}}q^{1-\frac{n}{m}}$.*

Lemmas 2.4 and 2.5 imply that $\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leqslant 8q^{\frac{n}{m}}\cdot\sqrt{N\ln(2mN(1+1/\varepsilon))/\pi}$ (for $\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}$) with overwhelming probability. Lemma 2.5 also implies that the probability of a random matrix $\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}$ being non-injective when mapping $(\mathbf{s},\mathbf{e})$ for small enough $\mathbf{e}$ to $\mathbf{sA}+\mathbf{e}$ is negligible. The proof follows similarly via a union bound argument as in [LS15, Lemma 5.1] and [BL26, Lemma 27].

Lemma 2.6. *Let q be a prime, $m > n$ and $\beta < \frac{1}{8\sqrt{N}}q^{1-\frac{n}{m}}$. Then,*

$$\mathsf{Pr}_{\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}}[\exists\mathbf{s}\neq\mathbf{0}\in\mathcal{R}_q^n,\mathbf{e}\in\mathcal{R}^m : \|\mathbf{e}\| \leqslant \beta \ \wedge\ \mathbf{sA}+\mathbf{e}=\mathbf{0}] \leqslant \frac{3(2\beta+1)^{mN}}{q^{(m-n)N/2}}\ .$$

The following regularity lemma allows us to argue uniformity of vectors $\mathbf{Ar}$ where $\mathbf{r}$ sampled from discrete Gaussian distribution with sufficiently large standard deviation. The standard deviation bound can be derived from the bounds obtained from the above, Lemma 2.2, and [GPV08, Corollary 2.8]. This resembles [LPR13, Corollary 2.8] which gives similar statement for a different matrix $\mathbf{A} = [\mathbb{I}_n \mid \overline{\mathbf{A}}]$ with $\overline{\mathbf{A}}\leftarrow\$\,\mathcal{R}_q^{n\times(m-n)}$.

Lemma 2.7 (Regularity). *Let $\mathcal{R}$ be a cyclotomic ring of degree N, $n < m, q \geqslant 2$ be integers. Let $\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}$. Then, with probability $1-2^{-N}$ over $\mathbf{A}$, the distribution of $\mathbf{Ar}^\top$, for $\mathbf{r}\leftarrow\$\,\mathcal{D}_{\mathcal{R}^m,\sigma_r}$ where $\sigma_r \geqslant \frac{8}{\sqrt{\pi}}q^{\frac{n}{m}}\sqrt{N\ln(2mN(1+1/\varepsilon))}$, is within statistical distance 2ε from the uniform distribution over $\mathcal{R}_q^n$.*

We also recall the following theorem for LWE self-reduction from [GMPW20] adapted to ring/module settings which we prove in Section A.

Lemma 2.8 (Theorem 5.3 of [GMPW20], adapted). *Let $\varepsilon > 0$, $\mathbf{A}\in\mathcal{R}_q^{n\times m}$ be primitive, let $\mathbf{t} = \mathbf{sA}+\mathbf{e}$ for some $\mathbf{s}\in\mathcal{R}_q^n$ and $\mathbf{e}\in\mathcal{R}^m$ and let $\sigma_x,\sigma_y > 0$ such that $\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leqslant (\sigma_x^{-2}+\sigma_y^{-2}\cdot\|\mathsf{Rot}(\mathbf{e})\|^2)^{-1/2} \leqslant \sigma_x$. Then, the following distributions are within $\frac{4\varepsilon}{1-\varepsilon}$ statistical distance, with $\overline{\mathbf{\Sigma}} = \sigma_x^2\mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^\top + \sigma_y^2\mathbb{I}_N \in \mathbb{Z}^{N\times N}$.*

1. $\{(\mathbf{a}^\top = \mathbf{Ax}^\top, b = \mathbf{tx}^\top + y) \mid \mathbf{x}\leftarrow\$\,\mathcal{D}_{\mathcal{R}^m,\sigma_x}, y\leftarrow\$\,\mathcal{D}_{\mathcal{R},\sigma_y}\}$
2. $\{(\mathbf{a}^\top\leftarrow\$\,\mathcal{R}_q^n, b = \mathbf{sa}^\top + e) \mid e\leftarrow\$\,\mathcal{D}_{\mathcal{R},\sqrt{\overline{\mathbf{\Sigma}}}}\}$.

Trapdoor generation and LWE inversion. We recall the gadget vector $\mathbf{g} = (1,\mathsf{b},\ldots,\mathsf{b}^{\lceil\log_\mathsf{b} q\rceil-1})$ and the gadget matrix [MP12] $\mathbf{G} = \mathbb{I}\otimes\mathbf{g}\in\mathcal{R}_q^{n\times n\lceil\log_\mathsf{b} q\rceil}$. We recall lattice trapdoor generation and LWE sample inversion adapted from [MP12] for ring-element matrices. The bounds follows from Lemma A.1.

Lemma 2.9 (Gadget Trapdoor [MP12]). *Let $n,\mathsf{b},q \geqslant 2$ be positive integers, $m > n\lceil\log_\mathsf{b} q\rceil + n$, $\varepsilon > 0$. Then, there exist the following algorithms:*

- $\mathsf{GenTrap}(q, \mathsf{b}, n, m)$: *is a probabilistic algorithm outputting* $(\mathbf{A} \in \mathcal{R}_q^{n\times m}, \mathbf{R})$ *such that* $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbb{I}\end{bmatrix} = \mathbf{G} \pmod q$. *More precisely,* $\mathbf{A} = [\mathbf{B} \mid \mathbf{G} - \mathbf{BR}]$ *for* $\mathbf{B} \leftarrow\!\!\$\ \mathcal{R}^{n\times(m-n\lceil\log_{\mathsf{b}} q\rceil)}$ *and* $\mathbf{R} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^{(m-n\lceil\log_{\mathsf{b}} q\rceil)\times n\lceil\log_{\mathsf{b}} q\rceil}, \sigma_{\mathsf{td}}}$. *We have that for* $\sigma_{\mathsf{td}} \geqslant \frac{8}{\sqrt{\pi}} q^{\frac{n}{m-n\lceil\log_{\mathsf{b}} q\rceil}} \sqrt{N \ln(2(m-n\lceil\log_{\mathsf{b}} q\rceil)N(1+1/\varepsilon))}$, *the following distribution* $\{\mathbf{A} : (\mathbf{A}, \mathbf{R}) \leftarrow\!\!\$\ \mathsf{GenTrap}(q, \mathsf{b}, n, m)\}$ *is within* $2^{-N} + 2\varepsilon$ *statistical distance from the uniform distribution on* $\mathcal{R}_q^{n\times m}$.
- $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{t} \in \mathcal{R}_q^m)$: *is a deterministic algorithm outputting* $(\mathbf{s}, \mathbf{e})$. *If* $\mathbf{t} = \mathbf{sA} + \mathbf{e}$ *for some* $\mathbf{s} \in \mathcal{R}_q^n$ *and* $\|\mathbf{e}\| \leqslant \frac{q}{2\sqrt{(\mathsf{b}^2+1)(\|\mathbf{R}\|^2+1)}}$ *(for* $(\mathbf{A}, \mathbf{R})$ *output from* $\mathsf{GenTrap}$*), the output of* $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{t} \in \mathcal{R}_q^m)$ *is exactly* $(\mathbf{s}, \mathbf{e})$.

REJECTION SAMPLING. We recall results about rejection sampling implied by [Lyu12, Lemma 4.4, 4.5, and 4.7, and Theorem 4.6] which were used in several prior works [LNP22, DOTT22]. We adapt the statement to give concrete bounds based on a parameter t as stated in [DOTT22, Lemma 2 and 3]. Note that these can be adapted directly to rings with the coefficient embeddings (so m is replaced with mN).

Lemma 2.10 (Rejection Sampling [Lyu12, LNP22]). *Let* $V \subseteq \mathbb{Z}^m$ *with* $\max_{\mathbf{v}\in V} \|\mathbf{v}\| \leqslant T$, $\sigma_r = \alpha T$, *with* $\alpha = \omega(\sqrt{\log(m)})$ *and fix some* $t = \omega(\sqrt{\log(m)}) > 0$. *For any* $\mathbf{v} \in V$, *the following two distributions* $\mathcal{D}_1, \mathcal{D}_2$ *have statistical distance* $\frac{\delta}{M}$:

1. $\mathcal{D}_1$: *Sample* $\mathbf{r} \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z}^m,\sigma_r}$ *and return* $\mathbf{z} = \mathbf{r} + \mathbf{v}$ *with probability* $\min\left(1, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_r}(\mathbf{w})}{M\mathcal{D}_{\mathbb{Z}^m,\mathbf{v},\sigma_r}(\mathbf{w})}\right)$. *Otherwise, output* $\perp$.
2. $\mathcal{D}_2$: *Sample* $\mathbf{z} \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z}^m,\sigma_r}$ *and return* $\mathbf{z}$ *with probability* $1/M$. *Otherwise, output* $\perp$.

Here, $M = \exp\left(\frac{t}{\alpha} + \frac{1}{2\alpha^2}\right)$ *and* $\delta = 2\exp\left(-\frac{t^2}{2}\right)$. *Moreover, the first distribution outputs* $\neq \perp$ *with probability at least* $\frac{1-\delta}{M}$.

2.4 Assumptions

We restate the Search Module Learning with Errors (sMLWE) assumption with discrete Gaussian errors. The assumption $\mathsf{sMLWE}_{n,m,q,\sigma}$ was considered in [LS15] was shown to be as hard as the worst-case Module Generalized Independent Vectors Problem (Mod-GIVP) for approximation factor $\gamma = \sqrt{8N}n \cdot \omega(\sqrt{\log N})q/\sigma$ when $\sigma > \sqrt{n} \cdot \omega(\sqrt{\log N})$.

Definition 2.11 (Search MLWE). *Let* $n = n(\kappa), m = m(\kappa), q = q(\kappa), \sigma = \sigma(\kappa), N = N(\kappa)$ *with* N *being a power of two, and* $\mathcal{R} = \mathbb{Z}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$. *The* $\mathsf{sMLWE}_{n,m,q,\sigma}$ *assumption states that for any PPT* $\mathcal{A}$, *the following advantage defined over* $\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}, \mathbf{s} \leftarrow\!\!\$\ \mathcal{R}_q^n, \mathbf{e} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma}$ *is negligible in* κ

$$\mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma}(\mathcal{A}, \kappa) := \Pr\left[\mathbf{s} = \mathbf{s}' \middle| \mathbf{s}' \leftarrow\!\!\$\ \mathcal{A}(\mathbf{A}, \mathbf{sA} + \mathbf{e})\right] .$$

3 Global Advantage from Monotone Couplings

In this section we present a coupling-based technique for analyzing multi-round experiments in which each round draws a fresh sample and the adversary's success is decided by a single-round predicate. The main message is that, under a per-query ("local") dominance condition, a reduction's overall success probability over m adaptive rounds can be lower bounded by reasoning about a single round. Moreover, the resulting transformation incurs only constant-factor overhead in predicate evaluations.

3.1 Real and Ideal Experiments

Let D be a distribution defined on a finite support X. Consider the following idealized experiment where an adversary, specified by a function $\mathcal{A}$ that gets to make adaptive queries $q_1, q_2, \ldots$ from some query space $\mathcal{Q}$ to an oracle. Every time the adversary makes a query q_i, we sample a fresh $c \leftarrow\!\!\$\ D$ and evaluate a predicate $A(c, q_i)$. If $A(c, q_i) = 1$, we declare a win and output $\top$. Otherwise the game continues.

Experiment Real :

- Set $\mathsf{st}_1 = \varnothing$ and $c_1 = \varnothing$.
- For $i = 1, \ldots, m$:
  - Compute $(q_i, \mathsf{st}_{i+1}) = \mathcal{A}(c_i, \mathsf{st}_i)$
  - Choose $c \leftarrow\!\!\$\ D$.
  - If it holds $A(c, q_i) = 1$ output $\top$
  - Otherwise, set $c_{i+1} = c$.
- Output $\perp$

Now consider a second distribution D' defined on a domain X' and a second predicate $B(c', q)$ which takes a $c' \in X'$ and a $q \in \mathcal{Q}$ and outputs 0 or 1 based on c' and the query q. Our goal is to find a $c' \in X'$ and a $q \in \mathcal{Q}$ with $B(c', q) = 1$. Looking ahead to the next section, such a pair (c', q_i) will enable us to solve an underlying computational problem. Our goal now is to find such a pair (c', q_i) with the same probability that the adversary wins the real experiment (i.e. $\mathsf{Pr}[\mathsf{Real} = \top]$), while only incurring a small additive overhead over the runtime/complexity of $\mathcal{A}$ that essentially accounts for processing $\mathcal{A}$'s queries.

Consider the following experiment Ideal.

Experiment Ideal :

- Set $\mathsf{st}_1 = \varnothing$ and $c_1 = \varnothing$.
- For $i = 1, \ldots, m$:
  - Compute $(q_i, \mathsf{st}_{i+1}) = \mathcal{A}(c_i, \mathsf{st}_i)$
  - Repeat the following steps until an exit condition is reached.
    - Sample an unbiased coin $\beta \leftarrow\!\!\$\ \{0, 1\}$
    - If $\beta = 1$ sample $c' \leftarrow\!\!\$\ D'$, and if $B(c', q_i) = 1$ output $\top$. Otherwise continue the repeat loop.
    - If $\beta = 0$ sample $c \leftarrow\!\!\$\ D$ and if $A(c, q_i) = 0$ set $c_{i+1} = c$ and exit the repeat loop. Otherwise continue the repeat loop.
- Output $\perp$

Clearly, if the experiment Ideal outputs $\top$, then we have found a $c' \in X'$ and a $q_i \in \mathcal{Q}$ with $B(c', q_i) = 1$. Hence it suffices to lower bound the probability that Ideal outputs $\top$.

3.2 From Local to Global Advantage

We say that B has local advantage over A if it holds for every query q in the query space $\mathcal{Q}$ that

$$\mathsf{Pr}_{c' \leftrightarrow\!\!\$\ D'}[B(c', q) = 1] \geqslant \mathsf{Pr}_{c \leftrightarrow\!\!\$\ D}[A(c, q) = 1].$$

In the rest of this section we will establish that given that the *local advantage condition holds*, then there also is a global advantage:

$$\mathsf{Pr}[\mathsf{Ideal} = \top] \geqslant \mathsf{Pr}[\mathsf{Real} = \top].$$

Furthermore, we will show that the number of queries made by Ideal to A and B is only twice the number of queries made by Real to A in expectation.

Success Probability. As discussed above, the event $A(c, q_i) = 1$ triggers the success condition of the adversary in the experiment Real in round i, whereas $B(c', q_i) = 1$ triggers the success condition of the adversary in the experiment Ideal in round i. To declutter notation, we fix a query $q \in \mathcal{Q}$ and denote $U(c) = A(c, q)$ and $V(c') = B(c', q)$. The following Lemma is the technical core of our proof.

Note that for a $c \leftarrow\!\!\$\ D$ the event $U(c) = 1$ follows a Bernoulli distribution. Specifically, if we set $p = \mathsf{Pr}_{c \leftarrow\!\!\$\ D}[U(c) = 1]$, then $U(c)$ follows Ber_p. Consider a distribution Y_1 which draws a sample $c \leftarrow\!\!\$\ D$, and whenever $U(c) = 1$ it outputs $\top$, otherwise c. Hence, this distribution outputs $\top$ with probability p. Furthermore, if it does not output $\top$ then we have the guarantee that its output c follows D conditioned on $U(c) = 0$.

Now consider a second distribution D' and a second predicate $V(\cdot)$, and assume for simplicity that it holds that $\mathsf{Pr}_{c' \leftarrow\!\!\$\ D'}[V(c') = 1] = \mathsf{Pr}_{c \leftarrow\!\!\$\ D}[U(c) = 1]$. Ideally, we would like to do the following: First sample $c' \leftarrow\!\!\$\ D'$ and check if $V(c') = 1$ to determine if we output $\top$. If this is not the case, we sample a $c \leftarrow\!\!\$\ D$ conditioned on $U(c) = 0$. Doing this naively could trigger a large number of resampling attempts for c, as $\mathsf{Pr}_c[U(c) = 0]$ could be very small. The lemma provides an efficient sampler that produces exactly the same distribution as the first sampler, but if it outputs $\top$ we also obtain a c' with $V(c') = 1$. This sampler only needs two sampling attempts in expectation. Importantly, the lemma also generalizes to the case that $\mathsf{Pr}_{c' \leftarrow\!\!\$\ D'}[V(c') = 1] \geqslant \mathsf{Pr}_{c \leftarrow\!\!\$\ D}[U(c) = 1]$ via a small modification of the first sampler that artificially increases its probability of outputting $\top$.

Lemma 3.1. *Let D and D' be probability distributions on some finite supports X and X' respectively. Furthermore, let $U(\cdot)$ and $V(\cdot)$ be predicates defined on X and X' respectively. Let $p = \mathsf{Pr}_{c \leftarrow\!\!\$\ D}[U(c) = 1]$ and $p' = \mathsf{Pr}_{c' \leftarrow\!\!\$\ D'}[V(c') = 1]$ and assume that $p' \geqslant p$. Set $\delta = \frac{p'-p}{1+p'-p} \in [0, 1/2]$.*

Consider the following random variables Y_1 and Y_2 supported on $X \cup \{\top\}$, where $\top \notin X$ is a special output symbol indicating a success.

- *Y_1: Choose $z \leftarrow\!\!\$\ \mathsf{Ber}_\delta$. If $z = 1$ output $\top$. Otherwise:*
  - *Sample $c \leftarrow\!\!\$\ D$.*
  - *If $U(c) = 1$ output $\top$, otherwise c.*
- *Y_2: Repeat the following steps until an output is produced*
  - *Sample an unbiased coin $\beta \leftarrow\!\!\$\ \{0, 1\}$*
  - *If $\beta = 1$ sample $c' \leftarrow\!\!\$\ D'$, and if $V(c') = 1$ output $\top$.*
  - *If $\beta = 0$ sample $c \leftarrow\!\!\$\ D$, and if $U(c) = 0$ output c.*

Then it holds that Y_1 and Y_2 are identically distributed. Furthermore, the stopping time T for the sampler of Y_2 follows the geometric distribution with parameter $\gamma = \frac{1}{2}(1 + p' - p) \geqslant \frac{1}{2}$. Hence sampling Y_2 takes $\leqslant 2$ tries in expectation.

Proof. Both definitions never output $x \in X$ with $U(x) = 1$. By definition of Y_1 it holds that

$$\begin{aligned}
\mathsf{Pr}[Y_1 = \top] &= \mathsf{Pr}_{z,c}[z = 1 \text{ or } U(c) = 1] \\
&= \mathsf{Pr}_z[z = 1] + \mathsf{Pr}[U(c) = 1] - \mathsf{Pr}[z = 1] \cdot \mathsf{Pr}[U(c) = 1] \\
&= \delta + p - \delta \cdot p \\
&= p + \frac{(p' - p)(1 - p)}{1 + p' - p} \\
&= \frac{p'}{1 + p' - p} \,.
\end{aligned}$$

Moreover, it holds for every $x \in X$ with $U(x) = 0$ that

$$\mathsf{Pr}[Y_1 = x] = \mathsf{Pr}[z = 0 \text{ and } c = x] = \left(1 - \frac{p' - p}{1 + p' - p}\right) \cdot \mathsf{Pr}[c = x] = \frac{\mathsf{Pr}[c = x]}{1 + p' - p} \,,$$

where the second equality holds as z and c are independent.

We will analyze Y_2 in a round-by-round fashion. Assume that the sampling procedure has reached round i, i.e. no output has been produced in rounds $1, \ldots, i-1$. It holds that

- $\mathsf{Pr}[\text{Output } \top \text{ in round } i] = \frac{1}{2}\mathsf{Pr}_{c'}[V(c') = 1] = \frac{1}{2} \cdot p'$
- For every $x \in X$ with $U(x) = 0$ it holds that $\mathsf{Pr}[\text{Output } x \text{ in round } i] = \frac{1}{2} \cdot \mathsf{Pr}[c = x]$.

Hence it holds that

$$\mathsf{Pr}[\text{Output some } c \in X \text{ in round } i] = \sum_{x \in X \text{ s.t. } U(x)=0} \frac{1}{2}\mathsf{Pr}[c = x] = \frac{1}{2}\mathsf{Pr}_c[U(c) = 0] = \frac{1}{2} \cdot (1-p) \,.$$

It follows for each round i that

$$\mathsf{Pr}[\text{No output in round } i] = 1 - \frac{1}{2}(1 + p' - p) = \frac{1}{2}(1 - p' + p) =: 1 - \gamma \,.$$

Note that $\gamma \geqslant 1/2$ as $p' \geqslant p$. Hence, as all rounds are independent it holds that

$$\begin{aligned}
\mathsf{Pr}[Y_2 = \top] &= \sum_{i=1}^{\infty} (1-\gamma)^{i-1} \cdot \mathsf{Pr}[\text{Output } \top \text{ in round } i] \\
&= \left(\sum_{i=0}^{\infty} (1-\gamma)^i \right) \frac{1}{2} p' = \frac{p'}{2\gamma} = \frac{p'}{1 + p' - p} \,.
\end{aligned}$$

Note further that this also immediately yields that $\mathsf{Pr}[Y_2 \neq \top] = \frac{1-p}{1+p'-p}$. It follows via the same geometric sum argument that for $x \in X$ with $U(x) = 0$ that

$$\begin{aligned}
\mathsf{Pr}[Y_2 = x] &= \sum_{i=1}^{\infty} (1-\gamma)^{i-1} \cdot \mathsf{Pr}[\text{Output } x \text{ in round } i] \\
&= \left(\sum_{i=0}^{\infty} (1-\gamma)^i \right) \frac{1}{2} \mathsf{Pr}[c = x] \\
&= \frac{\mathsf{Pr}[c = x]}{2\gamma} \\
&= \frac{\mathsf{Pr}[c = x]}{1 + p' - p} \,.
\end{aligned}$$

Consequently, Y_1 and Y_2 are identically distributed.

To establish the second claim, note that the probability that the stopping time T is i is exactly $(1-\gamma)^{i-1} \cdot \gamma$. Hence, we get that

$$\begin{aligned}
\mathsf{E}[T] &= \sum_{i=1}^{\infty} i \cdot (1-\gamma)^{i-1} \cdot \gamma \\
&= \frac{\gamma}{(1 - (1-\gamma))^2} \\
&= \frac{1}{\gamma} \\
&\leqslant 2 \,.
\end{aligned}$$

□

We can now prove the main theorem of this section, which establishes that $\mathsf{Pr}[\mathsf{Ideal} = \top] \geqslant \mathsf{Pr}[\mathsf{Real} = \top]$ via a simple monotone coupling argument.

Theorem 3.2. *Let $\mathcal{Q}$, X, X', D, D', $A(\cdot,\cdot)$ and $B(\cdot,\cdot)$ be as above. Assume that it holds for every $q \in \mathcal{Q}$ that $\mathsf{Pr}_{c' \leftarrow\!\$\, D'}[B(c',q)=1] \geqslant \mathsf{Pr}_{c \leftarrow\!\$\, D}[A(c,q)=1]$. Then it holds that $\mathsf{Pr}[\mathsf{Ideal}=\top] \geqslant \mathsf{Pr}[\mathsf{Real}=\top]$.*

Proof. We will show the claim via a sequence of hybrid experiments.

- Hybrid H_0: This is the experiment Real.
- Hybrid $H_{1,i}$ (for $i=1,\ldots,m$, where we set $H_{1,0}=H_0$): This is identical to $H_{1,i-1}$, except that the i-th query q_i is processed differently. Let $p=\mathsf{Pr}_{c \leftarrow\!\$\, D}[A(c,q_i)=1]$ and $p'=\mathsf{Pr}_{c' \leftarrow\!\$\, D'}[B(c',q_i)=1]$ and set $\delta=\frac{p'-p}{1+p'-p} \in [0,1/2]$. In the i-th query, compute the response as follows: Choose $z_i \leftarrow\!\$\, \mathsf{Ber}_\delta$. If $z_i=1$ output $\top$. Otherwise:
  - Sample $c \leftarrow\!\$\, D$.
  - If $A(c,q_i)=1$ output $\top$, otherwise set $c_{i+1}=c$.

We will briefly show that $\mathsf{Pr}[H_{1,i}=\top] \geqslant \mathsf{Pr}[H_{1,i-1}=\top]$. Clearly, conditioned on $z_i=0$ it holds that $H_{1,i-1}$ and $H_{1,i}$ are identically distributed, hence $\mathsf{Pr}[H_{1,i}=\top|z_i=0]=\mathsf{Pr}[H_{1,i-1}=\top]$. On the other hand, if $z_i=1$ then $H_{1,i}$ will output $\top$ with certainty if it reaches the i-th query and is distributed identically to $H_{1,i-1}$ before the i-th query. Consequently it holds that $\mathsf{Pr}[H_{1,i}=\top|z_i=1] \geqslant \mathsf{Pr}[H_{1,i-1}=\top]$ and we can conclude

$$\begin{aligned}\mathsf{Pr}[H_{1,i}=\top] &= \mathsf{Pr}[z_i=0]\cdot\mathsf{Pr}[H_{1,i}=\top|z_i=0]+\mathsf{Pr}[z_i=1]\cdot\mathsf{Pr}[H_{1,i}=\top|z_i=1]\\ &\geqslant \mathsf{Pr}[z_i=0]\cdot\mathsf{Pr}[H_{1,i-1}=\top]+\mathsf{Pr}[z_i=1]\cdot\mathsf{Pr}[H_{1,i-1}=\top]\\ &= \mathsf{Pr}[H_{1,i-1}=\top].\end{aligned}$$

The remaining hybrids are given as follows.

- Hybrid H_2: This is identical to $H_{1,m}$ (only for notational convenience).
- Hybrid $H_{3,i}$ (for $i=1,\ldots,m$, where we set $H_{3,0}=H_2$): This is identical to $H_{1,i-1}$, except that the i-th query q_i is processed differently. In the i-th query, repeat the following steps until an output Y_i is produced
  - Sample an unbiased coin $\beta \leftarrow\!\$\, \{0,1\}$
  - If $\beta=1$ sample $c' \leftarrow\!\$\, D'$, and if $B(c',q_i)=1$ output $Y_i=\top$. Otherwise continue loop.
  - If $\beta=0$ sample $c \leftarrow\!\$\, D$ and if $A(c,q_i)=0$ output $Y_i=c$. Otherwise continue loop.

  It follows immediately by Lemma 3.1 that hybrids $H_{3,i-1}$ and $H_{3,i}$ are identically distributed where we set $U(c)=A(c,q_i)$ and $V(c')=B(c',q_i)$.
- Hybrid H_4: This is identical to $H_{3,m}$ (only for notational convenience). This hybrid H_4 is identical to Ideal.

Efficiency. We will briefly discuss the efficiency of the experiment Ideal. By the third item of Lemma 3.1, as $\mathsf{Pr}_{c'}[B(c',q_i)=1] \geqslant \mathsf{Pr}_c[A(c,q_i)=1]$ we get that for each i the experiment Ideal makes no more than 2 sampling attempts in expectation. Hence, if we assign the same runtime cost to A and B, the expected runtime of Ideal is at most twice the expected runtime of Real.

Remark 3.3 (Probabilistic adversaries). We remark that the above theorem is stated for deterministic algorithms $\mathcal{A}$. However, we can easily extend it to probabilistic $\mathcal{A}$ by fixing its random coins, applying the theorem, and considering the average success probability over the random coins of $\mathcal{A}$. This also applies to the results in the following subsections.

3.3 Strict Polynomial Time

We will need to consider a strict polynomial time version of the experiment Ideal. In this version, we will count the total number of restarts in the inner loop and abort if it exceeds a certain threshold. Let $\delta = \delta(\kappa)$ be an error bound, let $\tau > 1$ be a constant and set $T = -\frac{\ln(\delta \cdot (1 - e^{-2\gamma^2}))}{2\gamma^2}$, where $\gamma = \frac{(\tau - 1)}{2(\tau + 1)}$ is a constant depending on τ.

Experiment Ideal' :

- Set $\mathsf{st}_1 = \varnothing$, $c_1 = \varnothing$, and $\mathsf{rst} = 0$
- For $i = 1, \ldots, m$:
  - Compute $(q_i, \mathsf{st}_{i+1}) = \mathcal{A}(c_i, \mathsf{st}_i)$
  - Repeat the following steps until an exit condition is reached.
    - If $\mathsf{rst} > \max(\tau \cdot i, T)$ output $\perp$.
    - Sample an unbiased coin $\beta \leftarrow\!\!\$\ \{0, 1\}$
    - If $\beta = 1$ sample $c' \leftarrow\!\!\$\ D'$, and if $B(c', q_i) = 1$ output $\top$. Otherwise set $\mathsf{rst} = \mathsf{rst} + 1$ and continue repeat loop.
    - If $\beta = 0$ sample $c \leftarrow\!\!\$\ D$ and if $A(c, q_i) = 0$ set $c_{i+1} = c$ and exit repeat loop. Otherwise set $\mathsf{rst} = \mathsf{rst} + 1$ and continue repeat loop.
- Output $\perp$

First observe that unless Ideal' outputs $\perp$, its output is identically distributed to that of Ideal. Furthermore, Ideal' does abort if the number of restarts rst is larger than $\tau \cdot i$ for any $i > T$. Hence, for $i > T$ the average number of restarts per iteration is at most τ. It follows that the runtime of Ideal' is at most $1 + \tau$ times the runtime of Real.

Recall the Hoeffding bound for centered binomial distributions.

Lemma 3.4 (Hoeffding Bound). *Let $x_1, \ldots, x_n \leftarrow\!\!\$\ \mathsf{Ber}_{1/2}$. Then it holds for every $u \geqslant 0$ that*

$$\mathsf{Pr}\left[\sum_{i=1}^{n} x_i \geqslant \frac{n}{2} + u\right] \leqslant e^{-2u^2/n}.$$

We will need the following anytime Hoeffding bound.

Lemma 3.5. *Let $x_1, x_2, \ldots$ be a potentially infinite sequence of i.i.d. $\mathsf{Ber}_{1/2}$ distributed random variables. Let $\delta > 0$, $\gamma > 0$ and $T = -\frac{\ln(\delta \cdot (1 - e^{-2\gamma^2}))}{2\gamma^2}$. Then it holds that*

$$\mathsf{Pr}\left[\exists n \geqslant T : \sum_{i=1}^{n} x_i \geqslant \left(\frac{1}{2} + \gamma\right) n\right] < \delta.$$

Proof. It holds that

$$\begin{aligned}\mathsf{Pr}\left[\exists n \geqslant T \text{ s.t. } \sum_{i=1}^{n} x_i \geqslant \left(\frac{1}{2}+\gamma\right) n\right] &\leqslant \sum_{n \geqslant T} \mathsf{Pr}\left[\sum_{i=1}^{n} x_i \geqslant \left(\frac{1}{2}+\gamma\right) n\right] \\ &\leqslant \sum_{n \geqslant T} e^{-2\gamma^2 n} \\ &= e^{-2\gamma^2 T} \cdot \sum_{n=0}^{\infty} e^{-2\gamma^2 n} \\ &= e^{-2\gamma^2 T} \cdot \frac{1}{1-e^{-2\gamma^2}} \\ &= e^{-2\gamma^2 T - \ln(1-e^{-2\gamma^2})} = \delta.\end{aligned}$$

Here the first inequality is just the union bound and the second inequality is the Hoeffding bound (Lemma 3.4). □

Theorem 3.6. *Assume that in experiment* Ideal' *the adversary* $\mathcal{A}$ *makes at least* T *queries. Then the amortized complexity to answer a query is at most the equivalent of* 3 *calls to* A *or* B*. Moreover,* Ideal' *outputs* $\perp$ *with probability at most* δ *and conditioned on not outputting* $\perp$ *it is identically distributed to* Ideal*. Hence the statistical distance of* Ideal' *and* Ideal *is at most* δ*.*

Proof. We will show that given that there are at least T oracle queries, the probability that the *total* number of restarts of the inner loop of Ideal' across all iterations of the outer loop exceeds $\tau \cdot i$ is at most δ. We view the experiment from a slightly different angle. Each iteration of the inner loop is a Bernoulli trial which determines if a restart of the inner loop occurs or not. Now let j be an index that counts the total number of iterations of the inner loop across all iterations of the outer loop. By Lemma 3.1, for each iteration j of the inner loop the event y_j that a restart occurs follows a Bernoulli distribution Ber_{γ_j} for some parameter $\gamma_j \leqslant 1/2$. Note that while the y_j are independent, the parameters γ_j potentially depend on $y_1, \ldots, y_{j-1}$. We will show that for each iteration i of the outer loop the probability that the total number of restarts exceeds $\tau \cdot i$ is at most δ. The proof proceeds via a monotone coupling argument. Let $n = i + \mathsf{rst}$ be the total number of runs of the inner loop. The number of restarts in n runs of the inner loop follows $\sum_{j=1}^{n} y_j$, where $y_j \leftarrow\!\!\$\, \mathsf{Ber}_{\gamma_j}$. Now let $e_1, \ldots, e_n$ be such that e_i (independently) follow Ber_{τ_j} where $\tau_i = \frac{1-2\gamma_j}{2(1-\gamma_j)}$. Note that $\tau_j \in [0, 1/2]$ as $\gamma_j \in [0, 1/2]$. Now set $x_j = y_j \vee e_j$. Then it holds that

$$\begin{aligned}\mathsf{Pr}[x_j = 1] &= \mathsf{Pr}[y_j = 1] + \mathsf{Pr}[e_j = 1] - \mathsf{Pr}[x_j = 1] \cdot \mathsf{Pr}[e_j = 1] \\ &= \gamma_j + \tau_j - \gamma_j \tau_j \\ &= \gamma_j + \tau_j(1-\gamma_j) \\ &= \gamma_j + \frac{1}{2} - \gamma_j = \frac{1}{2}.\end{aligned}$$

Consequently, it holds that x_j follows $\mathsf{Ber}_{1/2}$ and that $y_j \leqslant x_j$ with probability 1 (as $\vee$ is monotone). Hence, it also holds that

$$\sum_{i=1}^{n} y_i \leqslant \sum_{i=1}^{n} x_i$$

with probability 1. Thus, the probability that in *any number* $n \geqslant T$ of runs of the inner loop the number of restarts exceeds $\left(\frac{1}{2}+\gamma\right) n$ is

$$\begin{aligned}\mathsf{Pr}\left[\exists n \geqslant T \text{ s.t. } \sum_{j=1}^{n} y_j \geqslant \left(\frac{1}{2}+\gamma\right) n\right] &\leqslant \mathsf{Pr}\left[\exists n \geqslant T \text{ s.t. } \sum_{j=1}^{n} x_j \geqslant \left(\frac{1}{2}+\gamma\right) n\right] \\ &< \delta\end{aligned}$$

by Lemma 3.5.

Fix an n and recall that $\mathsf{rst} = \sum_{j=1}^{n} y_j$. As $n = i + \mathsf{rst}$ it holds that

$$\begin{aligned} & \mathsf{rst} \geqslant \left(\frac{1}{2} + \gamma\right) n \\ \iff \quad & \mathsf{rst} \geqslant \left(\frac{1}{2} + \gamma\right) (i + \mathsf{rst}) \\ \iff \quad & \mathsf{rst} \geqslant \frac{1 + 2\gamma}{1 - 2\gamma} i = \tau \cdot i. \end{aligned}$$

Hence it follows that

$$\mathsf{Pr}[\exists i \geqslant T \text{ s.t. } \mathsf{rst} \geqslant \tau \cdot i] \leqslant \delta. \qquad \square$$

We obtain the following corollary of Theorems 3.2 and 3.6 by setting $\delta = 2^{-\kappa}$, choosing $T = O(\kappa)$ appropriately and adding T dummy queries. Further note that unless Ideal' outputs $\perp$, Ideal' and Ideal are identically distributed and therefore $\mathsf{Pr}[\mathsf{Ideal}' = \top] \geqslant \mathsf{Pr}[\mathsf{Ideal} = \top] - \delta \geqslant \mathsf{Pr}[\mathsf{Real} = \top] - \delta$.

Corollary 3.7. *Let $T = O(\kappa)$ and let $S(\kappa)$ be an upper bound on the time complexity of a call to either A or B. Assume that an adversary $\mathcal{A}$ makes $\mathcal{Q}(\mathcal{A})$ oracle queries. Let $\mathsf{Time}(\mathcal{A})$ be the runtime/complexity of $\mathcal{A}$ and $\mathsf{Time}(\mathsf{Ideal}')$ be the runtime of Ideal'. Then, the following holds*

$$\begin{aligned} \mathsf{Time}(\mathsf{Ideal}') &\leqslant \mathsf{Time}(\mathcal{A}) + 3 \cdot S(\kappa) \cdot (\mathcal{Q}(\mathcal{A}) + T)\ , \text{ and} \\ \mathsf{Pr}[\mathsf{Real} = \top] &\leqslant \mathsf{Pr}[\mathsf{Ideal}' = \top] + 2^{-\kappa}\ . \end{aligned}$$

3.4 Product Predicates

We will briefly discuss the main type of predicates we will be interested in, a type of predicates we call *product predicates*. In short, we call a predicate $U(x_1, x_2)$ a product predicate if it can be written as the product of two predicates $P_1(x_1)$ and $P_2(x_2)$, that is $U(x_1, x_2) = P_1(x_1) \cdot P_2(x_2)$.

Lemma 3.8. *Let P_1 and P_2 be predicates defined on X, and let D be a distribution on X. Define the predicates U and V as follows. U is defined on X^2 and we set*

$$U(x_1, x_2) = P_1(x_1) \cdot P_2(x_2),$$

that is, $U(x_1, x_2)$ is the logical and *of $P_1(x_1)$ and $P_2(x_2)$. The predicate V is defined on $X^2 \times X^2$ via*

$$V(x_1, x_1', x_2, x_2') = P_1(x_1) \cdot P_1(x_1') \vee P_2(x_2) \cdot P_2(x_2').$$

Now let c follow some distribution D on X. Then it holds that

$$\mathsf{Pr}_{c_1, c_2 \leftarrow\$ D}[U(c_1, c_2) = 1] \leqslant \mathsf{Pr}_{c_1, c_1', c_2, c_2' \leftarrow\$ D}[V(c_1, c_1', c_2, c_2') = 1].$$

Proof. Let $p_1 = \mathsf{Pr}_{c \leftarrow\$ D}[P_1(c) = 1]$ and $p_2 = \mathsf{Pr}_{c \leftarrow\$ D}[P_2(c) = 1]$. Then it holds that

$$\begin{aligned} \mathsf{Pr}_{c_1, c_2 \leftarrow\$ D}[U(c_1, c_2) = 1] &= \mathsf{Pr}_{c_1, c_2 \leftarrow\$ D}[P_1(c_1) = 1, P_2(c_2) = 1] \\ &= \mathsf{Pr}[P_1(c_1) = 1] \cdot \mathsf{Pr}[P_2(c_2) = 1] = p_1 \cdot p_2. \end{aligned}$$

Furthermore,

$$\begin{aligned} & \mathsf{Pr}_{c_1, c_1', c_2, c_2' \leftarrow\$ D}[V(c_1, c_1', c_2, c_2') = 1] \\ &= \mathsf{Pr}[P_1(c_1) \cdot P_1(c_1') = 1 \text{ or } P_2(c_2) \cdot P_2(c_2') = 1] \\ &= \mathsf{Pr}[P_1(c_1) = 1, P_1(c_1') = 1] + \mathsf{Pr}[P_2(c_2) = 1, P_2(c_2') = 1] \\ &\quad - \mathsf{Pr}[P_1(c_1) = 1, P_1(c_1') = 1, P_2(c_2) = 1, P_2(c_2') = 1] \\ &= p_1^2 + p_2^2 - p_1^2 \cdot p_2^2. \end{aligned}$$

Algorithm $\mathsf{SS}_{\mathcal{R}}.\mathsf{KeyGen}(1^{\kappa})$:

$\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}$

$\mathbf{s} \leftarrow\!\!\$\ \mathcal{R}_q^{n}; \mathbf{e} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_{\mathsf{sk}}}$

$\mathbf{t} \leftarrow \mathbf{s}\mathbf{A} + \mathbf{e}$

return $(\mathsf{pk} = (\mathbf{A}, \mathbf{t}), \mathsf{sk} = (\mathsf{pk}, \mathbf{s}, \mathbf{e}))$

Algorithm $\mathsf{SS}_{\mathcal{R}}.\mathsf{V}(\mathsf{pk}, \mu, \sigma)$:

parse $\sigma = (\bar{\mathbf{t}}, (\mathbf{u}_b, \bar{\mathbf{u}}_b, \mathbf{z}_b)_{b\in\{0,1\}})$

$\mathbf{H} \leftarrow \mathsf{H}_1(\mathbf{u}_0, \mathbf{u}_1, \mu)$

$(c_0, c_1) \leftarrow \mathsf{H}_2((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$

return $(\forall b \in \{0,1\}:$

$\quad \|\mathbf{u}_b + c_b \cdot \mathbf{t} - \mathbf{z}_b\mathbf{A}\| \leqslant \beta_{\mathbf{z}}$

$\quad \wedge\ \|\bar{\mathbf{u}}_b + c_b \cdot \bar{\mathbf{t}} - \mathbf{z}_b\mathbf{H}\| \leqslant \bar{\beta}_{\mathbf{z}})$

Algorithm $\mathsf{Rej}(\mathbf{w}, \mathbf{v}, \sigma)$:

$\delta \leftarrow\!\!\$\ [0,1)$ // Rejection Sampling

return $\left(\delta < \min\left(\frac{\mathcal{D}_{\mathcal{R}^m,\sigma}(\mathbf{w})}{M\mathcal{D}_{\mathcal{R}^m,\mathbf{v},\sigma}(\mathbf{w})}, 1\right)\right)$

Algorithm $\mathsf{SS}_{\mathcal{R}}.\mathsf{Sign}(\mathsf{sk}, \mu)$:

for $i \in [\ell]$: // Repeat until output

$\quad$ **for** $b \in \{0,1\}$:

$\quad\quad \mathbf{r}_b \leftarrow\!\!\$\ \mathcal{R}_q^n$; $\mathbf{d}_b \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_r}$

$\quad\quad \bar{\mathbf{d}}_b \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\bar{\sigma}_r}$

$\quad\quad \mathbf{u}_b \leftarrow \mathbf{r}_b\mathbf{A} + \mathbf{d}_b$

$\quad \mathbf{H} \leftarrow \mathsf{H}_1(\mathbf{u}_0, \mathbf{u}_1, \mu)$

$\quad \bar{\mathbf{e}} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sqrt{\mathbb{I}_{m'}\otimes\bar{\mathbf{\Sigma}}}}$; $\bar{\mathbf{t}} \leftarrow \mathbf{s}\mathbf{H} + \bar{\mathbf{e}}$

$\quad$ **for** $b \in \{0,1\} : \bar{\mathbf{u}}_b \leftarrow \mathbf{r}_b\mathbf{H} + \bar{\mathbf{d}}_b$

$\quad (c_0, c_1) \leftarrow \mathsf{H}_2((\mathbf{u}_b, \bar{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \bar{\mathbf{t}}, \mu)$

$\quad$ **for** $b \in \{0,1\}$:

$\quad\quad \mathbf{z}_b \leftarrow \mathbf{r}_b + c_b \cdot \mathbf{s}$

$\quad\quad \mathbf{f}_b \leftarrow \mathbf{d}_b + c_b \cdot \mathbf{e}$; $\bar{\mathbf{f}}_b \leftarrow \bar{\mathbf{d}}_b + c_b \cdot \bar{\mathbf{e}}$

$\quad$ **if** $\forall b \in \{0,1\} : \neg\mathsf{Rej}(\mathbf{f}_b, c_b \cdot \mathbf{e}, \sigma_r) \wedge$

$\quad\quad \neg\mathsf{Rej}(\bar{\mathbf{f}}_b, c_b \cdot \bar{\mathbf{e}}, \bar{\sigma}_r)$ **then**

$\quad\quad\quad$ **return** $\sigma = (\bar{\mathbf{t}}, (\mathbf{u}_b, \bar{\mathbf{u}}_b, \mathbf{z}_b)_{b\in\{0,1\}})$

return $\perp$

Fig. 2. Description of lattice-based signature scheme from search Module-LWE $\mathsf{SS}_{\mathcal{R}} = \mathsf{SS}_{\mathcal{R}}[n, m, N, q, \ell, M, \mathsf{H}_1, \mathsf{H}_2]$. $\mathsf{H}_1 : \{0,1\}^* \to \mathcal{R}_q^{n\times m'}$ and $\mathsf{H}_2 : \{0,1\}^* \to \mathcal{C}^2$. We refer to Table 1 for parameter definition. All ring operations are over $\mathcal{R}_q$.

Last equality holds by independence of c_1, c_1', c_2, c_2'. Now note that it holds that

$$\begin{aligned} p_1^2 + p_2^2 - p_1^2 \cdot p_2^2 &= (p_1 - p_2)^2 + 2p_1p_2 - p_1^2p_2^2 \\ &= \underbrace{(p_1 - p_2)^2 + p_1p_2 \cdot (1 - p_1p_2)}_{\geqslant 0} + p_1p_2 \geqslant p_1p_2, \end{aligned}$$

where $(p_1 - p_2)^2 + p_1p_2 \cdot (1 - p_1p_2) \geqslant 0$ follows from the fact that $p_1, p_2 \in [0,1]$. Hence, the claim follows. □

4 Tight Signatures from Search Module-LWE

In this section, we introduce our tight signature scheme from search Module-LWE assumption in Section 4.1, and give its correctness and SUF-CMA security proof in Sections 4.2 and 4.3, respectively.

4.1 Construction

We give the construction of $\mathsf{SS}_{\mathcal{R}}$, a signature scheme based on the Search Module-LWE (sMLWE) assumption, in Figure 2. The construction can be seen as a lattice-based analogue of Chevallier-Mames signature scheme [Che05] or the generic construction from five-move identification schemes [KLP17]. Note that we do not present the scheme as a five-move identification as the aforementioned generic construction and security definitions do not apply directly to our scheme.[5]

As sketched in Section 1.1, our scheme makes use of two-fold parallel repetition on the vectors and elements $(\mathbf{u}_b, \bar{\mathbf{u}}_b, c_b, \mathbf{z}_b)$, which in a way can be viewed as a proof (with relaxed soundness) that $(\mathbf{t}, \bar{\mathbf{t}})$ are within the proximity of the row spans of the matrices $\mathbf{A}$ and $\mathbf{H}$, respectively. Additionally, we discuss some details omitted in the overview as follows:

[5] This is due to correctness error and that our secret key contains a discrete Gaussian noise.

Table 1. Table labeling the parameters used in the scheme.

Name	Value	Definition
n		Number of rows of $\mathbf{A}$ and $\mathbf{H}$
m		Number of columns of $\mathbf{A}$
b		Base integer for gadget matrix
m'	$\geqslant m + n\log_{\mathsf{b}} q$	Number of columns of $\mathbf{H}$
$\mathcal{C}$	$:= \{c \in \mathcal{R} : \|c\|_\infty = 1 \ \wedge \ \|c\|_1 = \gamma_c\}$	Challenge set for H_2
η	$:= \frac{8}{\sqrt{\pi}} q^{\frac{n}{m}} \sqrt{N\ln(2mN(1+1/\varepsilon))}$	Upper bound on $\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$
σ_{sk}	$\geqslant \sqrt{\log(2mN(1+1/\varepsilon))/\pi}$	Standard deviation for $\mathbf{e}$ in sk
$\overline{\mathbf{\Sigma}}$	$:= \sigma_x^2 \mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^T + \sigma_y^2 \mathbb{I}_N$	Covariance matrix for $\overline{\mathbf{e}}$
σ_x	$:= \sqrt{2}\eta$	
σ_y	$:= \eta\sigma_{\mathsf{sk}} N\sqrt{2m} = \sigma_x\sigma_{\mathsf{sk}} N\sqrt{m}$	
α, t	$= \omega(\sqrt{\log(mN)})$	Parameters for rejection sampling
σ_r	$:= \alpha\gamma_c\sigma_{\mathsf{sk}}\sqrt{mN}$	Standard deviation for $\mathbf{d}$
$\overline{\sigma}_r$	$:= \alpha\gamma_c\sigma_y\sqrt{2m'N}$	Standard deviation for $\overline{\mathbf{d}}$
$\beta_{\mathbf{z}}$	$:= \sigma_r\sqrt{mN}$	ℓ_2-norm bound for $\mathbf{u}_b + c_b\mathbf{t} - \mathbf{z}_b\mathbf{A}$
$\overline{\beta}_{\mathbf{z}}$	$:= \overline{\sigma}_r\sqrt{m'N}$	ℓ_2-norm bound for $\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}} - \mathbf{z}_b\mathbf{H}$
σ_{td}	$\geqslant \eta$	Standard deviation for trapdoor
q	$\geqslant 4\mathsf{b}\sigma_{\mathsf{td}}\overline{\beta}_{\mathbf{z}}\sqrt{m'm}N$	Prime modulus
M	$:= \exp\left(\frac{t}{\alpha} + \frac{2}{\alpha^2}\right)$	Rejection sampling parameter
ℓ		Max number of rejection

- The error $\overline{\mathbf{e}}$ is sampled with respect to the covariance matrix $\overline{\mathbf{\Sigma}}$, which depends on $\mathbf{e}$ in the secret key. This does not make the scheme insecure, since in our proof, we will simulate $\overline{\mathbf{t}}$ without the knowledge of $\mathbf{s}$ and $\mathbf{e}$.
- The challenge c_0, c_1 are from the challenge set $\mathcal{C}$ containing $\mathcal{R}$-elements with $\{-1, 0, 1\}$ entries and fixed ℓ_1-norm. This is standard within the Fiat-Shamir signatures and non-interactive zero-knowledge proof literature [LDK+22, LNP22].
- The signing algorithm does not immediately output the signature. It employs the rejection sampling techniques [Lyu09, Lyu12] on the values $\mathbf{f}_b$ and $\overline{\mathbf{f}}_b$ to ensure that the signature does not leak the errors $\mathbf{e}$ and $\overline{\mathbf{e}}$.

Additionally, we give the description of each parameter in the scheme with their necessary bounds in Table 1 and discuss asymptotic parameter choices below.

Asymptotic parameters. We briefly explain the bounds in Table 1.

- We require $\sigma_{\mathsf{sk}} = \max\{\sqrt{n}\log N, \sqrt{\log(2mN(1+1/\varepsilon))/\pi}\}$. The first ensures that sMLWE is hard based on worst-case assumptions [LS15]. The second term is $\eta_\varepsilon(\mathcal{R}^m)$ for applying the Gaussian tail-bound in Lemma 2.3 to bound $\|\mathbf{e}\|$.
- We require $\overline{\mathbf{\Sigma}} = \sigma_x^2\mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^T + \sigma_y^2\mathbb{I}_N$ with $\sigma_x = \sqrt{2}\eta, \sigma_y = \sigma_x\sigma_{\mathsf{sk}}N\sqrt{m}$ in order to apply Lemma 2.8 in our proof to rerandomize $\mathbf{t} = \mathbf{s}\mathbf{A} + \mathbf{e}$, which is an LWE sample with respect to $\mathbf{A}$, into LWE samples $\overline{\mathbf{t}}$ with respect to $\mathbf{H}$ without knowing the secret $\mathbf{s}$ and $\mathbf{e}$. We clarify more in the next section.
- We require $\sigma_r, \overline{\sigma}_r$ to be at least $\alpha\gamma_c$ factor more than $\|\mathbf{e}\|$ and $\|\overline{\mathbf{e}}\|$ (with overwhelming probability), respectively.
- The norm bounds $\beta_{\mathbf{z}}, \overline{\beta}_{\mathbf{z}}$ for verification need to be at least $\|\mathbf{d}\|$ and $\|\overline{\mathbf{d}}\|$ to ensure negligible correctness errors.
- The norm bound γ_c should be large enough that $|\mathcal{C}| \geqslant \binom{N}{\gamma_c} \cdot 2^{\gamma_c} \geqslant 2^{-2\kappa}$.

- α, t are set so that $e^{-t^2/2}$ is negligible and for $M = e^{t/\alpha+2/\alpha^2}$, $(1-1/M^4)^\ell$ is negligible. It suffices that $\alpha, t = \omega(\sqrt{\log(mN)})$, which is sublinear in κ.
- Finally, we require q at least $4\mathsf{b}\sigma_{\mathsf{td}}\overline{\beta}_{\mathbf{z}}\sqrt{mm'}N$ and $\frac{1}{8\sqrt{N}}q^{1-\frac{n}{m}} > 4\beta_{\mathbf{z}}$. The first bound is to ensure that our reduction can always extract using the gadget trapdoor. The second one is to ensure that the bound in Lemma 2.6 remains negligible. From each bound above, we can derive a bound on q in terms of $n, m, m', N, \alpha, \gamma_c, \sigma_{\mathsf{sk}}$ as

$$q^{1-\frac{2n}{m}} \geqslant 2^9\alpha\mathsf{b}\pi^{-1}\gamma_c\sigma_{\mathsf{sk}}(\ln(2mN)+2\kappa\ln(2))m(m')^{\frac{3}{2}}N^4 \ , \quad \text{and}$$
$$q \geqslant 32\cdot\alpha\gamma_c\sigma_{\mathsf{sk}}mN\sqrt{N}q^{\frac{n}{m}} \ .$$

  The first constraint dominates. In particular, one can set $m = n\log_{\mathsf{b}} q$ and $m' = 2n\log_{\mathsf{b}} q$, so the second bound is $8\mathsf{b}\alpha\gamma_c\sigma_{\mathsf{sk}}mN\sqrt{N}$, which is less than the following first bound

$$q \geqslant 2^9\alpha\mathsf{b}^3\pi^{-\frac{3}{2}}\gamma_c(\ln(2mN)+2\kappa\ln(2))^{\frac{3}{2}}m(m')^{\frac{3}{2}}N^4$$
$$\geqslant 2^{\frac{21}{2}}\pi^{-\frac{3}{2}}\alpha\mathsf{b}^3\gamma_c(\ln(2Nn\log_{\mathsf{b}} q)+2\kappa\ln(2))^{\frac{3}{2}}(n\log_{\mathsf{b}} q)^{\frac{5}{2}}N^4$$

  Note here that n and b are constants, and N, γ_c, α are linear (or sublinear) in κ, so the modulus size q is only polynomially large in κ.

Our signature contains $(2n+2m+3m')$ $\mathcal{R}_q$-elements, i.e., $(2n+2m+3m')N\lceil\log q\rceil$ bits. Also, we can further optimize this by modifying the signature format to contain $(\overline{\mathbf{t}}, (c_b, \mathbf{z}_b, \mathbf{f}_b, \overline{\mathbf{f}}_b)_{b\in\{0,1\}})$ since $\mathbf{u}_b, \overline{\mathbf{u}}_b$ can be recovered given these values and the public key. This counts to $(2n+m')N\lceil\log q\rceil + 2Nm\lceil\log\beta_{\mathbf{z}}\rceil + 2Nm'\lceil\log\overline{\beta}_{\mathbf{z}}\rceil$ bits. The size of $\overline{\mathbf{t}}$ dominates.

Remark 4.1 (Concrete parameters). We stress again that the parameter estimates of our scheme are only asymptotic and can be quite loose when considering concrete sizes. Here, more care is needed for concrete parameter analysis. One possible avenue is to improve on the sizes gained from the LWE rerandomization, which seems to be an artifact of the proof, e.g., assuming search MLWE on a large number of samples. Another is to use heuristic norm bounds of $\mathsf{Rot}(\mathbf{e})$ and the trapdoor $\mathbf{R}$ (which we bounded in the worst-case), e.g., as used in [MP12, LNP22, AGJ+24].

Correctness and security. We state the strong-unforgeability and correctness theorems below with the proofs in Section 4.3 and Section 4.2, respectively.

Theorem 4.2 (SUF-CMA of $\mathsf{SS}_{\mathcal{R}}$). *Let $n = n(\kappa), m = m(\kappa), N = N(\kappa), q = q(\kappa), \ell = \ell(\kappa), \sigma_{\mathsf{sk}} = \sigma_{\mathsf{sk}}(\kappa)$, and $\mathsf{H}_1, \mathsf{H}_2$ be random oracles, and $\mathsf{SS}_{\mathcal{R}} = \mathsf{SS}_{\mathcal{R}}[n, m, N, q, \ell, M, \mathsf{H}_1, \mathsf{H}_2]$. For any adversary $\mathcal{A}$ running in time $\mathsf{Time}(\mathcal{A}) = \mathsf{Time}(\mathcal{A}, \kappa)$ and making at most $\mathcal{Q}_{\mathrm{S}} = \mathcal{Q}_{\mathrm{S}}(\kappa), \mathcal{Q}_1 = \mathcal{Q}_1(\kappa), \mathcal{Q}_2 = \mathcal{Q}_2(\kappa)$ queries to oracles $\mathrm{S}, \mathsf{H}_1, \mathsf{H}_2$, respectively, there exist adversaries $\mathcal{B}$ and $\mathcal{B}'$ such that*

$$\mathsf{Adv}^{\mathsf{sufcma}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}, \kappa) + \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}', \kappa) + \delta_{\mathsf{negl}}(\kappa) \ ,$$

where $\delta_{\mathsf{negl}}(\kappa)$ denotes the additive terms in κ introduced in the proof. We note that the running time of $\mathcal{B}$ and $\mathcal{B}'$ are $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_{\mathrm{S}}\ell + \mathcal{Q}_1 + \mathcal{Q}_2)\cdot\mathsf{poly}(\kappa)$ and $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_{\mathrm{S}}\ell + \mathcal{Q}_1 + \mathcal{Q}_2)\cdot\mathsf{poly}(\kappa)$.

Our proof in the following subsection follows in two steps as sketched in the overview (modulo the 'strong unforgeability'). First, we show that the SUF-CMA security is implied *tightly* by the UF-KOA security and sMLWE assumption. Then, we show that the UF-KOA security is tightly implied by the sMLWE assumption. The techniques in Section 3 are used in the second step as discussed in Section 1.1. We remark that the $\mathsf{poly}(\kappa)$ factor in the running time reflects the costs for sampling elements (uniformly and from discrete Gaussian) and performing arithmetic operations (vector and matrix multiplications) in $\mathcal{R}, \mathcal{R}_q$.

Theorem 4.3 (Correctness of $\mathsf{SS}_{\mathcal{R}}$.). *The signature scheme $\mathsf{SS}_{\mathcal{R}}$ is correct with correctness error at most $\delta = (1-\frac{1}{M^4})^\ell + 8\ell e^{-t^2/2} + (5\ell+1)2^{-mN+1}$.*

The correctness proof follows by bounding (1) the distribution of the signing with the rejection is statistically close to alternatively sampling the values as in Lemma 2.10 (hence, the $8\ell e^{-t^2/2}$ term), (2) the probability that the signing algorithm does not output a signature in ℓ repetitions ($(1-\frac{1}{M^4})^\ell$ term), and (3) the probability that the output signature does not satisfy the verification (via the tail-bound from Lemma 2.3 resulting in the term $(5\ell+1)2^{-mN+1}$).

4.2 Correctness Proof

To show that our scheme is correct, we will show that for most signing keys (depending on the norm of the error $\mathbf{e}$) (1) the signing algorithm (over at most ℓ iterations) outputs a signature, and (2) the signature is valid.

First, due to Lemma 2.3, we know that at most 2^{-mN+1} probability, the norm of the error $\|\mathbf{e}\| \geqslant \sigma_{\mathsf{sk}}\sqrt{mN}$. Thus, we will assume from here on that $\|\mathbf{e}\| \leqslant \sigma_{\mathsf{sk}}\sqrt{mN}$.

Then, we show that the signing algorithm does not abort with high probability. In particular, we will apply the rejection sampling lemma (Lemma 2.10) on $\mathbf{f} = \mathbf{d} + c_b \cdot \mathbf{e}$ and $\overline{\mathbf{f}} = \overline{\mathbf{d}} + c_b \cdot \overline{\mathbf{e}}$, which requires that both $c_b \cdot \mathbf{e}$ and $c_b \cdot \overline{\mathbf{e}}$ have bounded norms. For the first term, we have that $\|c_b \cdot \mathbf{e}\| \leqslant \gamma_c \sigma_{\mathsf{sk}}\sqrt{mN}$. For the second term, again by Lemma 2.3, except with probability at most $2^{-m'N+1}$, $\|\overline{\mathbf{e}}\|_\leqslant \sqrt{s_1(\overline{\mathbf{\Sigma}})m'N} \leqslant \sqrt{m'N}\sqrt{\sigma_y^2 + \sigma_x^2 \|\mathsf{Rot}(\mathbf{e})\|^2}$. Therefore,

$$\begin{aligned}
\|c_b \cdot \overline{\mathbf{e}}\| &\leqslant \gamma_c \|\overline{\mathbf{e}}\| \\
&\leqslant \gamma_c \sqrt{s_1(\overline{\mathbf{\Sigma}})m'N} \\
&\leqslant \gamma_c \sqrt{m'N}\sqrt{\sigma_y^2 + \sigma_x^2 \|\mathsf{Rot}(\mathbf{e})\|^2} \\
&\leqslant \gamma_c \sqrt{m'N}\sqrt{\sigma_y^2 + \sigma_x^2 (\sqrt{N}\sigma_{\mathsf{sk}}\sqrt{mN})^2} \\
&= \gamma_c \sqrt{m'N}\sqrt{\sigma_y^2 + \sigma_x^2 \sigma_{\mathsf{sk}}^2 mN^2} \\
&\leqslant \gamma_c \sqrt{m'N}\sqrt{2\sigma_y^2} \\
&= \gamma_c \sigma_y \sqrt{2m'N}\,. \qquad (2)
\end{aligned}$$

The third inequality follows from $s_1(\overline{\mathbf{\Sigma}}) = s_1(\sigma_y^2 \mathbb{I}_N + \sigma_x^2 \mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^\top) = \sigma_y^2 + \sigma_x^2 s_1(\mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^\top) = \sigma_y^2 + \sigma_x^2 \|\mathsf{Rot}(\mathbf{e})\|^2$. The fourth inequality follows from $\|\mathsf{Rot}(\mathbf{e})\| \leqslant \sqrt{N}\|\mathbf{e}\|_\leqslant \sqrt{N}\sigma_{\mathsf{sk}}\sqrt{mN}$.[6] Then for the second-to-last inequality, we observe that by how σ_y is set in Table 1, $\sigma_y = \sigma_x \sigma_{\mathsf{sk}} N\sqrt{m} \geqslant \sigma_x\sqrt{\|\mathsf{Rot}(\mathbf{e})\|}$.

Setting $T = \gamma_c\sigma_{\mathsf{sk}}\sqrt{mN}$ and $\overline{T} = \gamma_c\sigma_y\sqrt{2m'N}$, suffices for us to apply the lemma. This means that the distribution of $\mathbf{f}_b, \overline{\mathbf{f}}_b$ and the abort (proceeding to the next iteration) is at most $8e^{-t^2/2}$ statistical distance away from sampling them from discrete gaussians with standard deviation σ_r and $\overline{\sigma}_r$, respectively, and with abort probability of $1 - 1/M$ (for each $\mathbf{f}_b, \overline{\mathbf{f}}_b$ for $b \in \{0,1\}$). This counts as at most $1 - 1/M^4$ abort probability for one iteration; for ℓ iterations, this is at most $(1-1/M^4)^\ell$ and $8\ell e^{-t^2/2}$ statistical distance.

Finally, we consider the probability that $\mathbf{f}_b = \mathbf{u}_b - (\mathbf{z}_b\mathbf{A} - c_b\mathbf{t}) \pmod q$ or $\overline{\mathbf{f}}_b = \overline{\mathbf{u}}_b - (\mathbf{z}_b\mathbf{H} - c_b\overline{\mathbf{t}}) \pmod q$ in the signatures has bounds that exceeds $\beta_{\mathbf{z}} = \sigma_r\sqrt{mN}$ and $\overline{\beta}_{\mathbf{z}} = \overline{\sigma}_r\sqrt{m'N}$, respectively. By the tail bound in Lemma 2.3, we have that $\|\mathbf{f}_b\| \leqslant \sigma_r\sqrt{mN} = \beta_{\mathbf{z}}$ and $\|\mathbf{f}_b\| \leqslant \overline{\sigma}_r\sqrt{m'N} = \overline{\beta}_{\mathbf{z}}$ for $b \in \{0,1\}$ except with probability $4 \cdot 2^{-mN+1}$. Thus, we have shown the norm bound is satisfied on $\mathbf{f}$. For the check on $\mathbf{u}_b, \overline{\mathbf{u}}_b$, we have that for $b \in \{0,1\}$

$$\begin{aligned}
\mathbf{u}_b &= \mathbf{r}_b\mathbf{A} + \mathbf{d}_b \pmod q \\
&= (\mathbf{z}_b - c_b\mathbf{s})\mathbf{A} + \mathbf{f}_b - c_b\mathbf{e} \pmod q \\
&= \mathbf{z}_b\mathbf{A} - c_b(\mathbf{s}\mathbf{A} + \mathbf{e}) + \mathbf{f}_b \pmod q \\
&= \mathbf{z}_b\mathbf{A} - c_b\mathbf{t} + \mathbf{f}_b \pmod q\,.
\end{aligned}$$

[6] This follows from the dimensions of $\mathsf{Rot}(\mathbf{e})$ being $N \times mN$ in $\mathbb{R}$.

Since we have that $\|\mathbf{u}_b - (\mathbf{z}_b\mathbf{A} - c_b\mathbf{t}) \pmod q\| = \|\mathbf{f}_b\| \leqslant \beta_{\mathbf{z}}$, this check passes. Similarly, we can see that $\overline{\mathbf{u}}_b = \mathbf{z}_b\mathbf{H} - c_b\overline{\mathbf{t}} + \overline{\mathbf{f}}_b$ and that $\|\overline{\mathbf{f}}_b\| \leqslant \overline{\beta}_{\mathbf{z}}$, meaning that the signature verifies.

Combining all the errors (and over ℓ iterations), we have that the correctness error is at most

$$\begin{aligned}\delta &= 2^{-mN+1} + \left(1 - \frac{1}{M^4}\right)^{\ell} + 8\ell e^{-t^2/2} + \ell 2^{-m'N+1} + 4\ell \cdot 2^{-mN+1} \\ &\leqslant \left(1 - \frac{1}{M^4}\right)^{\ell} + 8\ell e^{-t^2/2} + (5\ell+1)2^{-mN+1} \,.\end{aligned}$$

□

4.3 SUF-CMA Proof

We prove the security in two steps: (1) show SUF-CMA security from UF-KOA security and Search MLWE, and (2) show UF-KOA security from Search MLWE. We state the following two lemmata for the tight reductions on these two steps. Then, we give the proofs in Sections 4.4 and 4.5, respectively.

Lemma 4.4 (UF-KOA implies SUF-CMA). *Let $\mathsf{SS}_{\mathcal{R}}, n, m, N, q, \ell, \sigma_{\mathsf{sk}}$ be as in Theorem 4.2. For any adversary $\mathcal{A}$ running in time $\mathsf{Time}(\mathcal{A}) = \mathsf{Time}(\mathcal{A}, \kappa)$ and making at most $\mathcal{Q}_{\mathrm{S}} = \mathcal{Q}_{\mathrm{S}}(\kappa), \mathcal{Q}_1 = \mathcal{Q}_1(\kappa), \mathcal{Q}_2 = \mathcal{Q}_2(\kappa)$ queries to $\mathrm{S}, \mathsf{H}_1, \mathsf{H}_2$, there exists an adversary $\mathcal{B}_{\mathsf{ufkoa}}$ running in time at most $\mathsf{Time}(\mathcal{B}_{\mathsf{ufkoa}}) = \mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_{\mathrm{S}}\ell + \mathcal{Q}_1 + \mathcal{Q}_2) \cdot \mathsf{poly}(\kappa)$, making at most $\mathcal{Q}'_1 = \mathcal{Q}_1 + \mathcal{Q}_2 + 1$ and $\mathcal{Q}'_2 = \mathcal{Q}_2 + 1$ queries to H_1 and H_2, respectively, and an adversary $\mathcal{B}$ running in time $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_{\mathrm{S}}\ell + \mathcal{Q}_1 + \mathcal{Q}_2) \cdot \mathsf{poly}(\kappa)$ such that*

$$\mathsf{Adv}^{\mathsf{sufcma}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\mathsf{ufkoa}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{B}_{\mathsf{ufkoa}}, \kappa) + \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}, \kappa) + \delta^{\mathsf{sufcma}} \,.$$

Here, $\delta^{\mathsf{sufcma}} = 5 \cdot 2^{-N} + \frac{(\mathcal{Q}_{\mathrm{S}}\ell + \mathcal{Q}_2)^2}{|\mathcal{C}|^2} + (\mathcal{Q}_{\mathrm{S}} + 3)2^{-mN+1} + 2(\mathcal{Q}_1 + \mathcal{Q}_2 + \mathcal{Q}_{\mathrm{S}}\ell)\mathcal{Q}_{\mathrm{S}}\ell \cdot q^{-Nn} + 2n \cdot q^{-N(m-n+1)/2} + 8\mathcal{Q}_{\mathrm{S}}\ell e^{-t^2/2} + m'\ell\mathcal{Q}_{\mathrm{S}} \cdot 2^{-2\kappa+3}$.

Proof (Sketch). The proof goes through three main steps:

- **Simulating $(\mathbf{u}_b, \overline{\mathbf{u}}_b, c_b, \mathbf{z}_b)_{b\in\{0,1\}}$ in each signing query μ:** This is done via an argument similar to honest-verifier zero-knowledge. We sample first $c_b, \mathbf{z}_b, \mathbf{f}_b, \overline{\mathbf{f}}_b$ to be independent of the signing key. Then, compute $\mathbf{u}_b = \mathbf{z}_b\mathbf{A} + \mathbf{f}_b - c_b\mathbf{t}$ and $\overline{\mathbf{u}}_b = \mathbf{z}_b\mathbf{H} + \overline{\mathbf{f}}_b - c_b\overline{\mathbf{t}}$. Next, program $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}})$ with $\mathbf{H} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m'}$, and $\mathsf{H}_2((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu) \leftarrow (c_0, c_1)$. Finally, the rejection probability is now changed to $1 - 1/M^4$. At this point, the signing simulation does not require $\overline{\mathbf{e}}$.

  We argue the statistical distance via Lemma 2.10, and the fact that the programming goes unnoticed by the adversary is due to the high minimum entropy of $(\mathbf{u}_b)_{b\in\{0,1\}}$. We refer to games $\mathbf{G}_1$ to $\mathbf{G}_3$ for more details.
- **Simulating $(\mathbf{H}, \overline{\mathbf{t}})$ in the signatures:** They are now computed as rerandomized MLWE samples of $(\mathbf{A}, \mathbf{t})$ via smaller Gaussian noises $\mathbf{x}_i$ and y_i. In particular, $\overline{\mathbf{t}} = \mathbf{t}\mathbf{X} + \mathbf{y}$ with respect to $\mathbf{H} = \mathbf{A}\mathbf{X}$ without knowing the secret $\mathbf{s}$ and $\mathbf{e}$; here, $\mathbf{X} = (\mathbf{x}_1^\top; \ldots; \mathbf{x}_{m'}^\top) \in \mathcal{R}^{m\times m'}$ where $\mathbf{x}_i \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m, \sigma_x}$ for $i = 1, \ldots, m$ and $\mathbf{y} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^{m'}, \sigma_y}$. This step is argued via Lemma 2.8 and corresponds to $\mathbf{G}_4$ in the proof. Now, the simulation is independent of $(\mathbf{s}, \mathbf{e})$.
- **Ruling out forgeries that correlate with simulated signatures:** This step, corresponding to $\mathbf{G}_5$, is purely for strong unforgeability. Indeed, in UF-CMA, the forgery is guaranteed to not correspond to the message queried to the signing oracle. We consider the forgery where its verification makes calls to H_1 or H_2 on inputs that were programmed. We can categorize the forgery into 3 types:
  1. The query corresponds to *a simulated signature that was not output.* This cannot occur except with negligible probability because these values stay hidden and $(\mathbf{u}_b)_{b\in\{0,1\}}$ has high minimum entropy.
  2. The query corresponds to *an input with H_2 programmed, and also to a simulated signature given to the adversary.* Since the query to H_2 (containing $((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu)$) is identical for the simulated signature and the forgery, only the $(\mathbf{z}_b)_{b\in\{0,1\}}$ part can differ. Inspecting the verification equation allows us to argue that this does not occur because $\mathbf{A}$ is injective for small errors (e.g., via Lemma 2.6).

3. The query corresponds to *an input with* H_1 *programmed but not* H_2 *programmed.* In this case, the simulated signature and the forgery share $(\mathbf{u}_b)_{b\in\{0,1\}}$ and μ, but with overwhelming probability, the challenges c_b, c'_b differ. Then, we can derive $(c_b - c'_b)\mathbf{t} \approx (\mathbf{z}_b - \mathbf{z}'_b)\mathbf{A}$ where $\mathbf{z}_b$ and $\mathbf{z}'_b$ are contained in the simulated signature and the forgery, respectively. Then, we can recover $\mathbf{e}$ and $\mathbf{s}$ accordingly, which breaks sMLWE assumption.

Lemma 4.5 (Search-MLWE implies UF-KOA). *Let* $\mathsf{SS}_{\mathcal{R}}, n, m, N, q, \ell, \sigma_{\mathsf{sk}}$ *be as in Theorem 4.2. For any adversary* $\mathcal{A}$ *running in time* $\mathsf{Time}(\mathcal{A}) = \mathsf{Time}(\mathcal{A}, \kappa)$ *and making at most* $\mathcal{Q}_1 = \mathcal{Q}_1(\kappa), \mathcal{Q}_2 = \mathcal{Q}_2(\kappa)$ *queries to* $\mathsf{H}_1, \mathsf{H}_2$*, there exists an adversary* $\mathcal{B}$ *running in time* $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_1 + \mathcal{Q}_2) \cdot \mathsf{poly}(\kappa)$ *such that*

$$\mathsf{Adv}^{\mathsf{ufkoa}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}, \kappa) + (\mathcal{Q}_1 + 1)(2^{-N+2} + 2^{-2\kappa+2}) + \frac{6\mathcal{Q}_2}{|\mathcal{C}|} \ .$$

Proof (Sketch). The proof follows the sketch made in Section 1.1. We first sample the matrices $\mathbf{H}$ (outputs of the random oracle H_1) with trapdoors $\mathbf{R}$ (via Lemma 2.9). Then, we proceed in a sequence of games analogous to the experiments $\mathsf{Real}, \mathsf{Ideal}, \mathsf{Ideal}'$ in Section 3. These changes respectively correspond to (1) relaxing the winning condition of $\mathcal{A}$ from outputting a forgery to making a query where the sampled (c_0, c_1) would allow $\mathcal{A}$ to output the forgery, (2) changing the winning condition to whether the game can sample a fork $(c_0, c_1), (c'_0, c'_1)$ at some point in its execution (justified by Theorem 3.2), and (3) limiting the running time of the simulation of H_2 over all queries (justified by Theorem 3.6). Finally, we efficiently simulate these introduced checks using the sampled trapdoor of $\mathbf{H}$. We again refer to the formal proof in Section 4.5.

4.4 Proof of Lemma 4.4

Let $\mathcal{A}$ be as in the lemma statement. We also make the following simplifying assumptions: (1) $\mathcal{A}$ makes queries to $\mathsf{H}_1, \mathsf{H}_2$ that will be made during forgery verification in advance, and (2) when queries to H_2 of the form $((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu)$ is made, the query $((\mathbf{u}_b)_{b\in\{0,1\}}, \mu)$ has been made before. These assumptions (implicitly) increase the query count to H_1 by $\mathcal{Q}_2 + 1$ and to H_2 by 1. Also, we denote the forgery of $\mathcal{A}$ as $(\mu^*, \sigma^* = (\overline{\mathbf{t}}^*, (\mathbf{u}^*_b, \overline{\mathbf{u}}^*_b, \mathbf{z}^*_b)_{b\in\{0,1\}}))$. Consider the following sequence of games $\mathbf{G}_i$ for $i = 0, \ldots, 5$.

Game $\mathbf{G}_0$ (Original Game): Original SUF-CMA game.

Game $\mathbf{G}_1$ (Abort if to-be-programmed RO query has been made): In each S query μ, if the query $((\mathbf{u}_b)_{b\in\{0,1\}}, \mu)$ to H_1 or $((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu)$ to H_2, that the game makes when honestly signing, was made before (either within previous S query or by $\mathcal{A}$), the game aborts. This game is introduced to make sure that the game can later on program the random oracle without the adversary noticing.

To argue the change in advantage, observe that the view of $\mathcal{A}$ stays the same given that the game does not abort. Hence, we analyze the probability that this abort could occur. In particular, this will be due to the high-min-entropy of $\mathbf{u}_b$; which we show below.

Lemma 4.6. *Let* n, m, N*, and* $\mathcal{R}$ *be as in the scheme. Then, for at least* $1 - 2^{-N}$ *fraction of* $\mathbf{A} \in \mathcal{R}_q^{n\times m}$*, for any* $\mathbf{v}$*,* $\mathsf{Pr}_{\mathbf{s} \leftarrow\!\$\, \mathcal{R}_q^n}[\mathbf{sA} = \mathbf{v}] \leqslant q^{-nN}$*.*

Proof (of Lemma 4.6). Recall that from Lemma 2.6 and how we set q in Table 1, we have that except with probability at most 2^{-N}, $\mathbf{A}$ is such that the map $(\mathbf{s}, \mathbf{e}) \mapsto \mathbf{sA} + \mathbf{e}$ is injective for small $\mathbf{e}$. Therefore, $\mathbf{sA}$ preserves the entropy of $\mathbf{s}$, meaning for any $\mathbf{v}$, either $\mathbf{v}$ is not in the span of $\mathbf{A}$, or $\mathbf{sA} = \mathbf{v}$ with probability q^{-Nn}. □

Therefore, by a union bound over all pairs of (a) queries to H_1 or H_2 by $\mathcal{A}$ and the game (at most $\mathcal{Q}_1 + \mathcal{Q}_2 + \mathcal{Q}_{\mathrm{S}}\ell$) and (2) the queries made during signing by the game (at most $\mathcal{Q}_{\mathrm{S}}\ell$), and also the bad event that $\mathbf{A}$ is not full-rank, we have

$$\mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] + 2^{-N} + (\mathcal{Q}_1 + \mathcal{Q}_2 + \mathcal{Q}_{\mathrm{S}}\ell)\mathcal{Q}_{\mathrm{S}}\ell \cdot q^{-nN} \ .$$

Game $\mathbf{G}_2$ (Program H_1 and H_2 to random): For each iteration $i \in [\ell]$ in each S query μ, the game simulates the signing as follows:

- Sample $\mathbf{H} \leftarrow_{\$} \mathcal{R}_q^{n \times m}$ and $(c_0, c_1) \leftarrow_{\$} \mathcal{C}^2$.
- Sample $\overline{\mathbf{e}} \leftarrow_{\$} \mathcal{D}_{\mathcal{R}^m, \overline{\sigma_{\mathsf{sk}}}}$ and set $\overline{\mathbf{t}} \leftarrow \mathbf{sH} + \overline{\mathbf{e}}$.
- For $b \in \{0,1\}$:
  - Sample $\mathbf{r}_b \leftarrow_{\$} \mathcal{R}_q^n, \mathbf{d}_b \leftarrow_{\$} \mathcal{D}_{\mathcal{R}^m, \sigma_r}$ and $\overline{\mathbf{d}}_b \leftarrow_{\$} \mathcal{D}_{\mathcal{R}^m, \overline{\sigma}_r}$.
  - Sample $\mathbf{z}_b \leftarrow \mathbf{r}_b + c_b \cdot \mathbf{s}, \mathbf{f}_b \leftarrow c_b \cdot \mathbf{e} + \mathbf{d}_b$, and $\overline{\mathbf{f}}_b \leftarrow c_b \cdot \overline{\mathbf{e}} + \overline{\mathbf{d}}_b$.
  - Set $\mathbf{u}_b \leftarrow \mathbf{z}_b\mathbf{A} + \mathbf{f}_b - c_b\mathbf{t} \pmod q, \overline{\mathbf{u}}_b \leftarrow \mathbf{z}_b\mathbf{H} + \overline{\mathbf{f}}_b - c_b\overline{\mathbf{t}} \pmod q$.
- Program $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}}, \mu) \leftarrow \mathbf{H}$ and $\mathsf{H}_2((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu) \leftarrow (c_0, c_1)$; if one of these queries have been made before, *the game aborts.*
- For $b \in \{0,1\}$, run $\mathsf{Rej}(\mathbf{f}_b, c_b\mathbf{e}, \sigma_r)$ and $\mathsf{Rej}(\overline{\mathbf{f}}_b, c_b\overline{\mathbf{e}}, \overline{\sigma}_r)$ (which is a probabilistic procedure, with the rejection sampling algorithm defined in Figure 2). If none of them reject, return $\sigma = (\overline{\mathbf{t}}, (\mathbf{u}_b, \overline{\mathbf{u}}_b, \mathbf{z}_b)_{b\in\{0,1\}})$. Otherwise, continue to the next iteration.

This game simply changes the ordering of how $(\mathbf{u}_b, \overline{\mathbf{u}}_b, \mathbf{z}_b, \mathbf{f}_b, \overline{\mathbf{f}}_b)_{b\in\{0,1\}}$ are computed and reprograms the random oracles H_1, H_2. In this game, we note that the computation of $\mathbf{u}_b, \overline{\mathbf{u}}_b$ only depends on $\mathbf{z}_b, \mathbf{f}_b, \overline{\mathbf{f}}_b$ only.

This however does not change the distribution of $\mathbf{u}_b, \overline{\mathbf{u}}_b$, since we can verify that for $b \in \{0,1\}$

$$\begin{aligned}\mathbf{u}_b &= \mathbf{r}_b\mathbf{A} + \mathbf{d}_b \pmod q \\ &= (\mathbf{z}_b - c_b\mathbf{s})\mathbf{A} + \mathbf{f}_b - c_b\mathbf{e} \pmod q \\ &= \mathbf{z}_b\mathbf{A} - c_b(\mathbf{sA} + \mathbf{e}) + \mathbf{f}_b \pmod q \\ &= \mathbf{z}_b\mathbf{A} + \mathbf{f}_b - c_b\mathbf{t} \pmod q\ .\end{aligned}$$

Similarly, we can show that $\overline{\mathbf{u}}_b = \mathbf{z}_b\mathbf{H} + \overline{\mathbf{f}}_b - c_b\overline{\mathbf{t}} \pmod q$. Moreover, because of the abort introduced in $\mathbf{G}_1$ and that $\mathbf{H}$ and c_0, c_1 are still uniformly sampled, the success probability of $\mathcal{A}$ remains the same as in $\mathbf{G}_1$, i.e., $\mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]$.

Game $\mathbf{G}_3$ (Apply rejection sampling): For each iteration $i \in [\ell]$ in each S query, the game changes how $\mathbf{u}_b, \overline{\mathbf{u}}_b$ and $\mathbf{z}_b$ are computed along with the abort:

- For $b \in \{0,1\}$:
  - Sample $\mathbf{z}_b \leftarrow_{\$} \mathcal{R}_q^n, \mathbf{f}_b \leftarrow_{\$} \mathcal{D}_{\mathcal{R}^m, \sigma_r}, \overline{\mathbf{f}}_b \leftarrow_{\$} \mathcal{D}_{\mathcal{R}^m, \overline{\sigma}_r}$.
  - Set $\mathbf{u}_b \leftarrow \mathbf{z}_b\mathbf{A} + \mathbf{f}_b - c_b\mathbf{t}, \overline{\mathbf{u}}_b \leftarrow \mathbf{z}_b\mathbf{H} + \overline{\mathbf{f}}_b - c_b\overline{\mathbf{t}}$.
- Program $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}}, \mu) \leftarrow \mathbf{H}$ and $\mathsf{H}_2((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu) \leftarrow (c_0, c_1)$; if one of these queries have been made before, *the game aborts.*
- With probability $\frac{1}{M^4}$ return $\sigma = (\overline{\mathbf{t}}, (\mathbf{u}_b, \overline{\mathbf{u}}_b, \mathbf{z}_b)_{b\in\{0,1\}})$. Otherwise, continue to the next iteration.

At this point, we do not rely on $\mathbf{s}, \mathbf{e}$ and $\overline{\mathbf{e}}$ to compute $\mathbf{z}_b, \mathbf{f}_b, \overline{\mathbf{f}}_b$ at all during the signing. This will allow us to change how $\overline{\mathbf{e}}$ is computed in the next game in order to make the simulation independent of $\mathbf{z}_b$ and $\mathbf{f}_b$, but only the public key $(\mathbf{A}, \mathbf{t})$.

To argue the advantage change of $\mathcal{A}$, we first notice that (in the dashed boxes) the only changes from $\mathbf{G}_2$ are (a) how $\mathbf{z}_b$ is sampled,and (b) how $\mathbf{f}_b, \overline{\mathbf{f}}_b$ are sampled along with the changed the rejection probability. For (a), we observe that $\mathbf{z}_b = \mathbf{r}_b + c_b\mathbf{s}$ and $\mathbf{r}_b$ is uniformly random, so $\mathbf{z}_b$ is uniformly random.

For (b), we will apply the rejection sampling lemma (Lemma 2.10) on $\mathbf{f}_b, \overline{\mathbf{f}}_b$ for $b \in \{0,1\}$. In particular, the lemma allows us to directly argue that the distribution of $\mathbf{f}_b, \overline{\mathbf{f}}_b$ along with the aborts in $\mathbf{G}_2$ and $\mathbf{G}_3$ are statistically indistinguishable. To apply the lemma to $\mathbf{f}_b$, we require that $\|c_b \cdot \mathbf{e}\| \leqslant \gamma_c \|\mathbf{e}\|$ is bounded by some T, and similarly for $\overline{\mathbf{f}}_b$ with $\|c_b \cdot \overline{\mathbf{e}}\| \leqslant \gamma_c \|\overline{\mathbf{e}}\|$ by some $\overline{T}$.

To do this, we use Lemma 2.3 to bound the norm of $\mathbf{e}$ and $\overline{\mathbf{e}}$ which are gaussians of standard deviation $\sigma_{\mathsf{sk}} \geqslant \eta_\varepsilon(\mathcal{R}^m)$ and covariance matrix $\overline{\mathbf{\Sigma}}$ with $\sqrt{s_m(\overline{\mathbf{\Sigma}})} \geqslant \sigma_x \geqslant \eta_\varepsilon(\mathcal{R}^m)$. This gives us that except with

probability 2^{-mN+1}, $\|\mathbf{e}\| \leqslant \sigma_{\mathsf{sk}}\sqrt{mN}$ and $\|\overline{\mathbf{e}}\| \leqslant \sqrt{m'Ns_1(\overline{\mathbf{\Sigma}})}$. By the union bound, this means that except with probability at most $(\mathcal{Q}_{\mathrm{S}}+1)2^{-mN+1}$, the errors $\mathbf{e}$ and $\overline{\mathbf{e}}$ in all sessions are bounded.

Setting $T = \gamma_c\sigma_{\mathsf{sk}}\sqrt{mN}$ and $\overline{T} = \sqrt{2}\sigma_y \geqslant \gamma_c\sqrt{m'Ns_1(\overline{\mathbf{\Sigma}})}$ (by how σ_y is set in Table 1 and the same bounds in Equation (2)), we have that $\sigma_r \geqslant \alpha T$ and $\overline{\sigma}_r \geqslant \alpha\overline{T}$. Then, by the rejection sampling lemma (Lemma 2.10) and conditioning on $\mathbf{e}$ and $\overline{\mathbf{e}}$ being bounded, we have that the statistical distance between each S simulation in $\mathbf{G}_2$ and $\mathbf{G}_3$ is at most $8\ell\exp(-t^2/2)$ with $t = \omega(\sqrt{\log mN})$. Then, we can bound the advantage of the adversary in this game as

$$\mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] + (\mathcal{Q}_{\mathrm{S}}+1)2^{-mN+1} + 8\mathcal{Q}_{\mathrm{S}}\ell e^{-t^2/2} .$$

Game $\mathbf{G}_4$ (Compute $(\mathbf{H}, \overline{\mathbf{t}})$ as rerandomized $(\mathbf{A}, \mathbf{t})$): For each iteration $i \in [\ell]$ in each S query, the game computes $\mathbf{H}$ and $\overline{\mathbf{t}}$ as follows:

- Sample $\mathbf{X} = (\mathbf{x}_1^\top; \ldots; \mathbf{x}_{m'}^\top) \in \mathcal{R}^{m\times m'}$ where $\mathbf{x}_i \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_x}$ for $i = 1, \ldots, m$ and $\mathbf{y} \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^{m'},\sigma_y}$.
- Set $\overline{\mathbf{t}} \leftarrow \mathbf{t}\mathbf{X} + \mathbf{y}$, and set $\mathbf{H} \leftarrow \mathbf{A}\mathbf{X}$.

At this point, the signing simulation only depends on the public key $(\mathbf{A}, \mathbf{t})$ which allows us to set up for the reductions in the subsequent games.

Observe that we can write $(\mathbf{H}, \overline{\mathbf{t}})$ as

$$\mathbf{H} = [\mathbf{h}_1^\top \mid \ldots \mid \mathbf{h}_{m'}^\top] \in \mathcal{R}_q^{n\times m'} , \text{ and } \overline{\mathbf{t}} = (\overline{t}_1, \ldots, \overline{t}_{m'}) \in \mathcal{R}_q^{m'} .$$

In $\mathbf{G}_3$, each $\mathbf{h}_i$ is uniformly random and $\overline{t}_i$ is its corresponding LWE sample with respect to $\mathbf{s}$ and the error $\overline{e}_i \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R},\overline{\mathbf{\Sigma}}}$. In $\mathbf{G}_4$, each $\mathbf{h}_i$ is rerandomized from $\mathbf{A}$ as $\mathbf{h}_i = \mathbf{A}\mathbf{x}_i^\top$ and $\overline{t}_i$ from $\mathbf{t}$ as $\mathbf{t}\mathbf{x}_i^\top + y_i$ with gaussian $\mathbf{x}_i \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_x}$ additional smoothing noise y_i for $i \in [m']$.

To argue that the change to advantage of $\mathcal{A}$ is negligible, we will apply Lemma 2.8 to argue the statistical distance of at most $4\varepsilon/(1-\varepsilon) \leqslant 2^{-2\kappa+3}$ per each sample. In particular, we need to argue that (a) $\mathbf{A}$ is primitive (i.e., surjective as a map $\mathcal{R}_q^m \to \mathcal{R}_q^n$), and (b) $\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leqslant (\sigma_x^{-2} + \sigma_y^{-2}\|\mathsf{Rot}(\mathbf{e})\|)^{-1/2}$.

For (a), we refer to Lemma 2.2, which for our setting with $q = 5 \pmod 8$ states that the probability that $\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}$ is surjective is at least $1 - 2n\cdot q^{-N(m-n+1)/2}$. For (b), let $\eta = \frac{8}{\sqrt{\pi}}q^{\frac{n}{m}}\sqrt{N\ln(2mN(1+1/\varepsilon))}$ and observe that setting $\sigma_x \approx \sqrt{2}\eta$ and $\sigma_y \approx \sigma_x\sigma_{\mathsf{sk}}\sqrt{mN} \geqslant \sigma_x\sqrt{N}(\sqrt{mN}\sigma_{\mathsf{sk}})$. Note that $\|\mathsf{Rot}(\mathbf{e})\| \leqslant \sqrt{N}\|\mathsf{Rot}(\mathbf{e})\| \leqslant N\sqrt{m}\sigma_{\mathsf{sk}}$, so $\sigma_y \geqslant \sigma_x\|\mathsf{Rot}(\mathbf{e})\|$ where the last inequality holds except with probability at most 2^{-mN+1} due to Lemma 2.3. Then, we see that

$$\frac{1}{\sigma_x^2} + \frac{\|\mathsf{Rot}(\mathbf{e})\|^2}{\sigma_y^2} \leqslant \frac{1}{\sigma_x^2} + \frac{1}{\sigma_x^2} = 2\sigma_x^{-2} \approx \eta^{-2} .$$

Therefore, $\eta \leqslant (\sigma_x^{-2} + \sigma_y^{-2}\|\mathsf{Rot}(\mathbf{e})\|^2)^{-1/2}$, proving (b). Then, with Lemmas 2.4 and 2.5, we have that $\eta \geqslant \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$ except with probability 2^{-N}.

Therefore, applying combining the distance over at most $m'\cdot\ell\cdot\mathcal{Q}_{\mathrm{S}}$ samples, we have that

$$\mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] + 2n\cdot q^{-N(m-n+1)/2} + 2^{-mN+1} + m'\ell\mathcal{Q}_{\mathrm{S}}\cdot 2^{-2\kappa+3} + 2^{-N} .$$

Game $\mathbf{G}_5$ (Rule out forgery from programmed RO): The game aborts if $\mathcal{A}$ wins the game $\mathbf{G}_4$ and the forgery $(\mu^*, \sigma^* = (\overline{\mathbf{t}}^*, (\mathbf{u}_b^*, \overline{\mathbf{u}}_b^*, \mathbf{z}_b^*)_{b\in\{0,1\}}))$ of $\mathcal{A}$ corresponds to $((\mathbf{u}_b^*)_{b\in\{0,1\}}, \mu^*)$ where $\mathbf{H}^* = \mathsf{H}_1((\mathbf{u}_b^*)_{b\in\{0,1\}}, \mu^*)$ or $\mathsf{H}_2((\mathbf{u}_b^*, \cdot)_{b\in\{0,1\}}, \cdot, \mu^*)$ is programmed. The main idea of introducing this game is to ensure that the forgery that $\mathcal{A}$ outputs does not relate to a simulated one, which will allow us to construct a reduction to UF-KOA.

We claim that this introduced abort only occurs with negligible probability. To see this, we consider two possible cases:

Case (1): $((\mathbf{u}_b^*)_{b\in\{0,1\}}, \mu^*)$ appears in a signature from a signing query; for convenience, we denote that signature as $(\overline{\mathbf{t}}', (\mathbf{u}_b^*, \overline{\mathbf{u}}_b', \mathbf{z}_b')_{b\in\{0,1\}})$ of μ^*, highlighting the identical values. Here, there are two possibilities:

(1.a) $(\bar{\mathbf{t}}', (\bar{\mathbf{u}}'_b)_{b\in\{0,1\}}) = (\bar{\mathbf{t}}^*, (\bar{\mathbf{u}}^*_b)_{b\in\{0,1\}})$ but $(\mathbf{z}'_b)_{b\in\{0,1\}} \neq (\mathbf{z}^*_b)_{b\in\{0,1\}}$ or
(1.b) $(\bar{\mathbf{t}}', (\bar{\mathbf{u}}'_b)_{b\in\{0,1\}}) \neq (\bar{\mathbf{t}}^*, (\bar{\mathbf{u}}^*_b)_{b\in\{0,1\}})$.
In case **(1.a)**, let $(c^*_0, c^*_1) = \mathsf{H}_2((\mathbf{u}^*_b, \bar{\mathbf{u}}^*_b)_{b\in\{0,1\}}, \mathbf{H}^*, \bar{\mathbf{t}}^*, \mu^*)$, and the validity of the signature implies that for $b \in \{0,1\}$, $\|\mathbf{u}^*_b + c^*_b\mathbf{t} - \mathbf{z}^*_b\mathbf{A}\|, \|\mathbf{u}^*_b + c^*_b\mathbf{t} - \mathbf{z}'_b\mathbf{A}\| \leqslant \beta$. Accordingly, we have that

$$\left\|(\mathbf{z}^*_b - \mathbf{z}'_b)\mathbf{A}\right\| \leqslant 2\beta \text{ for } b \in \{0,1\} .$$

In this case, we can apply Lemma 2.6 to argue that for overwhelming probability of sampling $\mathbf{A}$, this event cannot occur due to the norm bound being small and $\mathbf{z}^*_b \neq \mathbf{z}'_b$. In particular, we have that $\mathsf{Pr}[\textbf{Case (1)}] \leqslant 2^{-N}$ with how we set q in Table 1.

In case **(1.b)**, let $(c^*_0, c^*_1) = \mathsf{H}_2((\mathbf{u}^*_b, \bar{\mathbf{u}}^*_b)_{b\in\{0,1\}}, \mathbf{H}^*, \bar{\mathbf{t}}^*, \mu^*)$, and $(c'_0, c'_1) = \mathsf{H}_2((\mathbf{u}^*_b, \bar{\mathbf{u}}'_b)_{b\in\{0,1\}}, \mathbf{H}^*, \bar{\mathbf{t}}', \mu^*)$. Then, if $(c'_0, c'_1) \neq (c^*_0, c^*_1)$ (which occurs except with probability $\binom{\mathcal{Q}_\mathrm{S}\ell+\mathcal{Q}_2}{2}/|\mathcal{C}|^2$), we let $b \in \{0,1\}$ be such that $c'_b \neq c^*_b$; the validity of the signature implies that there exists $\mathbf{f}^*_b$ and $\mathbf{f}'_b$ with norms at most $\beta_\mathbf{z}$ such that they are efficiently computable from the signature and the simulated (but not output) signature, and

$$\mathbf{f}^*_b = \mathbf{u}_b + c^*_b\mathbf{t} - \mathbf{z}^*_b\mathbf{A} , \qquad \mathbf{f}'_b = \mathbf{u}_b + c'_b\mathbf{t} - \mathbf{z}'_b\mathbf{A}$$

Then, we have that

$$\begin{aligned} \mathbf{f}^*_b - \mathbf{f}'_b &= (c^*_b - c'_b)\mathbf{t} - (\mathbf{z}^*_b - \mathbf{z}'_b)\mathbf{A} \\ \mathbf{f}^*_b - \mathbf{f}'_b &= (c^*_b - c'_b)(\mathbf{s}\mathbf{A} + \mathbf{e}) - (\mathbf{z}^*_b - \mathbf{z}'_b)\mathbf{A} \\ \mathbf{f}^*_b - \mathbf{f}'_b - (c^*_b - c'_b)\mathbf{e} &= ((c^*_b - c'_b)\mathbf{s} - (\mathbf{z}^*_b - \mathbf{z}'_b))\mathbf{A} . \end{aligned}$$

Note that $\|\mathbf{f}^*_b - \mathbf{f}'_b - (c^*_b - c'_b)\mathbf{e}\| \leqslant 2\beta_\mathbf{z} + 2\gamma_c \|\mathbf{e}\|$.
Similar to the above case, we will apply Lemma 2.6 and Lemma 2.3 (the latter ensuring that $\|\mathbf{e}\| \leqslant \sigma_{\mathsf{sk}}\sqrt{mN}$, so $2\gamma_c \|\mathbf{e}\| \leqslant 2\beta_\mathbf{z}$) to argue that with overwhelming probability $1 - 2^{-N}$ over $\mathbf{A}$, $(c^*_b - c'_b)\mathbf{s} - (\mathbf{z}^*_b - \mathbf{z}'_b) = \mathbf{0}$ and $\mathbf{f}^*_b - \mathbf{f}'_b - (c^*_b - c'_b)\mathbf{e} = \mathbf{0}$. Then, one can efficiently compute $\mathbf{e}$ and $\mathbf{s}$ as follows:

- Compute $\mathbf{e} = (c^*_b - c'_b)^{-1}(\mathbf{f}^*_b - \mathbf{f}'_b)$ in the 'field' $\mathcal{K} = \mathbb{Q}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$. Note that this works because $\mathbf{f}^*_b, \mathbf{f}'_b, \mathbf{e}$, and $(c^*_b - c'_b)$ are all small norm elements/vectors in $\mathcal{R} \subseteq K$; therefore, the addition and multiplications performed are over the base ring $\mathcal{R}$ not $\mathcal{R}_q$.
- Compute $\mathbf{s}$ from $\mathbf{t} - \mathbf{e} = \mathbf{s}\mathbf{A}$, by solving the linear equations over $\mathcal{R}_q$. Due to the injectivity of $\mathbf{A}$, there exists only one such $\mathbf{s}$.

Therefore, the LWE secret can be efficiently extracted, so the probability of this case occurring can be bounded via a reduction to sMLWE. The reduction would take as input $(\mathbf{A}, \mathbf{t})$, identify that case **(1.b)** occurs, and apply the above extraction strategy. The running time of $\mathcal{B}$ is roughly $\mathsf{Time}(\mathcal{A}) + (\mathcal{Q}_\mathrm{S}\ell + \mathcal{Q}_1 + \mathcal{Q}_2)\mathsf{poly}(\kappa)$.

We can then see that via the union bound

$$\begin{aligned} \mathsf{Pr}[\textbf{Case (1.b)}] &\leqslant \mathsf{Pr}[(c'_0, c'_1) = (c^*_0, c^*_1)] + \mathsf{Pr}[\|\mathbf{e}\| > \sigma_{\mathsf{sk}}\sqrt{mN}] \\ &\quad + \mathsf{Pr}[(c'_0, c'_1) \neq (c^*_0, c^*_1) \ \wedge \ (c^*_b - c'_b)\mathbf{s} \neq (\mathbf{z}^*_b - \mathbf{z}'_b)] \\ &\quad + \mathsf{Pr}[(c'_0, c'_1) \neq (c^*_0, c^*_1) \ \wedge \ (c^*_b - c'_b)\mathbf{s} = (\mathbf{z}^*_b - \mathbf{z}'_b)] \\ &\leqslant \frac{(\mathcal{Q}_\mathrm{S}\ell + \mathcal{Q}_2)^2}{|\mathcal{C}|^2} + 2^{-mN+1} + 2^{-N} + \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}, \kappa) . \end{aligned}$$

Case (2): $((\mathbf{u}^*_b)_{b\in\{0,1\}}, \mu^*)$ was not given to the adversary at any signing query. This means that there exists a H_1 or H_2 query containing $((\mathbf{u}^*_b)_{b\in\{0,1\}}, \mu^*)$ that the output was programmed but these RO inputs were not revealed through the signing sessions.
We bound the probability of this case occurring in the same manner as the argument from $\mathbf{G}_0$ to $\mathbf{G}_1$ by considering the minimum entropy of the value $(\mathbf{u}^*_b)_{b\in\{0,1\}}$. Note that $\mathbf{z}_b$ for $b \in \{0,1\}$ are independently

sampled from the view of the adversary and the signing oracle rejection sampling are determined independently with probability $(1-1/M)^4$. Then, via Lemma 4.6, except with probability at most 2^{-N} over $\mathbf{A}$, a non-revealed $(\mathbf{u}_b^*)_{b\in\{0,1\}}$ can be guessed by $\mathcal{A}$ with probability at most q^{-nN}.
Applying the union bound over all pairs of (a) H_1 or H_2 queries $(\mathcal{Q}_1+\mathcal{Q}_2)$ and (b) the number of programming attempts $\mathcal{Q}_S\cdot\ell$, we have $\mathsf{Pr}[\textbf{Case (2)}]\leqslant 2^{-N}+(\mathcal{Q}_1+\mathcal{Q}_2)\mathcal{Q}_S\ell q^{-nN}$.

Therefore, combining all the identified cases, we have that

$$\mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}}\Rightarrow 1]\leqslant \mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}}\Rightarrow 1]+3\cdot 2^{-N}+\frac{(\mathcal{Q}_S\ell+\mathcal{Q}_2)^2}{|\mathcal{C}|^2}$$
$$+2^{-mN+1}+\mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B},\kappa)+(\mathcal{Q}_1+\mathcal{Q}_2)\mathcal{Q}_S\ell q^{-nN}\ .$$

Reduction to UF-KOA. Finally, we bound the probability of $\mathcal{A}$ winning in game $\mathbf{G}_5$ by considering the following reduction $\mathcal{B}_{\mathsf{ufkoa}}$ taking as input the verification key $\mathsf{pk}=(\mathbf{A},\mathbf{t})$, runs $\mathcal{A}(\mathsf{pk})$ while simulating the oracles as follows:

Random oracle $\mathsf{H}_1,\mathsf{H}_2$: If the query does not correspond to an input string str where $\mathsf{H}_1(\text{str})$ or $\mathsf{H}_2(\text{str})$ *is not programmed*, forward the query to its own H_1' or H_2' oracles accordingly, and return the output to $\mathcal{A}$.

Signing oracle S: On input μ, do the following for $i\in[\ell]$:

- Sample $\mathbf{X}=(\mathbf{x}_1;\ldots;\mathbf{x}_m)\in\mathcal{R}^{m\times m}$ where $\mathbf{x}_i\leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_x}$ for $i=1,\ldots,m$ and $\mathbf{y}\leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_y}$.
- Set $\overline{\mathbf{t}}\leftarrow\mathbf{t}\mathbf{X}^\top+\mathbf{y}$, Moreover, set $\mathbf{H}\leftarrow\mathbf{A}\mathbf{X}^\top$.
- For $b\in\{0,1\}$:
  - Sample $\mathbf{z}_b\leftarrow\!\!\$\ \mathcal{R}_q^n,\mathbf{f}_b\leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma_r},\overline{\mathbf{f}}_b\leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R}^m,\overline{\sigma}_r}$.
  - Set $\mathbf{u}_b\leftarrow\mathbf{z}_b\mathbf{A}+\mathbf{f}_b-c_b\mathbf{t},\overline{\mathbf{u}}_b\leftarrow\mathbf{z}_b\mathbf{H}+\overline{\mathbf{f}}_b-c_b\overline{\mathbf{t}}$.
- Program $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}},\mu)\leftarrow\mathbf{H}$ and $\mathsf{H}_2((\mathbf{u}_b,\overline{\mathbf{u}}_b)_{b\in\{0,1\}},\mathbf{H},\overline{\mathbf{t}},\mu)\leftarrow(c_0,c_1)$; if one of these queries have been made before, *the game aborts.*
- With probability $1/M^4$ return $\sigma=(\overline{\mathbf{t}},(\mathbf{u}_b,\overline{\mathbf{u}}_b,\mathbf{z}_b)_{b\in\{0,1\}})$. Otherwise, continue to the next iteration.

When $\mathcal{A}$ returns the forgery $(\mu^*,\sigma^*=(\mathbf{u}^*,\overline{\mathbf{t}}^*,\overline{\mathbf{u}}^*,\mathbf{z}^*))$ and the forgery does not correspond to a query to H_1 or H_2 that was programmed, then $\mathcal{B}$ forwards this forgery to its game. It is easy to see that by the change in game $\mathbf{G}_5$, we have

$$\mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}}\Rightarrow 1]\leqslant\mathsf{Adv}^{\mathsf{ufkoa}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{B}_{\mathsf{ufkoa}},\kappa)\ .$$

Note that the running time of $\mathcal{B}_{\mathsf{ufkoa}}$ is $\mathsf{Time}(\mathcal{B}_{\mathsf{ufkoa}})\approx\mathsf{Time}(\mathcal{A})+(\mathcal{Q}_S\ell+\mathcal{Q}_1+\mathcal{Q}_2)\cdot\mathsf{poly}(\kappa)$. The query counts $\mathcal{Q}_1'=\mathcal{Q}_1+\mathcal{Q}_2+1,\mathcal{Q}_2'=\mathcal{Q}_2+1$ of $\mathcal{B}_{\mathsf{ufkoa}}$ follow from the assumption we made at the start of the game.

Finally, combining the advantage changes, we claim the bound in the lemma statement as

$$\mathsf{Adv}^{\mathsf{sufcma}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{A},\kappa)\leqslant\mathsf{Adv}^{\mathsf{ufkoa}}_{\mathsf{SS}_{\mathcal{R}}}(\mathcal{B}_{\mathsf{ufkoa}},\kappa)+\mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B},\kappa)+\delta^{\mathsf{sufcma}}\ .$$

Here, $\delta^{\mathsf{sufcma}}=5\cdot 2^{-N}+\frac{(\mathcal{Q}_S\ell+\mathcal{Q}_2)^2}{|\mathcal{C}|^2}+(\mathcal{Q}_S+3)2^{-mN+1}+2(\mathcal{Q}_1+\mathcal{Q}_2+\mathcal{Q}_S\ell)\mathcal{Q}_S\ell\cdot q^{-Nn}+2n\cdot q^{-N(m-n+1)/2}+8\mathcal{Q}_S\ell e^{-t^2/2}+m'\ell\mathcal{Q}_S\cdot 2^{-2\kappa+3}$. □

4.5 Proof of Lemma 4.5

Let $\mathcal{A}$ be as in the lemma statement. We also make simplifying assumptions as in the proof of Lemma 4.4. In particular, we w.l.o.g. assume that a query $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}},\mu)$ has been made before and also queries to be made during forgery verification are already made by $\mathcal{A}$. Additionally, for simplicity and without loss of generality, we assume that the query to H_2 that $\mathcal{A}$ makes will be of the form $\mathsf{hq}=((\mathbf{u}_b,\overline{\mathbf{u}}_b)_{b\in\{0,1\}},\mathbf{H},\overline{\mathbf{t}},\mu)$ where $\mathbf{H}=\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}},\mu)$ (since such queries cannot be used for forgeries). We denote the forgery of $\mathcal{A}$ as $(\mu^*,\sigma^*=(\overline{\mathbf{t}}^*,(\mathbf{u}_b^*,\overline{\mathbf{u}}_b^*,\mathbf{z}_b^*)_{b\in\{0,1\}}))$. We then consider the following sequence of games.

Game $\mathbf{G}_0$ (Original UF-KOA game): Original UF-KOA game.

Game $\mathbf{G}_1$ (Sample $\mathbf{H}$ with trapdoor $\mathbf{R}$): For each H_1 query, instead of sampling $\mathbf{H}$ uniformly, we sample $(\mathbf{H} \in \mathcal{R}_q^{n \times m'}, \mathbf{R}) \leftarrow\!\!\$\ \mathsf{GenTrap}(q, \mathsf{b}, n, m')$. By Lemma 2.9, the distribution of $\mathbf{H}$ in the two games has statistical distance at most $2^{-N} + 2\varepsilon$. Thus, we have

$$\mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] + \mathcal{Q}_1 \cdot (2^{-N} + 2^{-2\kappa+1}) \,.$$

Game $\mathbf{G}_2$ (Relax winning condition): In this game, we relax the winning condition of $\mathcal{A}$ to the following: $\mathcal{A}$ now wins if it makes a query of the form $\mathsf{hq} = ((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu)$ to H_2 such that the sampled challenge $(c_0, c_1) \leftarrow\!\!\$\ \mathcal{C}^2$ satisfies:

$$\exists \mathbf{z}_0, \mathbf{z}_1 \in \mathcal{R}_q^n : \forall b \in \{0,1\} : \left\|\mathbf{u}_b + c_b\mathbf{t} - \mathbf{z}_b\mathbf{A}\right\| \leqslant \beta_{\mathbf{z}} \ \wedge \ \left\|\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}} - \mathbf{z}_b\mathbf{H}\right\| \leqslant \overline{\beta}_{\mathbf{z}} \,.$$

Note that the winning event of $\mathcal{A}$ in $\mathbf{G}_1$ implies this event occurring. This is because the challenge $(c_0^*, c_1^*) = \mathsf{H}_2((\mathbf{u}_b^*, \overline{\mathbf{u}}_b^*)_{b\in\{0,1\}}, \mathbf{H}^*, \overline{\mathbf{t}}^*, \mu^*)$ allows $\mathcal{A}$ to output valid responses $\mathbf{z}_0, \mathbf{z}_1$, so $\mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]$.

Game $\mathbf{G}_3$ (Change winning condition and sampling of H_2): In this game, we change *both the winning condition and how* H_2 *is simulated* as follows. On each query $((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}}, \mu)$ to H_2 where we assume that $\mathbf{H} = \mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}}, \mu)$ (for simplicity and without loss of generality), the game repeats the following until outputting $(c_0, c_1) \in \mathcal{C}^2$ or it terminates:

(Step 1) Sample uniformly random coin $\hat{\beta} \leftarrow\!\!\$\ \{0,1\}$.

(Step 2) If $\beta = 0$, sample $(c_0, c_1) \leftarrow\!\!\$\ \mathcal{C}^2$ and check if $A((c_0, c_1), (\mathsf{pk}, \mathsf{hq})) = 1$, where we define the predicate $A((c_0, c_1), (\mathsf{pk}, \mathsf{hq}))$ as follows

$$\exists \mathbf{z}_0, \mathbf{z}_1 \in \mathcal{R}_q^n : \quad \forall b \in \{0,1\} : \begin{matrix} \left\|\mathbf{u}_b + c_b\mathbf{t} - \mathbf{z}_b\mathbf{A}\right\| \leqslant \beta_{\mathbf{z}} \ \wedge \\ \left\|\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}} - \mathbf{z}_b\mathbf{H}\right\| \leqslant \overline{\beta}_{\mathbf{z}} \end{matrix} \,. \tag{3}$$

If the check *fails* output (c_0, c_1), otherwise, go back to (Step 1).

(Step 3) If $\beta = 1$, sample $c_0, c_1, c_0', c_1' \leftarrow\!\!\$\ \mathcal{C}$ and check if $B((c_0, c_1, c_0', c_1'), (\mathsf{pk}, \mathsf{hq})) = 1$, where we define the predicate $B((c_0, c_1, c_0', c_1'), (\mathsf{pk}, \mathsf{hq}))$ as follows

$$\exists \mathbf{s}' \in \mathcal{R}_q^n, b \in \{0,1\} : \quad \begin{matrix} \left\|(c_b - c_b')\mathbf{t} - \mathbf{s}'\mathbf{A}\right\| \leqslant 2\beta_{\mathbf{z}} \ \wedge \\ \left\|(c_b - c_b')\overline{\mathbf{t}} - \mathbf{s}'\mathbf{H}\right\| \leqslant 2\overline{\beta}_{\mathbf{z}} \end{matrix} \,. \tag{4}$$

Then, if the check *succeeds*, declare that $\mathcal{A}$ has won the game; otherwise, go back to (Step 1).

Note that the winning condition of $\mathcal{A}$ is now to make a H_2 query such that the game successfully samples $\beta = 1$ and (c_0, c_1, c_0', c_1') satisfying predicate B.

To bound the winning probability of $\mathcal{A}$ in this game, we will apply Theorem 3.2. In particular, we require the following lemma on the probability of sampling (c_0, c_1) and (c_0, c_1, c_0', c_1') that satisfies the predicates A and B, respectively.

Lemma 4.7. *Let* $\mathsf{pk} = (\mathbf{A}, \mathbf{t})$ *and* $\mathsf{hq} = ((\mathbf{u}_b, \overline{\mathbf{u}}_b)_{b\in\{0,1\}}, \mathbf{H}, \overline{\mathbf{t}})$ *be any public key and* H_2 *query. Then, we have that* $\mathsf{Pr}_{(c_0,c_1)}[A((c_0, c_1), (\mathsf{pk}, \mathsf{hq})) = 1] \leqslant \mathsf{Pr}_{(c_0,c_1,c_0',c_1')}[B((c_0, c_1, c_0', c_1'), (\mathsf{pk}, \mathsf{hq})) = 1]$.

Proof. Consider pk and hq as in the lemma statement. We will show this via Lemma 3.8 by introducing an intermediate predicates P_b and B' such that

$$\begin{aligned} P_b(c_b, (\mathsf{pk}, \mathsf{hq})) &:= \left(\exists \mathbf{z}_b \in \mathcal{R}_q^n : \begin{matrix} \left\|\mathbf{u}_b + c\mathbf{t} - \mathbf{z}_b\mathbf{A}\right\| \leqslant \beta_{\mathbf{z}} \ \wedge \\ \left\|\overline{\mathbf{u}}_b + c\overline{\mathbf{t}} - \mathbf{z}_b\mathbf{H}\right\| \leqslant \overline{\beta}_{\mathbf{z}} \end{matrix}\right) \,, \\ B'((c_0, c_1, c_0', c_1'), (\mathsf{pk}, \mathsf{hq})) &:= (P_0(c_0, (\mathsf{pk}, \mathsf{hq})) \ \wedge \ P_0(c_0', (\mathsf{pk}, \mathsf{hq}))) \vee \\ &\qquad (P_1(c_1, (\mathsf{pk}, \mathsf{hq})) \ \wedge \ P_1(c_1', (\mathsf{pk}, \mathsf{hq}))) \,. \end{aligned}$$

Notice that by the definition of predicate A in Equation (3), $A((c_0,c_1),(\mathsf{pk},\mathsf{hq})) = P_0(c_0,(\mathsf{pk},\mathsf{hq})) \wedge P_1(c_1,(\mathsf{pk},\mathsf{hq}))$. Thus, by Lemma 3.8, we have

$$\mathsf{Pr}[A((c_0,c_1),(\mathsf{pk},\mathsf{hq}))] \leqslant \mathsf{Pr}[B'((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq}))] \ .$$

Finally, we show that if $B'((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq})) = 1$, $B((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq})) = 1$. Notice that by definition of P_b and B', if B' is satisfied, we have that there exists $b \in \{0,1\}$ and $\mathbf{z}_b, \mathbf{z}_b' \in \mathcal{R}_q^n$ where

$$\begin{aligned} \left\|\mathbf{u}_b + c_b\mathbf{t} - \mathbf{z}_b\mathbf{A}\right\| \leqslant \beta_{\mathbf{z}} \ \wedge \ \left\|\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}} - \mathbf{z}_b\mathbf{H}\right\| \leqslant \overline{\beta}_{\mathbf{z}} \ \wedge \\ \left\|\mathbf{u}_b + c_b'\mathbf{t} - \mathbf{z}_b'\mathbf{A}\right\| \leqslant \beta_{\mathbf{z}} \ \wedge \ \left\|\overline{\mathbf{u}}_b + c_b'\overline{\mathbf{t}} - \mathbf{z}_b'\mathbf{H}\right\| \leqslant \overline{\beta}_{\mathbf{z}} \end{aligned}$$

Taking the difference of the corresponding terms and applying the triangle inequality, we have that for $\mathbf{s}' = \mathbf{z}_b - \mathbf{z}_b'$

$$\left\|(c_b - c_b')\mathbf{t} - \mathbf{s}'\mathbf{A}\right\| \leqslant 2\beta_{\mathbf{z}} \ \wedge \ \left\|(c_b - c_b')\overline{\mathbf{t}} - \mathbf{s}'\mathbf{H}\right\| \leqslant 2\overline{\beta}_{\mathbf{z}} \ .$$

This implies that B is satisfied, so

$$\mathsf{Pr}_{(c_0,c_1,c_0',c_1')}[B'((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq}))] \leqslant \mathsf{Pr}_{(c_0,c_1,c_0',c_1')}[B((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq}))] \ . \qquad \square$$

The above lemma and Theorem 3.2 then imply that $\mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1]$ with the Real and Ideal experiments corresponding to $\mathbf{G}_2^{\mathcal{A}}$ and $\mathbf{G}_3^{\mathcal{A}}$, respectively. We note that although the theorem is stated for deterministic algorithms $\mathcal{A}$, as we remarked in Remark 3.3, it applies to probabilistic $\mathcal{A}$'s as well.

Game $\mathbf{G}_4$ (Bound number of restarts in simulating H_2): In this game, the simulation of H_2 is changed. In particular, the game keeps track of a counter rst on the number of restarts in the sampling procedure described in $\mathbf{G}_3$. In particular, for the i-th query to H_2, the game repeats the following until abort or have an output determined.

- Check if $\mathsf{rst} > \max(\tau \cdot i, T)$ and abort if the check passes. Here, τ is a constant, and $T = \frac{-\ln(\delta(1-e^{-2\gamma^2}))}{2\gamma^2}$ with $\gamma = \frac{\tau-1}{2(\tau+1)}$.
- Sample an unbiased coin $\hat{\beta} \leftarrow\!\!\$\ \{0,1\}$.
- If $\beta = 0$, sample $(c_0,c_1) \leftarrow\!\!\$\ \mathcal{C}^2$ and check if $A((c_0,c_1),(\mathsf{pk},\mathsf{hq})) = 1$. Then, if the check *fails* output (c_0,c_1), otherwise, set $\mathsf{rst} \leftarrow \mathsf{rst} + 1$ and repeat.
- If $\beta = 1$, sample $c_0,c_1,c_0',c_1' \leftarrow\!\!\$\ \mathcal{C}$ and check if $B((c_0,c_1,c_0',c_1'),(\mathsf{pk},\mathsf{hq})) = 1$. Then, if the check *succeeds*, declare that $\mathcal{A}$ has won the game; otherwise, set $\mathsf{rst} \leftarrow \mathsf{rst} + 1$ and repeat.

This game now limits the number of restarts in the simulation of H_2 to be at most $\max(\tau\mathcal{Q}_2, T)$. Note that the view of $\mathcal{A}$ is identical to the prior game except when the abort occurs. Due to Theorem 3.6 and setting $\tau = 3$ and $\delta = 2^{-2\kappa}$ as in Corollary 3.7, we have that $\gamma = \frac{1}{4}$ and $T = 8(2\kappa+3) = O(\kappa)$. Then, the probability that the game aborts due to the introduced check is at most $\delta = 2^{-2\kappa}$. Therefore,

$$\mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] + 2^{-2\kappa} \ .$$

Accordingly, the game checks the predicates at most $3\mathcal{Q}_2 + 16\kappa$ times. We remark that Corollary 3.7 directly implies the game hop from $\mathbf{G}_2$ to $\mathbf{G}_4$. However, we make the transition in two steps to demonstrate the requirements for each step.

Game $\mathbf{G}_5$ (Use trapdoor to check predicates): This game replaces the inefficient checks for the two predicates $A(\cdot)$ and $B(\cdot)$ with the following checks using the trapdoor of $\mathbf{H}$ instead: Let $\mathsf{hq} = ((\mathbf{u}_b,\overline{\mathbf{u}}_b)_{b\in\{0,1\}},\mathbf{H},\overline{\mathbf{t}},\mu)$, $\mathbf{B}$ and $\mathbf{R}$ be the trapdoor of $\mathbf{H}$ sampled when the query $\mathsf{H}_1((\mathbf{u}_b)_{b\in\{0,1\}},\mu)$ is made.

- For $A((c_0,c_1),(\mathsf{hq},\mathsf{pk}))$: For $b \in \{0,1\}$, compute $\mathbf{z}_b \leftarrow \mathsf{Invert}(\mathbf{R},\mathbf{H},\overline{\mathbf{u}}_b - c_b\overline{\mathbf{t}})$ (i.e., viewing $\overline{\mathbf{u}}_b - c_b\overline{\mathbf{t}}$ as an LWE sample of $\mathbf{H}$ of which we know a trapdoor $\mathbf{R}$) and check the norm-bound inequalities in Equation (3) with respect to the computed $\mathbf{z}_0,\mathbf{z}_1$. If the Invert does not abort and check passes, output 1; otherwise, 0.

- For $B((c_0, c_1, c'_0, c'_1), (\mathsf{hq}, \mathsf{pk}))$: Compute $\mathbf{s}'_b \leftarrow \mathsf{Invert}(\mathbf{R}, \mathbf{H}, m, (c_b - c'_b)\overline{\mathbf{t}})$ and check if there exists b where

$$\left\|(c_b - c'_b)\mathbf{t} - \mathbf{s}'_b\mathbf{A}\right\| \leqslant 2\beta_{\mathbf{z}} \ \wedge \ \left\|(c_b - c'_b)\overline{\mathbf{t}} - \mathbf{s}'_b\mathbf{H}\right\| \leqslant 2\overline{\beta}_{\mathbf{z}}.$$

  If the Invert does not abort and check passes, output 1; otherwise, 0.

These changes are made so the reduction can efficiently simulate H_1 and H_2.

To argue the advantage change, we will show that these checks are equivalent to the ones made in the prior games except for some negligible probability. In particular, we consider the two cases on the outcome of A:

- $A((c_0, c_1), (\mathsf{hq}, \mathsf{pk})) = 1$: Then, there exists the $\mathbf{z}_0, \mathbf{z}_1$ satisfying the norm bounds in Equation (3). Accordingly, we can write $\mathbf{u}_b + c_b\mathbf{t}$ and $\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}}$ as

$$\mathbf{u}_b + c_b\mathbf{t} = \mathbf{z}_b\mathbf{A} + \mathbf{f}_b \ , \quad \overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}} = \mathbf{z}_b\mathbf{A} + \overline{\mathbf{f}}_b \ , \tag{5}$$

  where $\|\mathbf{f}_b\| \leqslant \beta_{\mathbf{z}}$ and $\left\|\overline{\mathbf{f}}_b\right\| \leqslant \overline{\beta}_{\mathbf{z}}$. Therefore, by Lemma 2.9 and if $\mathbf{R}$ is such that $\sqrt{\|\mathbf{R}\|^2 + 1} \leqslant \frac{q}{2\overline{\beta}_{\mathbf{z}}\sqrt{\mathsf{b}^2+1}}$, Invert will always output $\mathbf{z}_b$. Hence, the check in this game still returns 1.
- $A((c_0, c_1), (\mathsf{hq}, \mathsf{pk})) = 0$. In this case, there is $b \in \{0, 1\}$ where no $\mathbf{z}_b$ exists to write $\mathbf{u}_b + c_b\mathbf{t}$ or $\overline{\mathbf{u}}_b + c_b\overline{\mathbf{t}}$ in the form of Equation (5) with small enough error. Thus, any output $\mathbf{z}_b$ of Invert would not pass the check.

Similar argument applies as well to the predicate B by considering the vectors $(c_b - c'_b)\mathbf{t}$ and $(c_b - c'_b)\overline{\mathbf{t}}$. Therefore, the two games are equivalent except when $\mathbf{R}$ has norm $\|\mathbf{R}\| > \sqrt{\left(\frac{q}{2\overline{\beta}_{\mathbf{z}}\sqrt{\mathsf{b}^2+1}}\right)^2 - 1}$. By Lemmas 2.3 and 2.9, the norm of $\mathbf{R}$ is at most $\sigma_{\mathsf{td}}\sqrt{m'm}N$ (which is $\leqslant \frac{q}{4\overline{\beta}_{\mathbf{z}}\mathsf{b}} \leqslant \sqrt{\left(\frac{q}{2\overline{\beta}_{\mathbf{z}}\sqrt{\mathsf{b}^2+1}}\right)^2 - 1}$ by how we set q in Table 1) except with probability $2^{-N(m' - n\log_{\mathsf{b}} q)}$ per each sampling of $\mathbf{H}$. Thus, by the union bound, we have that

$$\mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] + \mathcal{Q}_1 \cdot 2^{-N(m' - n\log_{\mathsf{b}} q)} \ .$$

Game $\mathbf{G}_6$ (Abort if sampled challenges are identical): This game aborts if for the sampled challenges c_0, c_1, c'_0, c'_1 that determine the winning condition via predicate $B((c_0, c_1, c'_0, c'_1), (\mathsf{hq}, \mathsf{pk}))$, are such that $c_0 = c'_0$ *or* $c'_1 = c'_1$. This game is to ensure that the reduction will be able to extract the MLWE secret when a successful check of predicate B occurs as $c'_b - c_b \neq 0$, so it is invertible in $\mathcal{K} = \mathbb{Q}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$, where $\mathbf{s}'$ is extracted via Invert as in the check in $\mathbf{G}_5$.

Since c_0, c_1, c'_0, c'_1 are uniform in $\mathcal{C}$, the probability of this bad event occurring is at most $2/|\mathcal{C}|$ per a check of B. By the change in $\mathbf{G}_4$, the game only checks the predicate B for at most $3\mathcal{Q}_2$ times, by the union bound, we have that

$$\mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1] + \frac{6\mathcal{Q}_2}{|\mathcal{C}|} \ .$$

<u>Reduction $\mathcal{B}$.</u> We now give the reduction $\mathcal{B}$ to bound the probability of $\mathcal{A}$ winning in game $\mathbf{G}_6$. The reduction $\mathcal{B}$ takes as input $(\mathbf{A}, \mathbf{t})$ and runs $\mathcal{A}^{\mathsf{H}_1, \mathsf{H}_2}(\mathsf{pk} = (\mathbf{A}, \mathbf{t}))$ while simulating the oracles H_1 and H_2 as in $\mathbf{G}_6$ and ignores the final forgery of $\mathcal{A}$. Whenever the check on predicate B using the corresponding trapdoors (as introduced in $\mathbf{G}_5$) returns 1, let c_0, c_1, c'_0, c'_1 be the challenges sampled and $\mathbf{s}'_0, \mathbf{s}'_1$ be outputs of Invert during the check. The reduction $\mathcal{B}$ checks which $b \in \{0, 1\}$ is such that

$$\left\|(c_b - c'_b)\mathbf{t} - \mathbf{s}'_b\mathbf{A}\right\| \leqslant 2\beta_{\mathbf{z}} \ \wedge \ \left\|(c_b - c'_b)\overline{\mathbf{t}} - \mathbf{s}'_b\mathbf{H}\right\| \leqslant 2\overline{\beta}_{\mathbf{z}} \ .$$

Then, it computes

- $\mathbf{e}' = (c_b - c'_b)\mathbf{t} - \mathbf{s}'_b\mathbf{A}$
- $\mathbf{e} = (c_b - c'_b)^{-1}\mathbf{e}'$ treating $(c_b - c'_b)^{-1}$ as in the field $\mathcal{K} = \mathbb{Q}[\mathsf{X}]/\langle \mathsf{X}^N + 1\rangle$
- Compute and return $\mathbf{s}$ such that $\mathbf{t} - \mathbf{e} = \mathbf{s}\mathbf{A}$ by solving the linear equation.

Analysis of $\mathcal{B}$. By how $\mathcal{B}$ is constructed, the output $\mathbf{e}' = (c_b - c'_b)\mathbf{t} - \mathbf{s}'_b\mathbf{A}$ is such that $\|\mathbf{e}'\| \leqslant 2\beta_{\mathbf{z}}$. Since $\mathbf{t}$ is an sMLWE sample from the game, it can be written as $\mathbf{t} = \mathbf{s}\mathbf{A} + \mathbf{e}$ sampled as in the key generation algorithm. Then, if $\|\mathbf{e}\| \leqslant \sigma_{\mathsf{sk}}\sqrt{mN}$ (occurs except with probability at most 2^{-mN+1} via Lemma 2.3), we can write

$$\begin{aligned}\mathbf{e}' &= (c_b - c'_b)\mathbf{t} - \mathbf{s}'_b\mathbf{A}\\ \mathbf{e}' &= (c_b - c'_b)(\mathbf{s}\mathbf{A} + \mathbf{e}) - \mathbf{s}'_b\mathbf{A}\\ \mathbf{e}' - (c_b - c'_b)\mathbf{e} &= ((c_b - c'_b)\mathbf{s} - \mathbf{s}'_b)\mathbf{A}\ .\end{aligned}$$

Notice that $\|\mathbf{e}' - (c_b - c'_b)\mathbf{e}\| \leqslant 2\beta_{\mathbf{z}} + 2\gamma_c\|\mathbf{e}\| \leqslant 4\beta_{\mathbf{z}} < \frac{1}{8\sqrt{N}}q^{1-\frac{n}{m}}$. Hence, due to Lemma 2.6, with probability 2^{-N}, we have that $(c_b - c'_b)\mathbf{s} = \mathbf{s}'_b$ and $\mathbf{e}' = (c_b - c'_b)\mathbf{e}$. Thus, with $c_b \neq c'_b$, $\mathbf{e} = (c_b - c'_b)^{-1}\mathbf{e}'$ in K, and accordingly $\mathcal{B}$ finds $\mathbf{s}$ since $\mathbf{t} - \mathbf{e} = \mathbf{s}\mathbf{A}$. Therefore,

$$\mathsf{Pr}[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Adv}^{\mathsf{smlwe}}_{n,m,N,q,\sigma_{\mathsf{sk}}}(\mathcal{B}, \kappa) + 2^{-mN+1} + 2^{-N}\ .$$

Thus, we conclude the bound in the lemma by combining all the advantage changes. The running time follows from the change in game $\mathbf{G}_4, \mathbf{G}_5$. □

Acknowledgements

Chairattana-Apirom and Tessaro's research was partially supported by NSF grants CNS-2026774, CNS-2154174, CNS-2426905, a gift from Microsoft, and a Stellar Development Foundation Academic Research Award. Nico Döttling: Funded by the European Union (ERC, LACONIC, 101041207). Julian Loss was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and the European Union, ERC-2023-STG, Project ID: 101116713. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

ABB+17. Erdem Alkim, Nina Bindel, Johannes A. Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting TESLA in the quantum random oracle model. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 143–162. Springer, Cham, 2017. doi:10.1007/978-3-319-59879-6_9. Cited on Page 2.

AFLT12. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_34. Cited on Page 2.

AGHS13. Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 97–116. Springer, Berlin, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_6. Cited on Page 10.

AGJ+24. Sven Argo, Tim Güneysu, Corentin Jeudy, Georg Land, Adeline Roux-Langlois, and Olivier Sanders. Practical post-quantum signatures for privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 1523–1537. ACM Press, October 2024. doi:10.1145/3658644.3670297. Cited on Page 21.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_3. Cited on Page 3.

BJLS16. Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 273–304. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_10. Cited on Page 2.

BJRW23. Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module learning with errors with short distributions. *Journal of Cryptology*, 36(1):1, January 2023. doi:10.1007/s00145-022-09441-3. Cited on Page 9.

BKKP15. Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 256–279. Springer, Berlin, Heidelberg, March / April 2015. doi:10.1007/978-3-662-46447-2_12. Cited on Page 2.

BL16. Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 404–434. Springer, Berlin, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_14. Cited on Pages 2 and 3.

BL26. Katharina Boudgoust and Oleksandra Lapiha. Leftover hash lemma(s) over cyclotomic rings. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 62–91, Cham, 2026. Springer Nature Switzerland. doi:10.1007/978-3-032-25327-9_3. Cited on Page 10.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. doi:10.1145/168588.168596. Cited on Page 1.

BTT22. Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi. MuSig-L: Lattice-based multi-signature with single-round online phase. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 276–305. Springer, Cham, August 2022. doi:10.1007/978-3-031-15979-4_10. Cited on Page 10.

CAHT26. Rutchathon Chairattana-Apirom, Dennis Hofheinz, and Stefano Tessaro. Tight security for bbs signatures. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 246–276, Cham, 2026. Springer Nature Switzerland. doi:10.1007/978-3-032-25291-3_9. Cited on Page 2.

Che05. Benoît Chevallier-Mames. An efficient CDH-based signature scheme with a tight security reduction. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 511–526. Springer, Berlin, Heidelberg, August 2005. doi:10.1007/11535218_31. Cited on Pages 3 and 19.

DGJL21. Denis Diemert, Kai Gellert, Tibor Jager, and Lin Lyu. More efficient digital signatures with tight multi-user security. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 1–31. Springer, Cham, May 2021. doi:10.1007/978-3-030-75248-4_1. Cited on Page 2.

DOTT22. Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. *Journal of Cryptology*, 35(2):14, April 2022. doi:10.1007/s00145-022-09425-3. Cited on Page 11.

EEN+24. Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld. Plover: Masking-friendly hash-and-sign lattice signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VII*, volume 14657 of *LNCS*, pages 316–345. Springer, Cham, May 2024. doi:10.1007/978-3-031-58754-2_12. Cited on Page 2.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Berlin, Heidelberg, August 2005. doi:10.1007/11535218_10. Cited on Page 3.

GBL08. Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107. Springer, Berlin, Heidelberg, August 2008. doi:10.1007/978-3-540-85174-5_6. Cited on Page 2.

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, Part I*, volume 12110 of *LNCS*, pages 623–651. Springer, Cham, May 2020. doi:10.1007/978-3-030-45374-9_21. Cited on Pages 4, 10, and 36.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. doi:10.1145/1374376.1374407. Cited on Pages 2, 10, and 36.

HJ12. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Berlin, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_35. Cited on Page 2.

HLWG23. Shuai Han, Shengli Liu, Zhedong Wang, and Dawu Gu. Almost tight multi-user security under adaptive corruptions from LWE in the standard model. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 682–715. Springer, Cham, August 2023. doi:10.1007/978-3-031-38554-4_22. Cited on Page 3.

HOS25. Keitaro Hashimoto, Wakaha Ogata, and Yusuke Sakai. Signatures with tight adaptive corruptions from search assumptions. Cryptology ePrint Archive, Report 2025/115, 2025. URL: https://eprint.iacr.org/2025/115. Cited on Page 3.

JT20. Joseph Jaeger and Stefano Tessaro. Expected-time cryptography: Generic techniques and applications to concrete soundness. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 414–443. Springer, Cham, November 2020. doi:10.1007/978-3-030-64381-2_15. Cited on Page 7.

JTWZ26. Kaijie Jiang, Stefano Tessaro, Hoeteck Wee, and Chenzhi Zhu. Tweed: Adaptively secure lattice-based two-round threshold signatures. In *Advances in Cryptology - EUROCRYPT 2026*, 2026. To appear. Cited on Page 5.

KLP17. Eike Kiltz, Julian Loss, and Jiaxin Pan. Tightly-secure signatures from five-move identification protocols. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 68–94. Springer, Cham, December 2017. doi:10.1007/978-3-319-70700-6_3. Cited on Pages 3, 4, and 19.

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78372-7_18. Cited on Page 2.

KMP16. Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Berlin, Heidelberg, August 2016. doi:10.1007/978-3-662-53008-5_2. Cited on Page 2.

KW03. Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 2003*, pages 155–164. ACM Press, October 2003. doi:10.1145/948109.948132. Cited on Page 2.

KYY18. Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 253–282. Springer, Cham, December 2018. doi:10.1007/978-3-030-03329-3_9. Cited on Page 3.

LDK+22. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Cited on Pages 1, 2, and 20.

LLW20. Qiqi Lai, Feng-Hao Liu, and Zhedong Wang. Almost tight security in lattices with polynomial moduli - PRF, IBE, all-but-many LTF, and more. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 652–681. Springer, Cham, May 2020. doi:10.1007/978-3-030-45374-9_22. Cited on Page 3.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022. doi:10.1007/978-3-031-15979-4_3. Cited on Pages 11, 20, and 21.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010. doi:10.1007/978-3-642-13190-5_1. Cited on Page 2.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_3. Cited on Page 10.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015. doi:10.1007/s10623-014-9938-4. Cited on Pages 2, 10, 11, and 20.

LS18. Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78381-9_8. Cited on Page 9.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009. doi:10.1007/978-3-642-10366-7_35. Cited on Pages 2, 4, and 20.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_43. Cited on Pages 2, 4, 11, and 20.

MM11. Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Berlin, Heidelberg, August 2011. doi:10.1007/978-3-642-22792-9_26. Cited on Page 2.

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_41. Cited on Pages 10 and 21.

PFH+22. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Cited on Pages 1 and 2.

PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 461–473. ACM Press, June 2017. doi:10.1145/3055399.3055489. Cited on Page 2.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. doi:10.1007/s001450010003. Cited on Page 2.

PV05. Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, December 2005. doi:10.1007/11593447_1. Cited on Page 2.

PW22. Jiaxin Pan and Benedikt Wagner. Lattice-based signatures with tight adaptive corruptions and more. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 347–378. Springer, Cham, March 2022. doi:10.1007/978-3-030-97131-1_12. Cited on Pages 2 and 3.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603. Cited on Page 2.

RSW18. Miruna Rosca, Damien Stehlé, and Alexandre Wallet. On the ring-LWE and polynomial-LWE problems. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 146–173. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78381-9_6. Cited on Page 2.

Seu12. Yannick Seurin. On the exact security of Schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_33. Cited on Page 2.

YJW23. Yang Yu, Huiwen Jia, and Xiaoyun Wang. Compact lattice gadget and its applications to hash-and-sign signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 390–420. Springer, Cham, August 2023. doi:10.1007/978-3-031-38554-4_13. Cited on Page 2.

ZLPG19. Xiao Zhang, Shengli Liu, Jiaxin Pan, and Dawu Gu. Tightly secure signature schemes from the LWE and subset sum assumptions. *Theor. Comput. Sci.*, 795:326–344, 2019. URL: https://doi.org/10.1016/j.tcs.2019.07.015, doi:10.1016/J.TCS.2019.07.015. Cited on Page 3.

A Proof of Lemma 2.8 for Rings

Our proof will follow closely the proof structure from [GMPW20, Theorem 5.3]. We will first recall some important lemmata from [GPV08, GMPW20] adapted to rings. We adopt the notation $P \overset{\varepsilon}{\approx} Q$ from [GMPW20]: for any two distributions P, Q with the same support and $\varepsilon > 0$, we say that $P \overset{\varepsilon}{\approx} Q$ if and only if $\max_{x \in \mathsf{Supp}(P)} |\log P(x) - \log Q(x)| \leqslant \log(1+\varepsilon)$. We denote $\Lambda \cdot \mathbf{T}^\top$ as making linear transformation $\mathbf{T}$ to the (row) vectors in the lattice Λ.

Lemma A.1 (Corollary 2.8 [GPV08]). *Let $\Lambda \subseteq \Lambda'$ be m-dimensional full-rank lattices. Then, for any $\varepsilon \in (0,1)$, any $\sigma > \eta_\varepsilon(\Lambda')$, the distribution of $(\mathcal{D}_{\Lambda,\sigma} \mod \Lambda') \overset{4\varepsilon}{\approx} (\Lambda/\Lambda')$.*

Lemma A.2 (Lemma 2.5 of [GMPW20] adapted). *For any lattice $\Lambda, \varepsilon > 0$ and matrices $\mathbf{T}, \mathbf{S}$ with full column rank, we have that $\eta_\varepsilon(\Lambda)^2 \preceq \mathbf{S}\mathbf{S}^\top$ if and only if $\eta_\varepsilon(\Lambda \cdot \mathbf{T})^2 \preceq \mathbf{T}\mathbf{S}\mathbf{S}^\top\mathbf{T}^\top$.*

Lemma A.3 (Theorem 3.1 of [GMPW20] adapted). *For any $\varepsilon \in [0,1)$, $\bar{\varepsilon} = 2\varepsilon/(1-\varepsilon)$, a positive definite $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$, a lattice coset $A + \mathbf{a} \in \mathsf{Span}(\sqrt{\mathbf{\Sigma}})$ and a matrix $\mathbf{T} \in \mathbb{R}^{m' \times m}$ such that $\ker(\mathbf{T})$ is a Λ-subspace (i.e., $\mathsf{Span}(\ker(\mathbf{T}) \cap \Lambda) = \ker(\mathbf{T})$) and $\eta_\varepsilon(\Lambda \cap \ker(\mathbf{T})) \leqslant \sqrt{\mathbf{\Sigma}}$. Then, we have that $\mathcal{D}_{A,\sqrt{\mathbf{\Sigma}}} \cdot \mathbf{T}^\top \overset{\bar{\varepsilon}}{\approx} \mathcal{D}_{A\mathbf{T}^\top, \mathbf{T}\sqrt{\mathbf{\Sigma}}}$.*

Because $\mathbf{A}$ is primitive, for any $\mathbf{a} \in \mathcal{R}^n$ there exists an $\mathbf{x}^* \in \mathcal{R}^m$ such that $\mathbf{A}(\mathbf{x}^*)^\top = \mathbf{a}^\top$. We note that because Lemma A.1 and the assumption that $\sigma_x \geqslant \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$, we have that for any $\mathbf{a}$

$$\mathsf{Pr}_{\mathbf{x} \leftarrow\!\$\, \mathcal{D}_{\mathcal{R}^m, \sigma_x}}[\mathbf{A}\mathbf{x}^\top = \mathbf{a}^\top] = (1 \pm 4\varepsilon) q^{-nN} \ .$$

Then, conditioning on $\mathbf{A}\mathbf{x}^\top = \mathbf{a}^\top$, we have that $\mathbf{x}$ is distributed as $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}) + \mathbf{x}^*, \sigma_x}$. Now, notice that for $\mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e}$, we have that

$$\mathbf{b}\mathbf{x}^\top + y = \mathbf{s}\mathbf{A}\mathbf{x}^\top + \mathbf{e}\mathbf{x}^\top + y = \mathbf{s}\mathbf{a}^\top + \mathbf{e}\mathbf{x}^\top + y \ .$$

Thus, we are left to show that $\mathbf{e}\mathbf{x}^\top + y$ is distributed similarly to $\bar{e} \leftarrow\!\$\, \mathcal{D}_{\mathcal{R}, \sqrt{\overline{\mathbf{\Sigma}}}}$ where $\overline{\mathbf{\Sigma}} = \sigma_x^2 \mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^\top + \sigma_y$.

To do this, first observe that the distribution of $\mathbf{e}\mathbf{x}^\top + y$ is

$$\mathsf{Rot}(\mathbf{e}) \cdot (\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}) + \mathbf{x}^*, \sigma_x})^\top + \mathcal{D}_{\mathbb{Z}^N, \sigma_y} = [\mathsf{Rot}(\mathbf{e}) \mid \mathbb{I}_N](\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}) \times \mathbb{Z}^N + (\mathbf{x}^*, \mathbf{0}), \sqrt{\mathbf{\Sigma}'}})^\top \ ,$$

where $\mathbf{\Sigma}' = \begin{bmatrix} \sigma_x^2 \mathbb{I}_{mN} & \\ & \sigma_y^2 \mathbb{I}_N \end{bmatrix}$. Let $A' = \Lambda_q^\perp(\mathbf{A}) \times \mathbb{Z}^N + (\mathbf{x}^*, \mathbf{0})$ be the lattice coset. Applying Lemma A.3, we can conclude the proof as the resulting lattice coset $A'[\mathsf{Rot}(\mathbf{e}) \mid \mathbb{I}_N]^\top = \mathbb{Z}^N + \mathbf{x}^*\mathsf{Rot}(\mathbf{e})^\top = \mathbb{Z}^N$ and $\overline{\mathbf{\Sigma}} = \sigma_x^2 \mathsf{Rot}(\mathbf{e})\mathsf{Rot}(\mathbf{e})^\top + \sigma_y^2$. Hence, we only have to show that the requirements of the lemma are satisfied. Namely, $\ker([\mathsf{Rot}(\mathbf{e}) \mid \mathbb{I}_N])$ is a $(\Lambda' = \Lambda_q^\perp(\mathbf{A}) \times \mathbb{Z}^N)$-subspace and the smoothing parameter is bounded.

First, observe that the kernel of $[\mathsf{Rot}(\mathbf{e}) \mid \mathbb{I}_N]$ is $K = \{(\mathbf{v}^\top, -\mathsf{Rot}(\mathbf{e})\mathbf{v}^\top)^\top : \mathbf{v} \in \mathbb{R}^{mN}\}$. This is easy to see since $[\mathsf{Rot}(\mathbf{e}) \mid \mathbb{I}_N]$ is full-rank and K has dimension mN. Also, observe that $K \cap \Lambda'$ contains exactly vectors $(\mathbf{v}^\top, -\mathsf{Rot}(\mathbf{e})\mathbf{v}^\top)^\top$ with $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$. In particular, we can see Λ' as a result of the linear transformation $\mathbf{T} = \begin{bmatrix} \mathbb{I}_{mN} \\ \mathsf{Rot}(\mathbf{e}) \end{bmatrix}$ such that $\Lambda' = \Lambda_q^\perp(\mathbf{A}) \cdot \mathbf{T}^\top$. Since $\Lambda_q^\perp(\mathbf{A})$ is full-rank, $\mathsf{Span}(K \cap \Lambda') = K$.

Finally, recall from the hypothesis of the lemma that $\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leqslant s = (\sigma_x^{-2} + \sigma_y^{-2} \|\mathsf{Rot}(\mathbf{e})\|^2)^{-1/2}$; via Lemma A.2 we only need to show that $s^2\mathbf{T}\mathbf{T}^\top \preceq \mathbf{\Sigma}'$, i.e., $\mathbf{\Sigma}' - s^2\mathbf{T}\mathbf{T}^\top \succeq 0$.

Notice that $\mathbf{\Sigma}' = \sqrt{\mathbf{\Sigma}'}\sqrt{\mathbf{\Sigma}'}^\top$ is full rank. Then, it suffices to show that $\mathbb{I}_{(m+1)N} - s^2\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T}\mathbf{T}^\top(\sqrt{\mathbf{\Sigma}'}^\top)^{-1} \succeq 0$, or equivalently,

$$s_1(s^2\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T}\mathbf{T}^\top(\sqrt{\mathbf{\Sigma}'}^\top)^{-1}) \leqslant 1 \ .$$

We observe that $\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T} = \begin{bmatrix} \sigma_x^{-1}\mathbb{I}_{mN} \\ \sigma_y^{-1}\mathsf{Rot}(\mathbf{e}) \end{bmatrix}$. Therefore, we consider

$$\begin{aligned}
s_1(s^2\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T}\mathbf{T}^\top(\sqrt{\mathbf{\Sigma}'}^\top)^{-1}) &= s^2\left\|\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T}\mathbf{T}^\top(\sqrt{\mathbf{\Sigma}'}^\top)^{-1}\right\| \\
&= s^2\left\|\mathbf{T}^\top(\sqrt{\mathbf{\Sigma}'}^\top)^{-1}\sqrt{\mathbf{\Sigma}'}^{-1}\mathbf{T}\right\|^2 \\
&= s^2\left\|\begin{bmatrix}\sigma_x^{-1}\mathbb{I}_{mN} & \sigma_y^{-1}\mathsf{Rot}(\mathbf{e})^\top\end{bmatrix}\begin{bmatrix}\sigma_x^{-1}\mathbb{I}_{mN} \\ \sigma_y^{-1}\mathsf{Rot}(\mathbf{e})\end{bmatrix}\right\| \\
&= s^2\left\|\sigma_x^{-2}\mathbb{I}_{mN} + \sigma_y^{-2}\mathsf{Rot}(\mathbf{e})^\top\mathsf{Rot}(\mathbf{e})\right\| \\
&\leqslant s^2(\sigma_x^{-2} + \sigma_y^{-2}\left\|\mathsf{Rot}(\mathbf{e})^\top\mathsf{Rot}(\mathbf{e})\right\|) \\
&\leqslant s^2(\sigma_x^{-2} + \sigma_y^{-2}\left\|\mathsf{Rot}(\mathbf{e})\right\|^2) \leqslant 1\ ,
\end{aligned}$$

where the last inequality follows from how s is defined. Thus, we conclude the proof by consider the distribution of $(\mathbf{A}\mathbf{x}^\top, \mathbf{e}\mathbf{x}^\top + y)$. For any $\mathbf{a} \in \mathcal{R}^n, e \in \mathcal{R}$

$$\begin{aligned}
&\mathsf{Pr}_{\mathbf{x}\leftarrow\$\,\mathcal{D}_{\mathcal{R}^m,\sigma_x}, y\leftarrow\$\,\mathcal{D}_{\mathcal{R},\sigma_y}}[\mathbf{A}\mathbf{x}^\top = \mathbf{a}^\top, \mathbf{e}\mathbf{x}^\top + y = e] \\
&= \mathsf{Pr}_{\mathbf{x}}[\mathbf{A}\mathbf{x}^\top = \mathbf{a}^\top]\mathsf{Pr}_{\mathbf{x}\leftarrow\$\,\mathcal{D}_{\mathcal{R}^m,\sigma_x}, y\leftarrow\$\,\mathcal{D}_{\mathcal{R},\sigma_y}}[\mathbf{e}\mathbf{x}^\top + y = e | \mathbf{A}\mathbf{x}^\top = \mathbf{a}^\top] \\
&= (1 \pm 4\varepsilon)q^{-nN} \cdot \mathsf{Pr}_{\mathbf{x}\leftarrow\$\,\mathcal{D}_{\Lambda_q^\perp(\mathbf{A})+\mathbf{x}^*,\sigma_x}, y\leftarrow\$\,\mathcal{D}_{\mathcal{R},\sigma_y}}[\mathbf{e}\mathbf{x}^\top + y = e] \\
&= (1 \pm 4\varepsilon)q^{-nN} \cdot (1 \pm 2\bar{\varepsilon})\mathcal{D}_{\mathcal{R},\sqrt{\bar{\mathbf{\Sigma}}}}(e)\ .
\end{aligned}$$

Thus, the statistical distance is at most $\frac{1}{2}(4\varepsilon + 2\bar{\varepsilon}) \approx 2\bar{\varepsilon}$. □