Two-Round Threshold Signature from Algebraic One-More Learning with Errors

Thomas Espitau[1], Shuichi Katsumata[1,2], Kaoru Takemure* [1,2]

[1]PQShield

{thomas.espitau, shuichi.katsumata, kaoru.takemure}@pqshield.com

[2]AIST

July 2, 2024

Abstract

Threshold signatures have recently seen a renewed interest due to applications in cryptocurrency while NIST has released a call for multi-party threshold schemes, with a deadline for submission expected for the first half of 2025. So far, all lattice-based threshold signatures requiring two-rounds or less are based on heavy tools such as (fully) homomorphic encryption (FHE) and homomorphic trapdoor commitments (HTDC). This is not unexpected considering that most efficient two-round signatures from classical assumptions either rely on idealized model such as algebraic group models or on one-more type assumptions, none of which we have a nice analogue in the lattice world.

In this work, we construct the first efficient two-round lattice-based threshold signature without relying on FHE or HTDC. It has an offline-online feature where the first round can be preprocessed without knowing message or the signer sets, effectively making the signing phase non-interactive. The signature size is small and shows great scalability. For example, even for a threshold as large as 1024 signers, we achieve a signature size roughly 11 KB. At the heart of our construction is a new lattice-based assumption called the *algebraic one-more learning with errors* (AOM-MLWE) assumption. We believe this to be a strong inclusion to our lattice toolkits with an independent interest. We establish the selective security of AOM-MLWE based on the standard MLWE and MSIS assumptions, and provide an in depth analysis of its adaptive security, which our threshold signature is based on.

*Most of this work was done while this author was a PhD student at The University of Electro-Communications, Japan.

Contents

1 Introduction

A T-out-of-N threshold signature [Des90, DF90] allows to distribute a secret signing key to N signers, where any set of the $T \leq N$ signers can collaborate to sign a message. Security guarantees that a set of signers less than T cannot produce a valid signature. While threshold signatures have always been a topic of interest, in recent years, it has seen a renewed real-world interest largely due to applications in cryptocurrency, where secure and reliable storage of cryptographic keys is vital. Such interest has led US agency NIST to release a call for multi-party threshold schemes [PB23], with a deadline for submission expected for the first half of 2025.

Current State of Post-Quantum Threshold Signature. *Classically* secure threshold signature has approached a high state of maturity with the recent rapid developments. We now have a plethora of efficient solutions, covering many design choices, such as threshold BLS [Bol03, BL22], threshold ECDSA [GG18, LN18, DKLs19, DOK+20, CGG+20, CCL+20, DJN+20], and threshold Schnorr [SS01, GJKR07, KG20, Lin22, BCK+22, CGRS23, CKM23b].

While development on *post-quantum* threshold signature has been elusive for many years, we have started to see some interesting progress lately. The first *round-optimal* (i.e., one-round) lattice-based threshold signature was by Boneh et al. [BGG+18], later optimized by Agrawal, Stehlé, and Yadav [ASY22]. This remained mainly of theoretical interest as they required a threshold fully homomorphic encryption (FHE) to compute a standard (non-thresholdized) signature. Very recently, Gur, Katz, and Silde [GKS23] building on similar ideas, constructed a *two-round* threshold signature based on a threshold linear homomorphic encryption and homomorphic trapdoor commitment (HTDC) [GVW15b, DOTT21]. They provide a rough estimate claiming a signature size of around 47 KB with 3 MB communication per signer for the 3-out-of-5 setting. While this brings the original idea of [BGG+18] closer to practice, it does not scale well due to the heavy use of HTDC. In an independent and concurrent work, del Pino et al. [PKM+24] constructed a *three-round* threshold signature without relying on any heavy tools such as FHE or HTDC for the first time. As such, [PKM+24] has a small signature size of 13 KB with only 40 KB communication per user, achieving great scalability supporting a threshold T as large as 1024, a parameter range considered by NIST [PB23].

A Closer Look at Round Complexity. While [PKM+24] brings lattice-based threshold signatures to the practical regime, the main drawback is that it requires three rounds. In environments where signers are using network-limited devices or unreliable networks for transmission, multiple rounds may become a performance bottleneck. This is why there is a strong interest in a round-optimal or a so-called *offline-online efficient* two-round protocol [BD22, Section 5.3.5]. The latter type allows preprocessing the first-round without knowing the set of T signers and the message to be signed, effectively making the online signing phase non-interactive.

In the classical setting, we have efficient solutions for both of these types: threshold BLS [Bol03, BL22] offers a round-optimal protocol, whereas threshold Schnorr such as FROST and its variants [KG20, BCK+22, CGRS23] offer an offline-online efficient two-round protocol. This is in sharp contrast to the post-quantum setting where we currently need heavy tools like FHE or HTDC for threshold signatures offering two rounds or less.

Barriers to 2(≥)-Round Lattice Schemes. When we look at how these classical protocols achieve low round complexity, the fundamental barriers in replicating them in the lattice setting become clear. First, the round-optimal threshold BLS is based on the BLS signature [BLS01]; a signature scheme using the rich algebraic properties of bilinear maps, something thought to be highly unlikely to be reproducible from lattices. On the other hand, the two-round threshold Schnorr like FROST only requires standard group operations for the construction. Unfortunately, the security proof relies on either the algebraic group model (AGM) [FKL18] or a variant of the one-more discrete logarithm (OM-DL) problem, both of which we do not have a nice analogue in the lattice world.[1] Indeed, as exemplified with FHE computation, since lattice operations can be non-algebraic, an idealized model like AGM does not seem to meaningfully capture lattice adversaries. To make matters worse, this does not seem to be just an artifact of the proof technique as a

[1] Note that while we have one-more-ISIS [AKSY22], an assumption having "one-more" in its name, it is qualitatively quite different from those considered in the classical setting. See Section 1.2 for more discussion.

simple adaptation of the classical constructions is known to lead to insecure schemes.

In summary, to construct a lattice-based threshold signature with two rounds or less, we need to develop new techniques not yet in our lattice toolkits. This brings us to the main question of this work:

Can we replicate the classically secure efficient $2(\geq)$*-round threshold signatures from lattices?*

1.1 Our Contribution

In this work, we construct a new lattice-based offline-online efficient two-round threshold signature. Unlike prior works on lattice-based one or two-round threshold signatures [BGG+18, ASY22, GKS23], we do not rely on heavy tools such as FHE or HTDC. At a high level, our scheme is similar to the simple threshold Schnorr protocol FROST [KG20], one of the most popular classically secure two-round threshold signatures. In fact, it can be viewed as a thresholdized version of Raccoon [dPEK+23], a lattice-based signature scheme by del Pino et al., submitted to the additional NIST call for proposals [NIS22]. This interchangeability is a desirable property as it allows us to seamlessly use our threshold signature in an ecosystem with Raccoon.

At the heart of our construction is a new lattice (falsifiable [Nao03, GW11]) assumption named the *algebraic one-more module Learning with Errors* (AOM-MLWE) assumption. AOM-MLWE is defined, in spirit, similarly to the algebraic one-more discrete logarithm (AOM-DL) assumption, originally introduced by Nick, Ruffing, and Seurin [NRS21] to establish the security of the multi-signature scheme called MuSig2. AOM-DL is a strictly weaker assumption than the (non-falsifiable and non-algebraic) OM-DL. Informally, in OM-DL, an adversary has access to a very strong oracle that solves the discrete logarithm of any group element of its choice; in contrast, in AOM-DL, an adversary is limited to access this DL solving oracle on an *algebraic* combination of the provided challenge instances.[2] While the distinction may seem insignificant at first sight, it has a large impact in the lattice setting. This extra algebraic restriction on the adversary is the key allowing us to provide a well-defined and non-trivial definition.[3] See Section 4.1 for more detailed discussions on why a non-algebraic OM-MLWE would be difficult to define and use.

In more detail, half of our work is devoted to a theoretical and practical analysis of the newly introduced AOM-MLWE assumption. As typical with any lattice-based assumptions, the hardness of AOM-MLWE problem is dictated by many parameters. The most unique restriction to AOM-MLWE is the "allowed" algebraic combinations that an adversary can query to the MLWE solving oracle. Since MLWE secrets are small, there are several trivial queries an adversary can make to break the AOM-MLWE problem with a naive parameter selection. In our work, we pinpoint what these "weak" instances are and analyze the hardness of AOM-MLWE for specific "hard" instances, one of which underlies our threshold signature. Concretely, we first show that a *selective* variant of AOM-MLWE (sel-AOM-MLWE) of these hard instances is as secure as MLWE and MSIS — a variant where the adversary must commit to all the queries at the outset of the security game. We then provide an in-depth cryptanalysis analyzing the effect of an adaptive adversary and heuristically establish that an adaptive adversary is no stronger than a selective adversary.

It is worth noting that we have recently seen a boom in new lattice-based assumptions, used to construct exciting primitives: one-more-ISIS [AKSY22], K-R-ISIS [ACL+22], BASIS [WW23], evasive LWE [Wee22, Tsa22], only to name a few. While some (variants of the) assumptions can be based on standard lattice-assumptions, many of them are still new and have not undergone scrutiny, both from theory and practical cryptanalysis. Within this landscape, our assumption is in spirit closest to the *adaptive* LWE problem by Quach, Wee, and Wichs [QWW18], used to construct adaptively secure laconic function evaluation schemes and attribute-based encryption schemes [LLL22]. Similarly to AOM-MLWE, while adaptive LWE is heuristically thought to be as hard as LWE, the selective variant is implied by the standard definition of LWE. We view this as one characteristic that differentiates AOM-MLWE from recent assumptions.

[2] In more detail, in OM-DL, the adversary is given $g^{\mathbf{a}} := (g^{a_i})_{i\in[Q]}$ as the challenge; can query any $h \in \mathbb{G}$ to the oracle; and receives $\mathsf{dlog}_g(h)$. In contrast, in AOM-DL, the adversary can only query $\mathbf{d} \in \mathbb{Z}_p^Q$ and receives $\langle \mathbf{a}, \mathbf{d}\rangle$, making the oracle efficient.

[3] Note that this is fundamentally different from the AGM where the adversary is restricted to be algebraic. In AOM-MLWE, while the adversary can only make algebraic queries to the MLWE solving oracle, it has otherwise no algebraic restrictions.

The second half of our work is devoted to the construction of our two-round threshold signature. The starting point of our construction is the recent efficient three-round threshold signature by del Pino et al. [PKM+24], which is in a bird's eye view, an analog of the folklore construction of a three-round Schnorr signature using Shamir's secret sharing protocol [Sha79a]. Our high-level strategy to make it two-round is similar to FROST [KG20], however, there arise many lattice-related complications. As we explained above, the hardness of AOM-MLWE is dictated by the choice of the parameters, and consequently, our threshold signature must be constructed meticulously to comply with these restrictions. Along the way, as an independent interest, we resolve one of the open problems stated in [PKM+24]. In their construction, they required each signer to maintain a long-term state and to authenticate their views with a standard (non-thresholdized) signature for unforgeability. Our two-round construction resolves both issues without any overhead.

Lastly, our two-round threshold signatures are *practical* with an aggregated signature size of roughly 11 KB. Our scheme naturally supports threshold up to 1024 participants, an upper limit of the "large" requirements of NIST preliminary call for threshold [PB23]. The main overhead is the offline phase where signers must exchange the preprocessing tokens with a size of a couple of hundred kilobytes. See Section 8.3 for more details.

1.2 Related Works

Other Post-Quantum Threshold Signatures. Bendlin, Krehbiel, and Peikert [BKP13] constructed a threshold signature based on the GPV signature [GPV08]. The protocol relies on generic multi-party computation (MPC) to perform Gaussian sampling. Khaburzaniya et al [KCLM22] recently proposed a threshold signatures from hash-based signatures using STARK. They report a signature of size 170 KB for a threshold of size 1024 signers, with an aggregation time of 4 to 20 seconds. While there are some isogeny-based threshold signatures [CS20, DM20], they only support sequential aggregation and thus requires numerous rounds to aggregate the signature.

Post-Quantum Multi-Signatures. Most closest to threshold signatures are multi-signatures. It can be viewed as an N-out-of-N threshold signature where each signers posses an individual signing key, rather than a secret share of one signing key. Unlike threshold signatures, constructing lattice-based multi-signatures has been more fruitful [FSZ22, DOTT21, DOTT22, BTT22, Che23]. The recent work by Boschini et al. [BTT22] and Chen [Che23] achieve a two-round protocol with signatures size roughly 100 KB and 30 KB, respectively. While Chen's protocol has smaller signature size, it does not offer offline-online efficiency as Boschnini et al's protocol.

Related Lattice Assumptions. We review two lattice-based assumptions that seem most similar to our AOM-MLWE assumption. The one-more-ISIS assumption was introduced by Agrawal et al. [AKSY22] to construct a blind signature. While the assumption includes the term "one-more" and is formalized as a one-more style assumption, it is qualitatively quite different from those considered in the classical setting like OM-DL. In essence, the assumption claims that given a lattice trapdoor $\mathbf{T} \in \mathbb{Z}^{m\times m}$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, i.e., $\mathbf{T}$ is short and $\mathbf{AT} = \mathbf{0} \mod q$, it is difficult to create a lattice trapdoor $\mathbf{T}'$ with a better quality than $\mathbf{T}$. Such notion of "quality" is lattice specific. The *hint* MLWE (Hint-MLWE) assumption was recently introduced by Kim et al. [KLSS23]. This assumption claims that the MLWE problem $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ remains hard even given *hints* $(c_i \cdot \mathbf{s} + \mathbf{z}_i, c_i \cdot \mathbf{e} + \mathbf{z}'_i)_{i\in[Q]}$, where c_i is some random small element and $(\mathbf{z}_i, \mathbf{z}'_i)$ are sampled from a discrete Gaussian distribution. When the samples $(\mathbf{z}_i, \mathbf{z}'_i)$ are super-polynomially larger than $(c_i \cdot \mathbf{s}, c_i \cdot \mathbf{e})$ it is clear that Hint-MLWE is as hard as MLWE. Kim et al. showed that even under milder conditions, Hint-MLWE are as hard as MLWE. While it shares similarity to AOM-MLWE since the adversary receives some information on the MLWE secret, the main difference is that in AOM-MLWE, the adversary obtains the *exact* value of the *adversarially chosen* inner product of the MLWE secrets.

Further Properties of Threshold Signatures. We consider only the key generation that should be executed by a trusted dealer in this paper. To avoid relying on the trusted dealer, we could use the distributed key generation (DKG), e.g., a lattice-based DKG with respect to a LWE-type verification key [GKS23,

ENP24]. We remain the DKG, that can be used for our scheme, and the security analysis of our schemes with a concrete DKG as interesting future works.

Our proposed scheme does not provide a way to detect misbehavior when a resulting signature is invalid. The identifiable abort (IA) is one of the well-known solutions to enable the detection of it. Specifically, participants execute the IA protocol and eventually identify the misbehavior when the signing protocol aborts or the resulting signature is invalid. The robustness is a property that ensures honest users can generate a valid signature, even in the presence of malicious users. Espitau et al. [ENP24] proposed a lattice-based robust threshold signature scheme by constructing verifiable short secret sharing. We also leave an IA protocol for our scheme and a robust signing protocol as important future works.

Concurrent Work. Chairattana-Apirom et al. [CATZ24] proposed an offline-online efficient two-round lattice-based threshold signature scheme without relying on heavy tools, independent of our result. Their scheme is constructed based on the variant of FROST [KG20] proposed by Tessaro and Zhu [TZ23], which is based on linear hash functions, and a linear secret sharing schemes with small coefficients. While the security is based on the standard lattice assumption MSIS, their signature size is around 220 KB (resp. 380 KB) for 5 (resp. 32) participants. This suggests a trade-off between the efficiency and the strength of the assumption, and an interesting open problem is to achieve the best of the two schemes.

Differences from the Conference Version. The previous version of this paper will appear in the 44th Annual International Cryptology Conference [EKT24]. The main differences between this version and the previous one are as follows:

- In Section 1.2, we added an overview of a concurrent work [CATZ24] that also proposes a lattice-based two-round threshold signature scheme.
- In Section 3, we included formal definitions of all the primitives used in the paper and added discussions on the definition of threshold signatures.
- In Section 4.1, we added more explanation on the AOM-MLWE assumption, e.g., why we need to consider the *algebraic* one-more MLWE assumption, not the *non-algebraic* one.
- In the previous version, we required the AOM-MLWE problem to be defined with respect to a matrix $\mathbf{A}$ having an invertible submatrix over $\mathcal{R}_q$. This was required to establish the hardness of (selective) AOM-MLWE based on the standard MSIS assumption. This negatively affected our resulting threshold signature as the public matrix was restricted to such $\mathbf{A}$. In this version, we remove this restriction by observing the specific reduction from the AOM-UMLWE problem to the AOM-MLWE problem, where the former is the AOM-MLWE problem with uniform secret.
- We added omitted proofs of Theorems 4.5 and 6.1 and Lemma 4.9 regarding the hardness of AOM-UMLWE.
- In Section 5, we first remove the aforementioned restriction on our public matrix $\mathbf{A}$. Second, we modified our threshold signature scheme by applying the optimization of masking introduced in [KRT24]. While the row mask was explicitly output as a part of the partial signature in the previous version, we no longer require this. In our modified scheme, each signer simply subtracts the row mask from the response $\mathbf{z}$. We revised our figures and explanations in Section 2. Moreover, we updated the security proof of our threshold signature scheme and added an omitted theorem for the correctness.
- In Section 7, we made a minor revision of the presentation of the attacks and gave more details on the way to avoid the divisibility condition in Section 7.3. Also, we included an overview of the core-SVP methodology in Section 7.5.
- In Section 8, we added more detailed explanations of parameter selection.
- In Appendix A, we provided a visual aid for masking technique of [PKM+24].
- In Appendix B, we showed an alternative reduction from MLWE and MSIS to sel-AOM-UMLWE, which we believe is beneficial for understanding the hardness of AOM-MLWE.

2 Technical Overview

We provide an overview of our offline-online efficient two-round threshold signature and establish its security based on the AOM-MLWE assumption. We then discuss the hardness of the assumption.

2.1 Two-Round Threshold Signature from AOM-MLWE

We first explain how we arrive at our threshold signature assuming AOM-MLWE is hard.

Base Signature Scheme. We use Lyubashevsky's lattice-based signature scheme [Lyu09, Lyu12] as our starting point. Let us briefly recall the protocol. Let $\mathbf{A} \in \mathcal{R}_q^{k\times\ell}$ and $\mathbf{t} = \mathbf{A}\cdot\mathbf{s}+\mathbf{e} \in \mathcal{R}_q^k$ for "short" vectors $(\mathbf{s},\mathbf{e})$. The verification and signing keys are set as $(\mathsf{vk},\mathsf{sk}) = ((\mathbf{A},\mathbf{t}),(\mathbf{s},\mathbf{e}))$. To sign a message M, the signer first constructs a *commitment* $\mathbf{w} = \mathbf{A}\cdot\mathbf{r}+\mathbf{e}'$, where $(\mathbf{r},\mathbf{e}')$ are "short" vectors sampled from some specific distribution. A *challenge* $c \leftarrow \mathsf{H}(\mathsf{vk},\mathsf{M},\mathbf{w})$, followed by a "short" *response* $(\mathbf{z},\mathbf{z}') := (c\cdot\mathbf{s}+\mathbf{r}, c\cdot\mathbf{e}+\mathbf{e}')$ is then computed. Finally, $(c,\mathbf{z},\mathbf{z}')$ is the signature. To verify, we check if $(\mathbf{z},\mathbf{z}')$ are short and that $c = \mathsf{H}(\mathsf{vk},\mathsf{M},\mathbf{A}\cdot\mathbf{z}+\mathbf{z}'-c\cdot\mathbf{t})$.

While it is standard to perform rejection sampling [Lyu09, Lyu12] to make the distribution of the responses independent of the signing key, we rely on noise "flooding" [GKPV10]. This allows the signers to never abort and works very well in the interactive setting. This is the approach also taken in recent lattice-based threshold signatures [ASY22, GKS23, PKM+24], using the Rényi divergence to granularly control the amount of noise flood required.

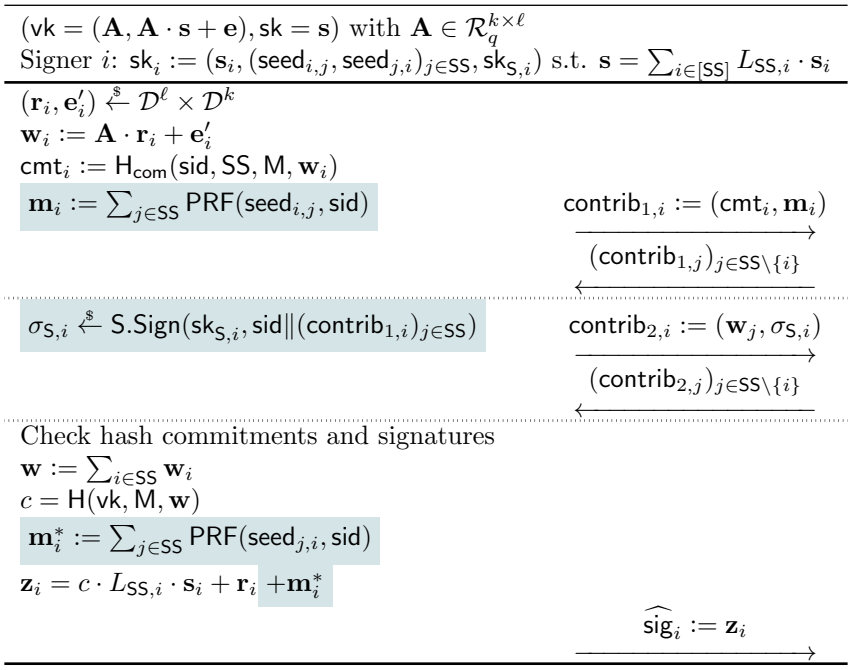

Figure 1: Simplified three-round threshold signature of [PKM+24]. $\mathsf{sid} \in \{0,1\}^*$ is a session identifier and $\mathsf{SS} \subseteq [N]$ is the set of active users. The second round is only initiated once signer i obtains $T = |\mathsf{SS}|$ first-round contributions. The final aggregated signature is $(c,\mathbf{z},\mathbf{h})$ where $\mathbf{z} := \sum_{i\in\mathsf{SS}}(\mathbf{z}_i - \mathbf{m}_i)$ and $\mathbf{h} := -\mathbf{A}\mathbf{z} + c\cdot\mathbf{t} + \mathbf{w}$. Notice that $\mathbf{h} = c\cdot\mathbf{e} + \sum_{i\in\mathsf{SS}}\mathbf{e}_i'$. The verification algorithm checks that $(\mathbf{z},\mathbf{h})$ are short and $c = \mathsf{H}(\mathsf{vk},\mathsf{M},\mathbf{A}\mathbf{z} - c\cdot\mathbf{t} + \mathbf{h})$. By ignoring the highlights in blue, we arrive at an insecure adaptation of a naive threshold Schnorr.

A Naive Extension to a Threshold Signature. One naive way to thresholdize Lyubashevsky's signature is to use Shamir's secret sharing protocol to share the signing key. This is depicted in Fig. 1 (ignoring the blue highlights). The partial signing key $(\mathbf{s}_i)_{i\in[N]}$ satisfy $\mathbf{s} = \sum_{i\in\mathsf{SS}} L_{\mathsf{SS},i}\cdot\mathbf{s}_i$ for any set $\mathsf{SS} \subset [N]$ with $|\mathsf{SS}| = T$,

where $L_{\mathsf{SS},i}$ is the Lagrange coefficient. Correctness follows from observing that $\mathbf{z} = \sum_{i\in\mathsf{SS}} \mathbf{z}_i = c \cdot \mathbf{s} + \mathbf{r}$, where $\mathbf{r} := \sum_{i\in\mathsf{SS}} \mathbf{r}_i$. It is worth mentioning that the signers need to perform a hash-and-open with the commitment $\mathbf{w}_i$ to force a malicious signer i^* to prepare its commitment $\mathbf{w}_{i^*}$ independently from the honest users' commitments. This is a procedure required for classical three-round schemes as well [BN06, CKM23a].

Unfortunately, it turns out this naive construction is insecure due to lattice-specific reasons. Since Lagrange coefficients can be arbitrarily large over modulo q, this forces the partial response $\mathbf{z}_i = c^* \cdot \mathbf{s}_i + \mathbf{r}_i$ to be large, where $c^* = c \cdot L_{\mathsf{SS},i}$. Similarly to why Lyubashevsky's signature becomes easily forgeable for large challenge spaces, the partial signing key $\mathbf{s}_i$ can be recovered from such a partial response using a large challenge c^*. While there are several workarounds to overcome large Lagrange coefficients, e.g. [ABV+12, BLMR13, BGG+18, LST18, BGG+18, DLN+21, AL21, ASY22, CSS+22], they are notorious for being highly impractical and/or non-scalable. For instance, one of the most simple and common approaches [ABV+12, BGG+18] require the modulus q to grow with at least $O(N!^2)$ — even for a small $N = 15$, we would require $q > 2^{80}$.[4]

Three-Round Threshold Signature by del Pino et al. Very recently, del Pino et al. [PKM+24] came up with a simple and elegant solution to sidestep this issue. Their idea is to additively *mask* the individual responses by a random vector and devise a way to publicly remove only the sum of the masks. This is depicted in Fig. 1. Each signer additionally shares a pair-wise seed for a pseudorandom function (PRF). In the first round, signer i now computes a so-called *row* mask $\mathbf{m}_i := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$ and shares it along with the hash commitment cmt_i, where sid is some unique string defined per session. In the third round, it computes a *column* mask $\mathbf{m}_i^* := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid})$ and adds this to the response $\mathbf{z}_i$. Importantly, while the row masks $(\mathbf{m}_i)_{i\in\mathsf{SS}}$ are public, the column masks $(\mathbf{m}_i^*)_{i\in\mathsf{SS}}$ are kept private. Moreover, by construction, we have $\sum_{j\in\mathsf{SS}} \mathbf{m}_j = \sum_{j\in\mathsf{SS}} \mathbf{m}_j^*$. To offset the column masks, we subtract $\sum_{j\in\mathsf{SS}} \mathbf{m}_i$ from $\sum_{j\in\mathsf{SS}} \mathbf{z}_i$ to arrive at the desired aggregated response $\mathbf{z} = c \cdot \mathbf{s} + \mathbf{r}$.

The key observation to understand the security is that while the individual row masks $(\mathbf{m}_j)_{j\in\mathsf{SS}}$ are known to the adversary, the only knowledge the adversary gains on the column masks $(\mathbf{m}_j^*)_{j\in\mathsf{HS}}$ of honest signers $\mathsf{HS} \subset \mathsf{SS}$ are their sum $\sum_{j\in\mathsf{HS}} \mathbf{m}_j^*$; put differently, $(\mathbf{m}_j^*)_{j\in\mathsf{HS}}$ are distributed randomly, conditioned on their sum being $\sum_{j\in\mathsf{HS}} \mathbf{m}_j^*$. This observation is leveraged to move around the terms $c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i$ included in the partial responses $\mathbf{z}_i$ of the honest signers, effectively allowing the reduction to reconstruct the signing key $\mathbf{s}$ *under the hood* of the adversary's view. (See Appendix A for a pictorial example.)

We note that the security proof is easier said than done. The main source of difficulty is that an adversary can adaptively alter the views of the honest signers without being detected. In the context of the above intuition, this means moving the terms $c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i$ around consistently with the adversary's view becomes very difficult. To this end, [PKM+24] requires a standard signature scheme to authenticate the view of each honest signer. Moreover, so as not to sign on the same sid, the signers must remain stateful.

Making it Two-Round. To turn the protocol into a two-round protocol, we collapse the seemingly superfluous second round, consisting of only opening the hash commitment. Recall we required this hash-and-open to prevent a malicious signer i^* from creating a commitment $\mathbf{w}_{i^*}$ affecting the aggregated commitment $\mathbf{w}$. We follow a similar high-level approach taken by FROST [KG20] to prevent this while removing the second round. Our two-round threshold signature is depicted in Fig. 2. In the first round, each signer now generates a list of commitments *in the clear*. In the second round, they use a hash function G modeled as a random oracle to compute a random weight $(\beta_b)_{b\in[\mathsf{rep}]}$ and (locally) set the partial commitment $\mathbf{w}_j$ as $\mathbf{w}_j := \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{w}_{j,b}$. Moreover, the row and column masks $(\mathbf{m}_i, \mathbf{m}_i^*)$ are now created in the second round at the same time from the PRF evaluated on input $\mathsf{ctnt} = \mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$. Importantly, we no longer require a session-specific identifier sid as in [PKM+24]. Also, by applying the optimization of masking introduced in [KRT24], the row mask $\mathbf{m}_i$ can be implicitly included in the response $\mathbf{z}_i$ as opposed to explicitly including it in the partial signature. Otherwise, it proceeds as before. Notice the first round *pre-processing token* pp_i can be generated without the knowledge of the message or set of signers, making the protocol offline-online efficient.

[4] While Albrecht and Lai [AL21, Section 3.1] define the Lagrange interpolating polynomial on specific elements in $\mathcal{R}_q$ to handle the blowup more granularly, the concrete gain is unclear for a general T-out-of-N threshold.

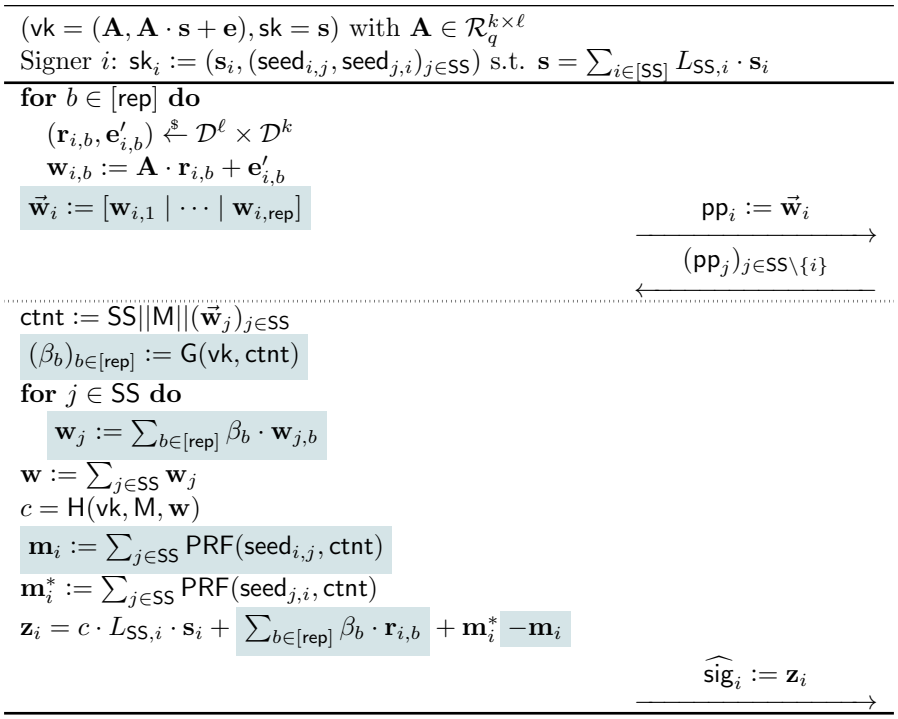

Figure 2: Our simplified offline-online efficient two-round threshold signature. The major differences between the three-round threshold signature in Fig. 1 are highlighted in blue. Concrete values of $\mathsf{rep} \in \mathbb{N}$ and the output of the hash function G is scheme specific, implicitly dictated by the parameters of the underlying AOM-MLWE assumption.

Before explaining the intuition of the security proof using AOM-MLWE, we note the effect of our modified mask evaluation. While it is a simple modification, including the commitments $(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$ in the PRF effectively "kills two birds with one stone". First of all, the signer no longer needs to maintain a state since a commitment $\vec{\mathbf{w}}_j$ has high min-entropy. That is, as long as the signers are correctly following the protocol, no adversary can trick them into using the same input to the PRF. This removes the need of using a session-specific identifier sid. Moreover, we are also able to remove the usage of standard signatures since $\mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$ and $\mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$ can be viewed as *random MACs* from signer i to j of the fact that i's view is ctnt, which effectively includes all the communication transcript. Noticing the role of signers i and j is symmetric, the random MAC embedded in the partial responses $\mathbf{z}_i$ and $\mathbf{z}_j$ cannot be removed unless both signers agree on the same ctnt. If ctnt agrees, then the reduction can move around the terms $c\cdot L_{\mathsf{SS},i}\cdot\mathbf{s}_i$ as explained prior. Otherwise, the responses remain random from the view of the adversary.

Security Proof with AOM-MLWE. It remains to explain how AOM-MLWE is used to prove security. The reduction is given $\mathbf{A}$, $\mathbf{t}$, and $(\mathbf{w}^{(k)}_{i,b})_{(k,i,b)\in[Q_\mathsf{S}]\times[N]\times[\mathsf{rep}]}$ as the challenge, where $\mathbf{t} = \mathbf{A}\mathbf{s}+\mathbf{e}$ and $\mathbf{w}^{(k)}_{i,b} = \mathbf{A}\mathbf{r}^{(k)}_{i,b} + \mathbf{e}^{(k)}_{i,b}$. The reduction sets $(\mathbf{A}, \mathbf{t})$ as the verification key and when the adversary invokes signer i on the k-th signing query, the reduction sets the pre-processing token as $\mathsf{pp}^{(k)}_i := \vec{\mathbf{w}}^{(k)} = (\mathbf{w}^{(k)}_{i,1}, \cdots \mathbf{w}^{(k)}_{i,\mathsf{rep}})$. Thanks to the above random MAC technique, we can guarantee the reduction to only be required to simulate partial responses of the form $\mathbf{z}_i = c\cdot\mathbf{s} + \sum_{b\in[\mathsf{rep}]}\beta_b\cdot\mathbf{r}^{(k)}_{i,b}$ + (public vector) or $\mathbf{z}_i = \sum_{b\in[\mathsf{rep}]}\beta_b\cdot\mathbf{r}^{(k)}_{i,b}$ + (public vector). Thus the reduction only needs to query the linear combination $(c, 0, \cdots, \beta_1, \cdots, \beta_{\mathsf{rep}}, \cdots, 0)$ or $(0, 0, \cdots, \beta_1, \cdots, \beta_{\mathsf{rep}}, \cdots, 0)$ to the MLWE solving oracle to simulate these partial responses.

The technically interesting part is what the reduction does once the adversary outputs a forgery. As typical with any signatures based on identification protocols, by relying on the forking lemma [PS00, BN06],

we can extract an (approximate) MLWE solution $(\mathbf{s}, \mathbf{e})$ relative to the verification key $\mathbf{t}$.[5] The difference between standard proofs is that the reduction's goal is to break AOM-MLWE, defined by the $\mathsf{rep} \cdot Q_{\mathsf{S}} \cdot N + 1$ MLWE instances. Observe that in the course of simulating the adversary, the reduction may make up to $2 \cdot Q_{\mathsf{S}} \cdot N$ queries to the MLWE solving oracle, where the factor 2 comes from running the adversary twice. We then require $\mathsf{rep} \geq 2$ at the minimum to non-trivialize the game as the reduction cannot query more than the number of challenges it receives. It is relatively easy to show that when $\mathsf{rep} \geq 2$, the reduction can (approximately) compute all of $(\mathbf{s}, \mathbf{e}), (\mathbf{r}_{i,b}^{(k)}, \mathbf{e}_{i,b}^{(k)})_{(k,i,b) \in [Q_{\mathsf{S}}] \times [N] \times [\mathsf{rep}]}$ from the partial responses and adversary's forgery, satisfying the winning condition of AOM-MLWE.

The only thing missing from our proof is establishing the hardness of the underlying AOM-MLWE assumption. The above does not yet tell us anything about how we should set $\mathsf{rep} \geq 2$, what the noise distributions should be, or what should the allowable linear combinations to the MLWE solving oracles be.

2.2 Analyzing Hardness of AOM-MLWE

In the classical setting, the hardness of the algebraic one-more discrete logarithm (AOM-DL) problem [NRS21] is easy-to-state and well-established. It is a strictly harder problem than the (non-algebraic) OM-DL problem, already widely believed to be difficult. Indeed, AOM-DL can be shown to be hard in the generic group model (GGM) [Sho97, Mau05].

Theoretical Hardness of AOM-MLWE. The situation vastly changes when looking at the algebraic one-more MLWE (AOM-MLWE) problem. We do not have an already established (non-algebraic) OM-LWE problem to base hardness on or any idealized model such as the GGM to formally argue its hardness. In fact, the problem becomes trivially insecure if we naively define AOM-MLWE. However, this is not unsuspected as MLWE already exhibits a similar phenomenon; one can always set the parameters for MLWE so that it becomes trivially insecure. The added complexity of analyzing AOM-MLWE comes from the need to take into account the extra information an adversary learns by querying the MLWE solving oracle.

Let us give a very simple example. Assume the AOM-MLWE challenge is $\mathbf{A}, (\mathbf{t}_i = \mathbf{A} \cdot \mathbf{s}_i + \mathbf{e}_i)_{i \in [2]}$ such that the secrets have infinity norm smaller than B. Then, the adversary can query the linear combination $(1, B)$ to the MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$ to obtain $\mathbf{s}_1 + B \cdot \mathbf{s}_2$. If $B \ll q$, then by taking modulo B, the adversary easily recovers $\mathbf{s}_1$ and $\mathbf{s}_2$. Since it solves two MLWE instances with one query, it breaks AOM-MLWE. In Section 4, we present less obvious "weak" parameters for which AOM-MLWE admits a more sophisticated attack.

We then turn all our findings on the weak parameters of AOM-MLWE into a constructive argument to establish the hardness of AOM-MLWE. Specifically, we provide several sets of "hard" parameters and prove that the *selective* AOM-MLWE is as hard as the standard MLWE and MSIS problems. Here, selective security means that the adversary must commit to all the linear combinations it queries to oracle $\mathcal{O}_{\mathsf{solve}}$ at the outset of the game. This establishes that to break AOM-MLWE, an adversary must cleverly use $\mathcal{O}_{\mathsf{solve}}$ in an *adaptive* manner. While our result does not formally say anything about the adaptive security of AOM-MLWE, it illustrates that there is nothing fundamentally wrong with the hard parameters. We draw a parallel between this situation to the numerous lattice-based primitives only proven selectively secure but are plausibly adaptively secure, e.g. [ABB10, GVW13, BGG+14, GV15, GVW15a, GVW15b]. Considering that most natural selectively secure cryptographic primitives are plausibly adaptively secure, it would be highly interesting to see any attack exploiting the adaptive nature of AOM-MLWE. We leave it as an important theoretical question to bridge selective and adaptive security, oftentimes very easy to establish in the classical setting using idealized models such as GGM.

Practical Hardness of AOM-MLWE. Lastly, we complement our theoretical analysis of AOM-MLWE with practical cryptanalysis. To provide a basic understanding of the techniques introduced, we present another simple attack, this one being purely statistical. Suppose we are allowed $Q - 1$ queries on the challenge $\mathbf{A}, (\mathbf{t}_i = \mathbf{A} \cdot \mathbf{s}_i + \mathbf{e}_i)_{i \in [Q]}$, where all secrets and errors have a norm bounded by B. We request all the $(\mathbf{s}_1 + \mathbf{s}_i, \mathbf{e}_1 + \mathbf{e}_i)_{i \in [2,Q]}$, such that summing them gives us the values of $(Q-1)\mathbf{s}_1 + \sum_i \mathbf{s}_i$ and $(Q-1)\mathbf{e}_1 + \sum_i \mathbf{e}_i$. These two equations effectively position $\mathbf{s}_1$ and $\mathbf{e}_1$ within balls of radius $\frac{B}{\sqrt{Q}}$. This estimation is highly precise

[5] For the attentive readers, since the reduction is playing an interactive game with AOM-MLWE, we must rely on a variant of the forking lemma with oracle access [EK18].

for large Q, enabling the reconstruction of $\mathbf{s}_1, \mathbf{e}_1$ and derivation of all other values using elementary linear algebra. We demonstrate that this attack can be generalized to provide statistical information on all the secrets and errors of the AOM-MLWE instance. By meticulously analyzing the geometry of generic queries, we can gather enough information to pinpoint the errors and secrets within a specific region of the space. Subsequently, we craft a final MLWE instance to decode these within the identified regions and solve the instance. In the context of our threshold signature schemes, we illustrate that the practical security of forgery, after collecting Q transcripts, is equivalent to solving an MLWE instance with parameters that are $\frac{1}{\sqrt{W \cdot Q}}$ times smaller than for a direct forgery. Here, W is the hamming weight of a challenge polynomial. Integrating this cryptanalysis with state-of-the-art lattice reduction estimation allows us to construct a set of parameters that align with the standard NIST levels I, III, and V.

3 Preliminary

3.1 Notations

We use lower (resp. upper) case bold fonts $\mathbf{v}$ (resp. $\mathbf{M}$) for vectors (resp. matrices). We always view vectors in the column form. We use v_i (resp. $\mathbf{m}_i$) to indicate the i-th entry (resp. column) of $\mathbf{v}$ (resp. $\mathbf{M}$). For $(\mathbf{v}, \mathbf{M}) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^{k \times \ell}$, $\mathbf{v}^\top \odot \mathbf{M}$ denotes the column-wise multiplication: $[v_1 \cdot \mathbf{m}_1 \mid \cdots \mid v_\ell \cdot \mathbf{m}_\ell]$. For $\mathbf{M} = [\mathbf{m}_1 \mid \cdots \mid \mathbf{m}_\ell] \in \mathcal{R}_q^{k \times \ell}$, $\|\mathbf{M}\|_2$ denotes $\max_{i \in [\ell]} \|\mathbf{m}_i\|_2$.

3.2 Lattices and Gaussians

For integers $n, q \in \mathbb{N}$, we define the ring $\mathcal{R}$ as $\mathbb{Z}[X]/(X^n + 1)$ and $\mathcal{R}_q$ as $\mathcal{R}/q\mathcal{R}$. For a positive real σ, let $\rho_\sigma(\mathbf{z}) = \exp\left(-\frac{\|\mathbf{z}\|_2}{2\sigma^2}\right)$. The discrete Gaussian distribution over $\mathbb{Z}^n$ and standard deviation σ is defined by its probability distribution function: $\mathcal{D}_{\mathbb{Z}^n,\sigma}(\mathbf{z}) = \frac{\rho_\sigma(\mathbf{z})}{\sum_{\mathbf{z}' \in \mathbb{Z}^n} \rho_\sigma(\mathbf{z}')}$. We may simply note $\mathcal{D}_\sigma$.

We denote $\mathcal{C} \subset \mathcal{R}_q$ as the set of polynomials with $\{-1, 0, 1\}$-coefficient and fixed hamming weight W, i.e., $\{c \in \mathcal{R}_q \mid \|c\|_\infty = 1 \wedge \|c\|_1 = W\}$. We denote $\mathbb{T} \subset \mathcal{R}_q$ as the set of all signed monomials, i.e., $\{(-1)^b \cdot X^i \mid (b, i) \in \{0, 1\} \times [n]\}$. We have the following guarantee on invertibility of differences of elements in $\mathbb{T}$ (see for example [BCK+14]).

Lemma 3.1. *Let n be a power of 2. For any distinct $a, b \in \mathbb{T}$, $(a-b)$ is invertible over $\mathcal{R}_q$ and $2 \cdot (a-b)^{-1} \in \mathcal{R}_q$ is a polynomial with coefficients in $\{-1, 0, 1\}$.*

The following is a tail-cut bound on discrete Gaussian distributions from [PKM+24]. It immediately follows from combining standard tail-cut bounds [MR04, Lyu12] with the Minkowski's inequality.

Lemma 3.2. *For $\mathbf{s} \xleftarrow{\$} \mathcal{D}_\sigma^k$ and $v \in \mathcal{R}$, we have*

$$\Pr\left[\|v \cdot \mathbf{s}\|_2 \geq e^{1/4} \|v\|_1 \sigma \cdot \sqrt{nk}\right] \leq 2^{-\frac{nk}{10}}.$$

The following follows from [MR04, GMPW20].

Lemma 3.3. *Let T be a positive integer and $\sigma > \sqrt{\frac{\log(2n)+\lambda}{\pi}}$. Then, the distribution of $x := \sum_{i \in T} x_i$ for $x \xleftarrow{\$} \mathcal{D}_\sigma$ is within statistical distance $2^{-\lambda}$ of the distribution $x \xleftarrow{\$} \mathcal{D}_{\sqrt{T} \cdot \sigma}$.*

3.3 Rényi Divergence

The Rényi divergence [Rén61] is a tool from information theory which has recently found many applications in lattice-based cryptography, see for instance [BLL+15, Pre17]. We use the "exponential form" of the Rényi divergence, as it is common in lattice-based cryptography.

Definition 3.4 (Rényi divergence). *Let $\mathcal{P}, \mathcal{Q}$ be two discrete distributions such that* $\mathsf{Supp}(\mathcal{P}) \subseteq \mathsf{Supp}(\mathcal{Q})$*, and $\alpha \in (1; +\infty)$. The Rényi divergence of order α is:*

$$R_\alpha(\mathcal{P}; \mathcal{Q}) = \left(\sum_{x \in X} \frac{\mathcal{P}(x)^\alpha}{\mathcal{Q}(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}}$$

Following Csiszár's f-divergence framework [Csi63], $(R_\alpha^{\alpha-1} - 1)$ is an f-divergence for $f : x \mapsto x^\alpha - 1$. Lemma 3.5 presents some properties of the Rényi divergence; proofs can be found in van Erven and Harremoës [vEH14] or Bai et al. [BLR+18].

Lemma 3.5. *For distributions $\mathcal{P}, \mathcal{Q}$ and finite families of independent distributions $(\mathcal{P}_i)_{i \in [n]}, (\mathcal{Q}_i)_{i \in [n]}$, the Rényi divergence satisfies the following properties:*

1. ***Data processing inequality.*** *For a (randomized) function f,*

$$R_\alpha(f(\mathcal{P}); f(\mathcal{Q})) \leq R_\alpha(\mathcal{P}; \mathcal{Q}).$$

2. ***Probability preservation.*** *For any event $E \subseteq \mathsf{Supp}(\mathcal{Q})$:*

$$\mathcal{P}(E) \leq \mathcal{Q}(E)^{\frac{\alpha-1}{\alpha}} \cdot R_\alpha(\mathcal{P}; \mathcal{Q}),$$

3. ***Multiplicativity.*** $R_\alpha(\prod_i \mathcal{P}_i; \prod_i \mathcal{Q}_i) = \prod_i R_\alpha(\mathcal{P}_i; \mathcal{Q}_i)$.

For discrete Gaussian distributions, we have the following [LSS14].

Lemma 3.6. *Let $\alpha \geq 2$ be an integer and $\mathbf{v} \in \mathcal{R}$. It holds that:*

$$R_\alpha(\mathcal{D}_{\sigma,\mathbf{v}}; \mathcal{D}_\sigma) = \exp\left(\frac{\alpha \|\mathbf{v}\|_2^2}{2\sigma^2} \right).$$

Due to the symmetry of discrete Gaussian distribution, we also have the same bound on $R_\alpha(\mathcal{D}_\sigma; \mathcal{D}_{\sigma,\mathbf{v}})$.

3.4 Linear Secret Sharing

We recall the *linear Shamir secret sharing* scheme [Sha79b]. Let $N < q$ be an integer such that for distinct $i, j \in [N]$, $(i - j)$ is invertible over $\mathbb{Z}_q$. Let $S \subseteq [N]$ be a set of cardinality at least T. Then, given $i \in S$, we define the Lagrange coefficient $L_{S,i}$ as

$$L_{S,i} := \prod_{j \in S \setminus \{i\}} \frac{-j}{i - j}.$$

Let $s \in \mathcal{R}_q$ be a secret to be shared, $P \in \mathcal{R}_q[X]$ a degree $T - 1$polynomial such that $P(0) = s$. Given any set of evaluation points $E = \{(i, y_i)\}_{i \in S}$ such that $y_i = P(i)$ for all $i \in S$, we note that

$$s = \sum_{i \in S} L_{S,i} \cdot y_i.$$

The notations naturally extend to secrets that are in vector form. With a slight abuse of notation, we say $\vec{P} \in \mathcal{R}_q^\ell[X]$ is of degree $T - 1$ if each entry of $\vec{P}$ is a degree $T - 1$ polynomial. Moreover, $\vec{P}(x)$ denotes the evaluation of each entry of $\vec{P}$ on the point x.

3.5 Pseudorandom Function

Definition 3.7. *Let* $\mathsf{PRF} := \{\mathsf{PRF}_\lambda : \{0,1\}^\lambda \times \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{n(\lambda)}\}_{\lambda\in\mathbb{N}}$ *be a function family. We say* PRF *is a pseudorandom function if for any efficient adversary* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(1^\lambda)$ *defined below is negligible:*

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(1^\lambda) := \left|\Pr[\mathcal{A}(1^\lambda)^{\mathsf{PRF}_\lambda(\mathsf{seed},\cdot)} : \mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda] - \Pr[\mathcal{A}(1^\lambda)^{\mathsf{Rand}(\cdot)} : \mathsf{Rand} \xleftarrow{\$} \mathsf{Func}(\{0,1\}^\ell, \{0,1\}^n)]\right|,$$

where $\mathsf{Func}(\mathcal{X},\mathcal{Y})$ *denotes the set of all functions from* $\mathcal{X}$ *to* $\mathcal{Y}$*.*

For notational simplicity, we may omit the subscript λ what follows.

3.6 Two-Round Threshold Signature

A two-round threshold signature scheme consists of the following efficient algorithms. Let N be the number of total signers and T be a reconstruction threshold s.t. $T \le N$. Also, let SS be a signer set such that $\mathsf{SS} \subseteq [N]$ with size T. Each signer $i \in [N]$ maintains a state st_i to retain a short-lived session specific information (see Remark 3.11).

$\mathsf{TS.Setup}(1^\lambda, N, T) \to \mathsf{tspar}$**:** The setup algorithm takes as input a security parameter 1^λ, the number N of total signers, and a reconstruction threshold $T \le N$ and outputs a public parameter tspar. We assume tspar includes N and T.

$\mathsf{TS.KeyGen}(\mathsf{tspar}) \to (\mathsf{vk}, (\mathsf{sk}_i)_{i\in[N]})$**:** The key generation algorithm takes as input a public parameter tspar and outputs a verification key vk and secret key shares $(\mathsf{sk}_i)_{i\in[N]}$. It implicitly sets up an empty state $\mathsf{state}_i := \emptyset$ for all N signers. We assume vk includes tspar.

$\mathsf{TS.PP}(\mathsf{vk}, i, \mathsf{sk}_i, \mathsf{st}_i) \to (\mathsf{pp}_i, \mathsf{st}_i)$**:** The signing algorithm for a pre-processing round takes as input a verification key vk, an index i of a signer, a secret key share sk_i, and a state st_i of the signer i, and outputs a pre-processing token pp_i and an updated state st_i.

$\mathsf{TS.Sign}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}}, \mathsf{sk}_i, \mathsf{st}_i) \to (\widehat{\mathsf{sig}}_i, \mathsf{st}_i)$**:** The signing algorithm takes as input a verification key vk, a signer set SS, a message M, an index $i \in \mathsf{SS}$ of a signer, a tuple of pre-processing tokens $(\mathsf{pp}_j)_{j\in\mathsf{SS}}$, a secret key share sk_i, and a state st_i of the signer i and outputs a partial signature $\widehat{\mathsf{sig}}_i$ and an updated state st_i.

$\mathsf{TS.Agg}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, (\widehat{\mathsf{sig}}_i)_{i\in\mathsf{SS}}) \to \mathsf{sig}$**:** The aggregation algorithm takes as input a verification key vk, a signer set SS, a message M, and a tuple of partial signatures $(\widehat{\mathsf{sig}}_i)_{i\in\mathsf{SS}}$ and outputs a signature sig.

$\mathsf{TS.Verify}(\mathsf{vk}, \mathsf{M}, \mathsf{sig}) \to 1$ or 0**:** The verification algorithm takes as input a verification key vk, a message M, and a signature sig and outputs 1 if sig is valid and 0 otherwise.

Below, we define the correctness of a two-round threshold signature scheme.

Definition 3.8 (Correctness). *We say that a two-round threshold signature scheme* TS *satisfies correctness if, for all* $\lambda \in \mathbb{N}$*,* $N, T \in \mathsf{poly}(\lambda)$ *s.t.* $T \le N$*, message* M*, and* $\mathsf{SS} \subseteq [N]$ *s.t.* $|\mathsf{SS}| = T$*, the following holds:*

$$\Pr\left[\mathsf{Game}^{\mathsf{ts\text{-}cor}}_{\mathsf{TS}}(1^\lambda, N, T, \mathsf{M}, \mathsf{SS}) = 1\right] \ge 1 - \mathsf{negl}(\lambda),$$

where $\mathsf{Game}^{\mathsf{ts\text{-}cor}}_{\mathsf{TS}}$ *is shown in Fig. 3.*

Now we define the unforgeability for a two-round threshold signature scheme.

$\mathsf{Game}^{\mathsf{ts\text{-}cor}}_{\mathsf{TS}}(1^\lambda, N, T, \mathsf{M}, \mathsf{SS})$

1 : **for** $i \in \mathsf{SS}$ **do** $\mathsf{st}_i := \emptyset$
2 : $\mathsf{tspar} \xleftarrow{\$} \mathsf{TS.Setup}(1^\lambda, N, T)$
3 : $(\mathsf{vk}, (\mathsf{sk}_i)_{i\in[N]}) \xleftarrow{\$} \mathsf{TS.KeyGen}(\mathsf{tspar})$
4 : **for** $i \in \mathsf{SS}$ **do**
5 : $\quad(\mathsf{pp}_i, \mathsf{st}_i) \xleftarrow{\$} \mathsf{TS.PP}(\mathsf{vk}, i, \mathsf{sk}_i, \mathsf{st}_i)$
6 : **for** $i \in \mathsf{SS}$ **do**
7 : $\quad(\widehat{\mathsf{sig}}_i, \mathsf{st}_i) \xleftarrow{\$} \mathsf{TS.Sign}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}}, \mathsf{sk}_i, \mathsf{st}_i)$
8 : $\mathsf{sig} \xleftarrow{\$} \mathsf{TS.Agg}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, (\widehat{\mathsf{sig}}_i)_{i\in\mathsf{SS}})$
9 : **return** $\mathsf{TS.Verify}(\mathsf{vk}, \mathsf{M}, \mathsf{sig})$

Figure 3: Correctness game for a two-round threshold signature scheme.

$\mathsf{Game}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, N, T)$

1 : $\mathsf{Q}_\mathsf{M} := \emptyset$ // Empty set
2 : $\mathsf{tspar} \xleftarrow{\$} \mathsf{TS.Setup}(1^\lambda, N, T)$
3 : $(\mathsf{CS}, \mathsf{st}_\mathcal{A}) \xleftarrow{\$} \mathcal{A}^{\mathsf{H}}(\mathsf{tspar})$
4 : **req** $[\![\mathsf{CS} \subseteq [N]]\!] \wedge [\![|\mathsf{CS}| \leq T-1]\!]$
5 : $\mathsf{HS} := [N] \backslash \mathsf{CS}$,
6 : **for** $i \in \mathsf{HS}$ **do** $\mathsf{st}_i := \emptyset$
7 : $(\mathsf{vk}, (\mathsf{sk}_i)_{i\in[N]}) \xleftarrow{\$} \mathsf{TS.KeyGen}(\mathsf{tspar})$
8 : $(\mathsf{sig}^*, \mathsf{M}^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{TS.PP}}, \mathcal{O}_{\mathsf{TS.Sign}}, \mathsf{H}}(\mathsf{vk}, (\mathsf{sk}_i)_{i\in\mathsf{CS}}, \mathsf{st}_\mathcal{A})$
9 : **if** $[\![\mathsf{M}^* \in \mathsf{Q}_\mathsf{M}]\!]$ **then**
10 : $\quad$ **return** 0
11 : **return** $\mathsf{TS.Verify}(\mathsf{tspar}, \mathsf{vk}, \mathsf{M}^*, \mathsf{sig}^*)$

$\mathcal{O}_{\mathsf{TS.PP}}(i)$

1 : **req** $[\![i \in \mathsf{HS}]\!]$
2 : $(\mathsf{pp}_i, \mathsf{st}_i) \xleftarrow{\$} \mathsf{TS.PP}(\mathsf{vk}, i, \mathsf{sk}_i, \mathsf{st}_i)$
3 : **return** pp_i

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}})$

1 : **req** $[\![\mathsf{SS} \subseteq [N]]\!] \wedge [\![i \in \mathsf{HS} \cap \mathsf{SS}]\!]$
2 : $\widehat{\mathsf{sig}}_i \xleftarrow{\$} \mathsf{TS.Sign}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}}, \mathsf{sk}_i, \mathsf{st}_i)$
3 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$
4 : **return** $\widehat{\mathsf{sig}}_i$

Figure 4: Unforgeability game for a two-round threshold signature scheme in the random oracle model, where H denotes the random oracle. In the above, the oracles return $\perp$ to $\mathcal{A}$ when $\mathsf{TS.PP}$ or $\mathsf{TS.Sign}$ output $\perp$ (i.e., fail to output a pre-processing token or a partial signature).

Definition 3.9 (Unforgeability). *For a two-round threshold signature scheme* TS*, the advantage of an adversary* $\mathcal{A}$ *against the unforgeability of* TS *in the random oracle model is defined as*

$$\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, N, T) = \Pr[\mathsf{Game}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, N, T) = 1],$$

where $\mathsf{Game}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, N, T)$ *is described in Fig. 4. We say that* TS *is unforgeable in the random oracle model if, for all* $\lambda \in \mathbb{N}$*,* $N, T \in \mathsf{poly}(\lambda)$ *s.t.* $T \leq N$*, and efficient adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, N, T) = \mathsf{negl}(\lambda)$ *holds.*

Remark 3.10 (Other Definitions). Our definition is equivalent to those of Bellare et al. [BCK+22], with the only difference that we exclude the *leader*. Bellare et al. assumes a communication model where a leader explicitly relays the communications between the signers. Unforgeability then captures the corruption of a leader. In contrast, we keep the communication model agnostic and exclude the leader for simplicity. This is without loss of generality as our unforgeability allows an adversary to control the communication channel between signeres. Indeed, it is easy to check that the unforgeability of [BCK+22] is equivalent to ours.[6]

[6] In [BCK+22], there are several definitions, depending on the difference of the trivial forgery. Our unforgeability is equivalent

Remark 3.11 (Session State). It is worth noting that while the signers maintain a state, our protocol only requires a short-lived state only specific to a particular session — this is a minimal requirement for any interactive protocol and is consistent with prior definitions. In contrast, a long-lived state is for example where the signer must keep track of all the previous messages being signed in different sessions. For instance, the three-round threshold signature by del Pino et al. [PKM+24] requires a session identifier $\mathsf{sid} \in \{0,1\}^*$ for each session and assumes the signers never sign the same sid.

Remark 3.12 (Q_{S}-Bounded Scheme). While our construction of threshold signature supports an unbounded polynomially many signing queries, we would require a super-polynomial sized modulus q, making the scheme impractical. To this end, we consider a more fined grained bounded scheme where unforgeability holds against any adversary making at most $Q_{\mathsf{S}} = \mathsf{poly}(\lambda)$ signing queries to $\mathcal{O}_{\mathsf{TS.Sign}}$. As per NIST's 2022 call for additional (post-quantum) signatures [NIS22], we set $Q_{\mathsf{S}} \approx 2^{64}$ for our concrete instantiation. Indeed, this is common practice among practical signatures such as Falcon [PFH+22] and Raccoon [dPEK+23], including the three-round threshold Raccoon [PKM+24].

3.7 Rounding and Norms Modulo q

This subsection is taken almost verbatim from [PKM+24]. In all of the following we fix positive integers q and n. We aim at giving a systematic treatment of the adaptation of the notions of norms and rounding maps to the ring of integers modulo q, $\mathbb{Z}_q$ and more generally in the free module $\mathbb{Z}_q^n$ of vectors mod q.

3.7.1 Length over Modular Integers

In this work, we use the so-called *canonical* unsigned representation of integers modulo q. Given an integer $x \in \mathbb{Z}$, this representation is the unique non-negative element $0 \leq t \leq q-1$ such that $x = t \mod q$. We will generically note this element $(x \mod q)$. Conversely, given a class $x + q\mathbb{Z} \in \mathbb{Z}_q$, we define the corresponding lift $\bar{x}$ to the unique integer in $x + q\mathbb{Z} \cap [0, \dots q-1]$.

For any norm $\|\cdot\|$ over $\mathbb{Q}^n$, we define the *length* of a (vector) class $\mathbf{x} + q\mathbb{Z}^n$ to be $\min_{\mathbf{z} \in \mathbf{x} + q\mathbb{Z}^n} \|\mathbf{z}\|$, and overload the notation as $\|\mathbf{x} + q\mathbb{Z}^n\|$, $\|\mathbf{x} \mod q\|$ or even $\|\mathbf{x}\|$ if the context is clear enough to avoid any ambiguity. As for the integers, we prefer to write simply $|x|$ when $n = 1$ to refer to the absolute value. [PKM+24] show that with the choices in this definition, $\|\cdot\|$ is indeed a *F-norm* over free modules over $\mathbb{Z}_q$. The only non-trivial point to show is the triangular inequality.

Lemma 3.13. *For any $q, n \in \mathbb{N} \setminus \{0\}$, and $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, we have*

$$|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|.$$

3.7.2 Modular Most-Significant Bit Decomposition

Let $\nu \in \mathbb{N} \setminus \{0\}$. Any integer $x \in \mathbb{Z}$ can be *uniquely* decomposed as:

$$x = 2^\nu \cdot x_\top + x_\bot, \quad (x_\top, x_\bot) \in \mathbb{Z} \times [-2^{\nu-1}, 2^{\nu-1} - 1], \tag{1}$$

which consists essentially in separating the lower-order bits from the higher-order ones. We define the function

$$\lfloor \cdot \rceil_\nu : \mathbb{Z} \to \mathbb{Z} \quad \text{s.t.} \quad \lfloor x \rceil_\nu = \lfloor x/2^\nu \rceil = x_\top,$$

where $\lfloor \cdot \rceil : \mathbb{R} \mapsto \mathbb{Z}$ denotes the rounding operator. More precisely the "rounding half-up" method $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor$ where half-way values are rounded up: e.g. $\lfloor 2.5 \rceil = 3$ and $\lfloor -2.5 \rceil = -2$. With a slight overload of notation, when $q > 2^\nu$, we extend $\lfloor \cdot \rceil_\nu$ to take inputs in $\mathbb{Z}_q$, in which case, we assume the output is an element in $\mathbb{Z}_{q_\nu}$ where $q_\nu = \lfloor q/2^\nu \rfloor$. Formally, we define:

$$\lfloor \cdot \rceil_\nu : \mathbb{Z}_q \mapsto \mathbb{Z}_{q_\nu} = \mathbb{Z}_{\lfloor q/2^\nu \rfloor} \quad \text{s.t.} \quad \lfloor x \rceil_\nu = \lfloor \bar{x}/2^\nu \rceil + q_\nu\mathbb{Z} = (\bar{x})_\top + q_\nu\mathbb{Z},$$

to TS-UF-0, which regards a forgery on a message as trivial if the message is queried to the signing oracle.

The function $\lfloor \cdot \rceil_\nu$ naturally extends to vectors coefficient-wise. The following is a special case of [PKM+24]. This bound on modular rounding operations are useful when arguing the small offset caused by performing modular rounding for efficiency.

Lemma 3.14. *Let ν, q be positive integers such that $q > 2^\nu$, $\nu \geq 4$, and set $q_\nu = \lfloor q/2^\nu \rfloor$. Moreover, assume q and ν satisfy $q_\nu = \lfloor q/2^\nu \rceil$, that is, q can be decomposed as $q = 2^\nu \cdot q_\nu + q_\perp$ for $q_\perp \in [0, 2^{\nu-1} - 1]$. Then, for any $x \in \mathbb{Z}_q$, we have*

$$\left| x - 2^\nu \cdot \overline{\lfloor x \rceil_\nu} \right| \leq 2^\nu - 1. \tag{2}$$

Moreover, for any $\mathbf{x}, \boldsymbol{\delta} \in \mathbb{Z}_q^n$, we have

$$\left\| 2^\nu \cdot \left(\overline{\lfloor \mathbf{x} + \boldsymbol{\delta} \rceil_\nu - \lfloor \mathbf{x} \rceil_\nu} \right) \mod q \right\| \leq \left\| 2^\nu \cdot \overline{\lfloor \boldsymbol{\delta} \rceil_\nu} \mod q \right\| + \|\mathbf{1}\| \cdot 2^\nu. \tag{3}$$

In the remainder of the paper, we will not be as precise as above for better readability. For instance, we might informally use x instead of the lift $\bar{x}$ or write $|2^\nu \cdot x|$ instead of $|2^\nu \cdot \bar{x} \mod q|$ when the context is clear and the distinction is unimportant.

3.8 Hardness Assumptions

We review some standard lattice-based hardness assumptions.

Definition 3.15 (MLWE). *Let ℓ, k, q be integers and $\mathcal{D}$ be a probability distribution over $\mathcal{R}_q$. The advantage of an adversary $\mathcal{A}$ against the* Module Learning with Errors $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ *problem is defined as:*

$$\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{A}}(1^\lambda) = |\Pr[1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e})] - \Pr[1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b})]|$$

where $(\mathbf{A}, \mathbf{b}, \mathbf{s}, \mathbf{e}) \leftarrow \mathcal{R}_q^{k \times \ell} \times \mathcal{R}_q^k \times \mathcal{D}^\ell \times \mathcal{D}^k$. The $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ assumption states that any efficient adversary $\mathcal{A}$ has negligible advantage. We may write $\mathsf{MLWE}_{q,\ell,k,\sigma}$ as a shorthand for $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ when $\mathcal{D}$ is the Gaussian distribution of standard deviation σ. Lastly, we also define a variant called uniform MLWE (UMLWE) *where the secret key is sampled from the uniform distribution $\mathcal{R}_q^\ell$.*

Definition 3.16 (MSIS). *Let ℓ, k, q be integers and $\beta > 0$ a real number. The advantage of an adversary $\mathcal{A}$ against the* Module Short Integer Solution $\mathsf{MSIS}_{q,\ell,k,\beta}$ *problem, is defined as:*

$$\mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{A}}(1^\lambda) = \Pr\left[\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}, \mathbf{s} \xleftarrow{\$} \mathcal{A}(\mathbf{A}) : (0 < \|\mathbf{s}\|_2 \leq \beta) \ \wedge \ \left[\mathbf{A} \mid \mathbf{I}\right] \mathbf{s} = \mathbf{0} \mod q \right].$$

The $\mathsf{MSIS}_{q,\ell,k,\beta}$ assumption states that any efficient adversary $\mathcal{A}$ has negligible advantage.

Lemma 3.17 (Hardness of MLWE ([LS15])). *Let $k(\lambda)$, $\ell(\lambda)$, $q(\lambda)$, $n(\lambda)$, $\sigma(\lambda)$ such that $q \leq \mathsf{poly}(n\ell)$, $k \leq \mathsf{poly}(\ell)$, and $\sigma \geq \sqrt{\ell} \cdot \omega(\sqrt{\log n})$. If $\mathcal{D}$ is a discrete Gaussian distribution with standard deviation σ, then the $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ problem is as hard as the worst-case lattice Generalized-Independent-Vector-Problem (GIVP) in dimension $N = n\ell$ with approximation factor $\sqrt{8 \cdot N\ell} \cdot \omega(\sqrt{\log n}) \cdot q/\sigma$.*

Lemma 3.18 (Hardness of MSIS([LS15])). *For any $k(\lambda), \ell(\lambda), q(\lambda), n(\lambda), \beta(\lambda)$ such that $q > \beta\sqrt{nk} \cdot \omega(\log(nk))$, and $\ell, \log q \leq \mathsf{poly}(nk)$. The $\mathsf{MSIS}_{q,\ell,k,\beta}$ problem is as hard as the worst-case lattice Generalized-Independent-Vector-Problem (GIVP) in dimension $N = nk$ with approximation factor $\beta\sqrt{N} \cdot \omega(\sqrt{\log N})$.*

The following will be a useful shorthand to be used in our security proof.

Definition 3.19. *For any $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$, positive integers rep and ν, let $\mathcal{D}^{\mathsf{bd\text{-}MLWE}}_{q,\ell,k,\sigma,\mathsf{rep},\nu}(\mathbf{A})$ be the distribution defined as $\left\{ \lfloor \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \rceil_\nu \mid (\mathbf{s}, \mathbf{e}) = (\sum_{i \in [\mathsf{rep}]} \mathbf{s}_i, \sum_{i \in [\mathsf{rep}]} \mathbf{e}_i), \forall i \in [\mathsf{rep}], (\mathbf{s}_i, \mathbf{e}_i) \xleftarrow{\$} \mathcal{D}_\sigma^\ell \times \mathcal{D}_\sigma^k \right\}$. That is, it samples rep $\mathsf{MLWE}_{q,\ell,k,\sigma}$ instances, aggregates them, and drops ν trailing bits.*

The following is an immediate application of the regularity lemma [LPR13]. While [PKM+24] provides a formal case for $\mathsf{rep} = 1$, it generalizes easily to any rep using Lemma 3.3.

Lemma 3.20. *For any* $\sigma > \sqrt{\frac{\log(2n\cdot\max\{\ell,k\})+\lambda}{\pi}}$ *and* $\sqrt{\mathsf{rep}}\cdot\sigma > 2n\cdot q^{\frac{1}{k+\ell}+\frac{2}{n\ell}}$ *and* $\nu < \log(q)-2$*, the following holds with all but probability* $2^{-\lambda}$*:*

$$\Pr_{\mathbf{A}\xleftarrow{\$}\mathcal{R}_q^{k\times\ell}}\left[H_\infty\left(\mathcal{D}^{\mathsf{bd\text{-}MLWE}}_{q,\ell,k,\sigma,\mathsf{rep},\nu}(\mathbf{A})\right) \geq n-1\right] \geq 1-2^{-n+1}.$$

3.9 Forking Lemma with Oracle Access

The forking lemma was originally introduced by Pointcheval and Stern [PS00] in the context of signature schemes. The lemma was later reformulated by Bellare and Neven [BN06] which extracts the purely probabilistic nature of the forking lemma. Below, we define a variant of their forking lemma defined by El Kaafarani and Katsuamta [EK18], allowing the forking algorithm to have access to a deterministic oracle. This type of formalization is useful when we are trying to reduce from an interactive assumption to the security of a scheme.

Lemma 3.21 (Forking Lemma with Oracle Access). *Fix an integer* $q_{\mathsf{Fork}} \geq 1$ *and a set* $\mathcal{H}$ *of size* $h \geq 2$*. Let* $\mathcal{A}$ *be a randomized algorithm that has oracle access to a deterministic algorithm* $\mathcal{O}$*, where on input* $\mathsf{par}, \vec{h} := (h_1, \cdots, h_{q_{\mathsf{Fork}}})$*, algorithm* $\mathcal{A}$ *returns* $J \in [0, \cdots, q_{\mathsf{Fork}}]$ *and an arbitrary string* σ*. Let* IG *be a randomized algorithm called the* input generator*. The accepting probability of* $\mathcal{A}$*, denoted* acc*, is defined below:*

$$\mathsf{acc} = \Pr\left[(\mathsf{par}, \overline{\mathsf{par}}) \xleftarrow{\$} \mathsf{IG},\ \vec{h} \xleftarrow{\$} \mathcal{H}^{q_{\mathsf{Fork}}},\ (J,\sigma) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}(\mathsf{par}, \vec{h}) : J \geq 1\right].$$

The forking algorithm $\mathsf{Fork}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}_{\mathcal{A}}$ *associated to* $\mathcal{A}$ *is a randomized oracle-calling algorithm that takes input* par *and proceeds as in Fig. 5. Let*

$$\mathsf{frk} = \Pr\left[(\mathsf{par}, \overline{\mathsf{par}}) \xleftarrow{\$} \mathsf{IG};\ (b, (\sigma_1, \sigma_2)) \xleftarrow{\$} \mathsf{Fork}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}_{\mathcal{A}}(\mathsf{par}) \ :\ b = 1\right].$$

Then,

$$\mathsf{frk} \geq \mathsf{acc}\cdot\left(\frac{\mathsf{acc}}{q} - \frac{1}{h}\right).$$

4 Algebraic One-More Module Learning with Errors

We introduce the "Algebraic One-More Module Learning with Errors" (AOM-MLWE) problem, a term coined in reference to the algebraic one-more discrete logarithm (AOM-DL) problem recently introduced by Nick, Ruffing, and Seurin [NRS21]. This problem represents a notably milder variant when compared to the conventional "non-algebraic" one-more challenges encountered in classical contexts, such as the One-More Discrete Logarithm (OM-DL) problem [BNPS02, BNPS03, BMV08, BFP21]. After presenting AOM-MLWE and discussing the subtleties of the interplay of parameters, we delve into its relationship with the well-established MLWE and MSIS problems.

4.1 Motivation

Before explaining the algebraic variant, let us recall what a *non-algebraic* one-more type problem looks like. Informally, a non-algebraic one-more type assumption provides an adversary with Q challenge instances. It is further given access to an oracle that solves any instance given by the adversary, where these instances do

Algorithm $\mathsf{Fork}_{\mathcal{A}}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}(\mathsf{par})$

$\mathsf{coin} \xleftarrow{\$} \{0,1\}^{\ell_{\mathcal{A}}}$ // $\ell_{\mathcal{A}}$-bit randomness used by $\mathcal{A}$
$\vec{h} := (h_1, \cdots, h_q) \xleftarrow{\$} \mathcal{H}^{q_{\mathsf{Fork}}}$
$(I, \sigma) := \mathcal{A}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}(\mathsf{par}, \vec{h}; \rho)$
if $[\![I = 0]\!]$ **then**
  return $(0, (\bot, \bot))$
$(h'_I, \cdots, h'_q) \xleftarrow{\$} \mathcal{H}^{q_{\mathsf{Fork}}-I+1}$
$\vec{h}' := (h_1, \cdots, h_{I-1}, h'_I, \cdots, h'_q)$
$(I', \sigma') := \mathcal{A}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}(\mathsf{par}, \vec{h}'; \rho)$
if $[\![I = I']\!] \wedge [\![h_I \neq h'_I]\!]$ **then**
  return $(1, (\sigma, \sigma'))$
else
  return $(0, (\bot, \bot))$

Figure 5: Description of the oracle-calling forking algorithm $\mathsf{Fork}_{\mathcal{A}}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}$.

not necessarily have to be the provided challenge instances. The problem then asks the adversary to output the Q solutions to the Q challenge instances while only given at most $Q-1$ access to the oracle.

For instance, in the context of the OM-DL problem, the adversary is presented with $(g^{a_i})_{i\in[Q]}$ as the set of challenge instances. The oracle is designed to solve $\mathsf{dlog}_g(h)$ for any queried group element h. It is noteworthy that while OM-DL qualifies as a strong "non-falsifiable" assumption [Nao03, GW11], primarily because the challenger is inefficient, OM-DL has proven successful and remains unsolved in practice. Indeed, we further gain confidence in the hardness of OM-DL as it was recently shown by Bauer, Fuchsbauer, and Plouviez [BFP21] to be hard in the generic group model [Sho97, Mau05].

Translating the OM-DL problem into the lattice-based framework presents a non-trivial challenge, primarily due to the fact that MLWE instances exhibit more inherent structure than their discrete logarithm (DL) counterparts. In the MLWE setting, let's assume that the adversary is given a set of challenges $(\mathbf{t}_i)_{i\in[Q]} = (\mathbf{A}\mathbf{s}_i + \mathbf{e}_i)_{i\in[Q]}$, and it queries the oracle for $\mathbf{t}_1 + \mathbf{t}_i$ for $i \in [2 : Q]$. In response, the oracle provides the corresponding MLWE solutions, namely $(\mathbf{s}_1 + \mathbf{s}_i, \mathbf{e}_1 + \mathbf{e}_i)$.

While this exchange doesn't immediately disclose the individual values of $\mathbf{s}_i$ and $\mathbf{e}_i$, with a sufficient number of samples, the adversary can statistically infer all these values — see Section 4.3 for more discussion. Such statistical attacks do not exist in the DL setting as each secret exponent a_i are distributed uniformly over $\mathbb{Z}_p$; this is in sharp contrast with lattices where the secrets are small. This is only one trivial attack against OM-MLWE and it is unclear whether other, more sophisticated attacks exist. In fact, the adversary may learn non-trivial information via non-algebraic attacks; for instance, it can perform bit decomposition on the $\mathbf{t}_i$'s and use it in a non-trivial manner to break OM-MLWE. This stands in contrast to the classical setting, where we have the generic group model (GGM) or the algebraic group model (AGM) [FKL18], both of which support the belief that such non-algebraic adversaries are not more useful.

In summary, OM-MLWE has two deficiencies. One is that, similarly to OM-DL, it is a non-falsifiable assumption since the challenger is inefficient. The other more significant one is that, while we can define OM-MLWE, it is unclear how to can gain confidence on its hardness as there is a plethora of plausible attacks against it. This is in sharp contrast to the classical setting where we have the GGM or AGM that lets us solely focus on the restricted and easy-to-analyze class of algebraic adversaries; since lattices naturally allow non-algebraic operations, it would be too restrictive to assume only an algebraic adversary.

4.2 Definition of AOM-MLWE

This brings us to the *algebraic* OM-MLWE problem, resolving both deficiencies of OM-MLWE. The term "algebraic" is employed because when the adversary queries the MLWE solving oracle with a vector $\mathbf{b}$, it must also provide a vector $\mathbf{d}$ that essentially "explains" the vector $\mathbf{b}$, as a proof that the query was made on a linear combination of the challenges. More formally, this requirement is expressed as $\mathbf{b} = \mathbf{Td}$, where $\mathbf{T} = [\mathbf{As}_1 + \mathbf{e}_1 \mid \cdots \mid \mathbf{As}_Q + \mathbf{e}_Q]$, is a matrix representing the set of Q MLWE challenges generated by the challenger.

In particular, the AOM-MLWE problem restricts the adversary to only query the MLWE oracle on a linear combination of the MLWE challenges. Notice that since the challenger knows the corresponding MLWE secrets, it can answer the adversary's queries efficiently, thus making the AOM-MLWE assumption falsifiable.

The algebraic restriction in the Algebraic One-More Module Learning with Errors (AOM-MLWE) problem offers additional advantages. Firstly, it simplifies the cryptanalysis process in comparison to the non-algebraic case, primarily because of the stringent limitations placed on the vector $\mathbf{b}$ that the adversary can query to the MLWE solving oracle. In our subsequent analysis, we demonstrate that the "selective" variant of the AOM-MLWE problem is as hard as solving the standard MLWE and MSIS problems. Importantly, we believe that no such analogous reduction exists in the non-algebraic setting, even when considering the selective scenario. This reduction not only enhances the credibility of the AOM-MLWE problem's hardness but also underscores that the only conceivable approach to weaken its security would be to exploit the adaptiveness.

4.2.1 Definition of AOM-MLWE

Formally, the AOM-MLWE problem is defined as follows, supposing we are working over the ring of integer $\mathcal{R}_q$ of a number field.

Definition 4.1 (AOM-MLWE). *Let ℓ, k, q, Q be integers and $(\mathcal{D}_i)_{i\in[Q]}$ be a set of probability distributions over $\mathcal{R}_q$ with $k \geq \ell$. Let $\mathcal{L}$ denote an efficiently checkable subset of $\mathcal{R}_q^{Q\times(Q-1)}$ and $B_{\mathcal{L}}$, $B_{\mathbf{s}}$, $B_{\mathbf{e}}$ be integers. The advantage of an adversary $\mathcal{A}$ against the (search)* Algebraic One-More Module Learning with Errors $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\ \mathcal{L},B_{\mathcal{L}},B_{\mathbf{s}},B_{\mathbf{e}}}$ *problem is defined as:*

$$\mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda) = \Pr\left[\mathsf{Game}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda, 1^Q) = 1\right],$$

where $\mathsf{Game}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}$ *is shown in Fig. 6. The* $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{s}},B_{\mathbf{e}}}$ *assumption states that any efficient adversary $\mathcal{A}$ has some negligible advantage. We also define a* selective *variant of* AOM-MLWE*, denoted as* sel-AOM-MLWE*, whose game is shown in Fig. 6.*

Previously, we allowed each MLWE sample to come from a different distribution. As we later see, for threshold signatures, we set the first MLWE sample to have a smaller noise compared to the other MLWE samples. Moreover, we weaken the winning condition of the adversary so that it only needs to solve an *approximate* MLWE problem. We insist that the adversary wins even if it recovers a solution to the MLWE problem where each MLWE challenge ($\mathbf{t}_i = \mathbf{As}_i + \mathbf{e}_i$) is modified to be $v_i \cdot \mathbf{t}_i$ for a small non-zero $v_i \in \mathcal{R}_q$. This relaxation captures a recurrent issue in lattice-based identification protocols, Fiat-Shamir based signatures, and zero-knowledge proof systems (see for instance [BCK+14, EK18, BLS19, ENS20] for some discussions). Moreover, we define the assumption to be Q-bounded, that is, any adversary is limited to making at most $Q-1$ queries to the MLWE solving oracle. If needed we can define a (polynomially) unbounded definition where the game is not quantified by Q.

4.2.2 Hermite Normal Form vs Uniform Secrets

Similarly to the standard definition of MLWE, we can define a variant of AOM-MLWE, denoted as AOM-UMLWE, where the secret $(\mathbf{s}_i)_{i\in[Q]}$ are sampled *uniformly* from $\mathcal{R}_q^{\ell\times Q}$ instead of from the same distribution as the noise. In particular, the challenger no longer checks the bound on $\widehat{\mathbf{S}}$ output by the adversary. It is easy to

$\mathsf{Game}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda, 1^Q)$

$(\mathsf{ctr}, \mathbf{D}) := (1, \perp)$ // $\mathbf{D}$ is an "empty" matrix
$\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$
for $i \in [Q]$ **do**
  $(\mathbf{s}_i, \mathbf{e}_i) \xleftarrow{\$} \mathcal{D}_i^\ell \times \mathcal{D}_i^k$
  $(\mathbf{S}, \mathbf{E}) := ([\mathbf{s}_1 \mid \cdots \mid \mathbf{s}_Q], [\mathbf{e}_1 \mid \cdots \mid \mathbf{e}_Q])$
  $\mathbf{T} := \mathbf{AS} + \mathbf{E} \in \mathcal{R}_q^{k\times Q}$
$(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}}) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{solve}}}(\mathbf{A}, \mathbf{T})$
// Check format of output, where $\mathcal{L} \subseteq \mathcal{R}_q^{Q\times(Q-1)}$
if $[\![(\mathbf{D}, \mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}}) \in \mathcal{L} \times \mathcal{R}_q^Q \times \mathcal{R}_q^{\ell\times Q} \times \mathcal{R}_q^{k\times Q}]\!]$
  // Check size of solution: v_i is the i-th entry of $\mathbf{v}$
  if $[\![\forall i \in [Q], 0 < \|v_i\|_2 \le B_{\mathcal{L}}$
      $\wedge \|\widehat{\mathbf{S}}\|_2 \le B_{\mathbf{s}} \wedge \|\widehat{\mathbf{E}}\|_2 \le B_{\mathbf{e}}]\!]$
    // Check if it is an *approximate* MLWE solution
    if $[\![\mathbf{v}^\top \odot \mathbf{T} = \mathbf{A}\widehat{\mathbf{S}} + \widehat{\mathbf{E}}]\!]$
      return 1
return 0

$\mathcal{O}_{\mathsf{solve}}(\mathbf{d})$

if $[\![\mathbf{d} \notin \mathcal{R}_q^Q]\!] \vee [\![\mathsf{ctr} \ge Q]\!]$
  return 0
$\mathbf{D} \leftarrow [\mathbf{D} \mid \mathbf{d}]$ // Update matrix $\mathbf{D} \in \mathcal{R}_q^{Q\times\mathsf{ctr}}$
$\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
return $(\mathbf{Sd}, \mathbf{Ed}) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^k$

$\mathsf{Game}^{\mathsf{sel\text{-}AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda, 1^Q)$

$\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$
for $i \in [Q]$ **do**
  $(\mathbf{s}_i, \mathbf{e}_i) \xleftarrow{\$} \mathcal{D}_i^\ell \times \mathcal{D}_i^k$
  $(\mathbf{S}, \mathbf{E}) := ([\mathbf{s}_1 \mid \cdots \mid \mathbf{s}_Q], [\mathbf{e}_1 \mid \cdots \mid \mathbf{e}_Q])$
  $\mathbf{T} := \mathbf{AS} + \mathbf{E} \in \mathcal{R}_q^{k\times Q}$
$\mathbf{D} \xleftarrow{\$} \mathcal{A}(\mathbf{A})$
if $[\![\mathbf{D} \notin \mathcal{L} \subseteq \mathcal{R}_q^{Q\times(Q-1)}]\!]$
  return 0
$(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}}) \xleftarrow{\$} \mathcal{A}(\mathbf{A}, \mathbf{T}, (\mathbf{SD}, \mathbf{ED}))$
// Remaining check is identical to $\mathsf{Game}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}$

Figure 6: The adaptive and selective algebraic one-more MLWE problem. In the selective setting, the adversary $\mathcal{A}$ commits to all the coefficients before observing the MLWE samples. Recall $\odot$ denotes the column-wise multiplication.

see that AOM-MLWE implies AOM-UMLWE. Below, we show the opposite indication. The proof is a slight modification of the standard reduction by Applebaum et al. [ACPS09] from (the non-structured) ULWE to LWE. The non-triviality comes from the way the reduction simulates the oracle $\mathcal{O}_{\mathsf{solve}}$ and how it transforms the solution. Looking ahead, handling uniform secrets will be more convenient when we later establish the hardness of the *selective* AOM-UMLWE (sel-AOM-UMLWE) based on the hardness of MLWE and MSIS.

Lemma 4.2 (AOM-UMLWE implies AOM-MLWE). *Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ split into $s \in [n]$ fields. If there exists an adversary $\mathcal{A}$ against the $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{s}},B_{\mathbf{e}}}$ problem, then we can construct an adversary $\mathcal{B}$ against the $\mathsf{AOM\text{-}UMLWE}_{q,\ell,k+\ell,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},\max\{B_{\mathbf{s}},B_{\mathbf{e}}\}}$ problem such that*

$$\left(1 - \frac{1}{q^{n\cdot(k-\ell+1)/s}}\right)^{s\ell} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda) \le \mathsf{Adv}^{\mathsf{AOM\text{-}UMLWE}}_{\mathcal{B}}(1^\lambda),$$

where $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$.

Proof. Let $\mathcal{B}$ be an adversary against AOM-UMLWE given $(\mathbf{A}, \mathbf{T})$ as input, where $(\mathbf{A}_0, \mathbf{A}_1) \xleftarrow{\$} \mathsf{GL}_\ell(\mathcal{R}_q) \times \mathcal{R}_q^{k\times\ell}$, $\mathbf{A}^\top := [\mathbf{A}_0^\top \mid \mathbf{A}_1^\top]$, $\mathbf{S} \xleftarrow{\$} \mathcal{R}_q^{\ell\times Q}$, $\mathbf{E} \xleftarrow{\$} \mathcal{D}_1^{k+\ell} \times \cdots \times \mathcal{D}_Q^{k+\ell}$, and $\mathbf{T} = \mathbf{AS} + \mathbf{E} \in \mathcal{R}_q^{(k+\ell)\times Q}$. Using [BJRW23, Appendix A], $\mathbf{A}$ contains ℓ row vectors forming an invertible matrix with probability at least $(1 - \frac{1}{q^{n\cdot(k-\ell+1)/s}})^{s\ell}$. $\mathcal{B}$ gives up the reduction and outputs 0 if such row vectors do not exist. Otherwise, let $\mathbf{A}_0 \in \mathcal{R}_q^{\ell\times\ell}$ and $\mathbf{A}_1 \in \mathcal{R}_q^{k\times\ell}$ denote the first ℓ and last k rows of $\mathbf{A}$ and assume without loss of generality that $\mathbf{A}_0$ is invertible over $\mathcal{R}_q$. Define $(\mathbf{E}_0, \mathbf{T}_0, \mathbf{E}_1, \mathbf{T}_1) \in (\mathcal{R}_q^{\ell\times Q})^2 \times (\mathcal{R}_q^{k\times Q})^2$ similarly.

We now describe how $\mathcal{B}$ internally uses an adversary $\mathcal{A}$ against AOM-MLWE. $\mathcal{B}$ first computes $\mathbf{A}^* = -\mathbf{A}_1\mathbf{A}_0^{-1}$ and $\mathbf{T}^* = \mathbf{T}_1 - \mathbf{A}_1\mathbf{A}_0^{-1}\mathbf{T}_0$ and invokes $\mathcal{A}$ on input $(\mathbf{A}^*, \mathbf{T}^*) \in \mathcal{R}_q^{k\times\ell} \times \mathcal{R}_q^{k\times Q}$. When $\mathcal{A}$ queries its oracle $\mathcal{O}_{\mathsf{solve}}$ for the i-th time ($i \le Q-1$) on input $\mathbf{d}_i \in \mathcal{L}$, $\mathcal{B}$ queries its own solve oracle on the same input and receives back $(\mathbf{u}_i, \mathbf{w}_i) = (\mathbf{Sd}_i, \mathbf{Ed}_i) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^{k+\ell}$. $\mathcal{B}$ then discards $\mathbf{u}_i$, parses $\mathbf{w}_i$ into $(\mathbf{u}_i^*, \mathbf{w}_i^*) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^k$

such that $\mathbf{u}_i^*$ and $\mathbf{w}_i^*$ are the first ℓ and last k entries of $\mathbf{w}_i$. It then returns $(\mathbf{u}_i^*, \mathbf{w}_i^*)$ to $\mathcal{A}$. Finally, when $\mathcal{A}$ outputs a solution $(\mathbf{v}^*, \widehat{\mathbf{S}}^*, \widehat{\mathbf{E}}^*) \in \mathcal{R}_q^Q \times \mathcal{R}_q^{\ell \times Q} \times \mathcal{R}_q^{k \times Q}$, $\mathcal{B}$ outputs the following as its solution:

$$(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}}) = \left(\mathbf{v}^*, -\mathbf{A}_0^{-1}\widehat{\mathbf{S}}^* + \mathbf{v}^{*\top} \odot \mathbf{A}_0^{-1}\mathbf{T}_0, \begin{bmatrix} \widehat{\mathbf{S}}^* \\ \widehat{\mathbf{E}}^* \end{bmatrix} \right) \in \mathcal{R}_q^Q \times \mathcal{R}_q^{\ell \times Q} \times \mathcal{R}_q^{(k+\ell) \times Q}.$$

It remains to analyze the winning probability of $\mathcal{B}$. It is clear that $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$. Given $\mathbf{A}_0\mathbf{S} + \mathbf{E}_0 = \mathbf{T}_0$ and the fact that $\mathbf{A}_0$ is invertible, we have $\mathbf{S} = -\mathbf{A}_0^{-1}(\mathbf{E}_0 - \mathbf{T}_0)$. Plugging this into $\mathbf{A}_1\mathbf{S} + \mathbf{E}_1 = \mathbf{T}_1$, we have $(-\mathbf{A}_1\mathbf{A}_0^{-1})\mathbf{E}_0 + \mathbf{E}_1 = \mathbf{T}_1 - \mathbf{A}_1\mathbf{A}_0^{-1}\mathbf{T}_0$, which is $\mathbf{A}^*\mathbf{E}_0 + \mathbf{E}_1 = \mathbf{T}^*$. Since $\mathbf{A}_1$ and $\mathbf{T}_1$ are distributed randomly, $\mathcal{B}$ perfectly simulates the AOM-MLWE problem instance to $\mathcal{A}$. Moreover, $\mathcal{B}$ perfectly simulates the solve oracle queries since $\mathbf{E}\mathbf{d}_i \in \mathcal{R}_q^{k+\ell}$ allows to compute $(\mathbf{E}_0\mathbf{d}_i, \mathbf{E}_1\mathbf{d}_i) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^k$. Finally, if $\mathcal{A}$ outputs a valid solution $(\mathbf{v}^*, \widehat{\mathbf{S}}^*, \widehat{\mathbf{E}}^*)$, then we have $\mathbf{v}^{*\top} \odot \mathbf{T}^* = \mathbf{A}^*\widehat{\mathbf{S}}^* + \widehat{\mathbf{E}}^*$. Substituting $\mathbf{T}$ and $\mathbf{v}$, we can rewrite the left hand side as

$$\mathbf{v}^{*\top} \odot \mathbf{T}^* = \mathbf{v}^\top \odot \mathbf{T}_1 - \mathbf{v}^\top \odot \mathbf{A}_1\mathbf{A}_0^{-1}\mathbf{T}_0.$$

Moreover, substituting $\mathbf{A}$, $\widehat{\mathbf{S}}$, and $\mathbf{v}$, we can rewrite the right hand side as

$$\mathbf{A}^*\widehat{\mathbf{S}}^* + \widehat{\mathbf{E}}^* = (-\mathbf{A}_1\mathbf{A}_0^{-1})\widehat{\mathbf{S}}^* + \widehat{\mathbf{E}}^* = \mathbf{A}_1(\widehat{\mathbf{S}} - \mathbf{v}^\top \odot \mathbf{A}_0^{-1}\mathbf{T}_0) + \widehat{\mathbf{E}}^*.$$

Since both sides are identical, we have

$$\mathbf{A}_1\widehat{\mathbf{S}} + \widehat{\mathbf{E}}^* = \mathbf{v}^\top \odot \mathbf{T}_1 - \mathbf{v}^\top \odot \mathbf{A}_1\mathbf{A}_0^{-1}\mathbf{T}_0 + \mathbf{A}_1(\mathbf{v}^\top \odot \mathbf{A}_0^{-1}\mathbf{T}_0) = \mathbf{v}^\top \odot \mathbf{T}_1,$$

where for the second equality, we used the fact that for any $\mathbf{u} \in \mathcal{R}_q^c$ and $(\mathbf{B}, \mathbf{C}) \in \mathcal{R}_q^{a \times b} \times \mathcal{R}_q^{b \times c}$, we have $\mathbf{u}^\top \odot (\mathbf{B}\mathbf{C}) = \mathbf{B}(\mathbf{u}^\top \odot \mathbf{C})$. Finally, from $\widehat{\mathbf{S}} = -\mathbf{A}_0^{-1}\widehat{\mathbf{S}}^* + \mathbf{v}^{*\top} \odot \mathbf{A}_0^{-1}\mathbf{T}_0$, we have $\mathbf{A}_0\widehat{\mathbf{S}} + \widehat{\mathbf{S}}^* = \mathbf{A}_0(\mathbf{v}^\top \odot \mathbf{A}_0^{-1}\mathbf{T}_0)$, where the right hand side is equal to $\mathbf{v}^\top \odot \mathbf{T}_0$ again due to commutativity. Combining all the arguments together, we have $\mathbf{A}\widehat{\mathbf{S}} + \widehat{\mathbf{E}} = \mathbf{v}^\top \odot \mathbf{T}$ as desired. Lastly, noting the size of $(\widehat{\mathbf{S}}^*, \widehat{\mathbf{E}}^*)$ translates to $\widehat{\mathbf{E}}$, the output of $\mathcal{B}$ is a valid solution when $\mathcal{A}$ outputs a valid solution. This completes the proof. □

4.2.3 AOM-UMLWE with Invertible Submatrix

Looking ahead, in Section 4.4, we establish the hardness of the (selective) AOM-UMLWE problem based on the hardness of MSIS and MLWE. For this, we need to restrict the challenge matrix $\mathbf{A}$ of AOM-UMLWE to contain ℓ rows that form an invertible matrix over $\mathcal{R}_q$ — we say that $\mathbf{A}$ *contains an invertible submatrix*. This restriction is explicitly used when reducing MSIS to selective AOM-UMLWE (see Footnote 11) and can be easily enforced by resampling $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}$ until an invertible matrix is found, an efficiently computable check using standard linear algebra. For certain choices of $(k, \ell, \mathcal{R}_q)$, this is without loss of generality as $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}$ satisfies this condition with overwhelming probability. In other cases like $k = \ell$ and $\mathcal{R}_q$ a fully-splitting ring, $\mathbf{A}$ is non-invertible with a non-negligible probability (see [BJRW23, Appendix A] for details). Throughout the paper, we may say AOM-UMLWE *with an invertible submatrix* to denote AOM-UMLWE with this restriction on $\mathbf{A}$.

It is worth highlighting that this restriction on $\mathbf{A}$ is *not* carried over to our threshold signature scheme, where $\mathbf{A}$ serves as the public matrix included in the verification key. In other words, the key generation algorithm does not require $\mathbf{A}$ containing an invertible submatrix. This is because our threshold signature relies on the hardness of the AOM-MLWE problem, as opposed to the AOM-UMLWE problem. Taking a closer look at Lemma 4.2, it can be checked that AOM-UMLWE *with an invertible matrix* also implies AOM-MLWE without any restriction on $\mathbf{A}$. Combining this with the implication that MSIS and MLWE implies (selective) AOM-UMLWE with an invertible matrix, as will be shown in Section 4.4, we establish hardness of (selective) AOM-MLWE based on standard lattice assumptions without any restriction on the challenge matrix $\mathbf{A}$. As an immediate corollary of the proof of Lemma 4.2, we state the following for completeness.

Corollary 4.3 (AOM-UMLWE with an invertible submatrix implies AOM-MLWE). *Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ split into $s \in [n]$ fields. If there exists an adversary $\mathcal{A}$ against the $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{s}},B_{\mathbf{e}}}$ problem, then we can construct an adversary $\mathcal{B}$ against the $\mathsf{AOM\text{-}UMLWE}_{q,\ell,k+\ell,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},\max\{B_{\mathbf{s}},B_{\mathbf{e}}\}}$ problem with an invertible submatrix such that*

$$\mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda) \leq \mathsf{Adv}^{\mathsf{AOM\text{-}UMLWE}}_{\mathcal{B}}(1^\lambda),$$

where $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$.

4.3 Preliminary Discussion on the Hardness of AOM-MLWE

Here, we provide an informal discussion on the hardness of (adaptive) AOM-MLWE under standard assumptions, this section provides insight into why we believe it is hard. We later use these insights to establish the hardness of the *selective* AOM-MLWE problem for well-chosen parameters based on MSIS and MLWE. This informal discussion will also form the basis for the cryptanalysis of the AOM-MLWE problem in Section 7.

4.3.1 When AOM-MLWE is Trivially Broken.

The AOM-MLWE problem is parameterized by many parameters, of which the space of accepted linear combinations $\mathcal{L} \subseteq \mathcal{R}_q^{Q\times(Q-1)}$ and the distributions $(\mathcal{D}_i)_{i\in[Q]}$ play one of the most fundamental roles. All other parameters such as the size of the MLWE solutions (i.e., $B_{\mathbf{s}}, B_{\mathbf{e}}$) and the accepted "slack" $B_{\mathcal{L}}$ are more standard and can be handled similarly following prior works on lattice-based cryptography. Throughout this section, for simplicity of explanation, assume $\mathcal{L} = \prod_{i\in[Q-1]} \hat{\mathcal{L}}$ which naturally embeds into $\oplus_{i\in[Q]}\mathcal{R}_q = \mathcal{R}_q^Q$ and allows to consider each column of $\mathcal{L}$ as being included in $\hat{\mathcal{L}}$. Specifically, the vector $\mathbf{d}$ the adversary $\mathcal{A}$ queries to the MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$ satisfies $\mathbf{d} \in \hat{\mathcal{L}}$.

First Insecure Example: Overstretched Queries and Separation of Secrets. In the scenario where $\mathcal{D}_i = \mathcal{D}$ for all $i \in [Q]$ and $\mathcal{D}$ represents a distribution that generates polynomials with coefficients having an ℓ_∞-norm smaller than $0 < B < \sqrt{q/2}$ and with the additional assumptions that $\mathbf{d} = [1 \mid B \mid \mathbf{0}]^\top \in \mathcal{R}_q^Q$ is within $\hat{\mathcal{L}}$, a non-trivial vulnerability emerges.

The combination of the secrets by $\mathbf{d}$ gives the equation $\mathbf{Sd} = \mathbf{s}_1 + B \cdot \mathbf{s}_2 \pmod q$. However, by the assumption on the size of $\mathbf{s}_i$ and B, the magnitude of $\mathbf{Sd}$ is much smaller than q, meaning that this equation actually holds without modular reduction. Then the adversary can easily recover $(\mathbf{s}_1, \mathbf{s}_2)$: we simply have $\mathbf{s}_1 = \mathbf{Sd} \mod B$ precisely corresponding to the original $\mathbf{s}_1$, as the entries of $\mathbf{s}_1$ are all smaller than B. The adjacent figure proposes a two-dimensional representation of the attack, where the reduction mod B are depicted by their decoding cells on the bottom. Significantly, in this situation, the adversary can extract two MLWE secrets using just one oracle query to the MLWE solver. This outcome effectively breaches the security of AOM-MLWE and underscores the critical requirement that the size of the MLWE secret and noise must be quite larger than the elements present in $\hat{\mathcal{L}}$.

Second Insecure Example: Anomalously Large Number of Queries and Statistical Recovery. Now assume $\hat{\mathcal{L}} = B_{\infty,1}(0)$ the ℓ_∞ ball of radius 1 and assert that the secrets are independently and uniformly distributed with coefficients bounded by $1 \ll B < q/2$.

While the above attack no longer works as the query vectors $\mathbf{d}$ are too small compared to the size of the secrets and noises, consider an adversary that queries the oracle on input $\mathbf{d}_i \in \mathcal{R}_q^Q$ for $i \in [2:Q]$, where $\mathbf{d}_i = (1, 0, \ldots, 0, 1, 0, \ldots, 0)^T$ is the vector with zero entries except for the 1-st and i-th entries which are set to 1. Hence, the adversary receives tuples of the form $(\mathbf{s}_i^* = \mathbf{s}_1 + \mathbf{s}_i)_{i\in[2:Q]} \pmod q$ and this result is also valid without modular reduction by assumption on B. The attacker can then construct the empirical estimator

$$\tilde{\mathbf{s}}_1 = \frac{1}{Q-1}\sum_{i=2}^{Q} \mathbf{s}_i^*$$

which converges towards its mean value $\mathbf{s}_1$. Quantitatively, this sum will be a random variable centered at $\mathbf{s}_1$ and of standard deviation of order $B/\sqrt{3Q}$. The attacker can then round $\tilde{\mathbf{s}}_1$ to the nearest integer and claim it as the secret $\mathbf{s}_1$. By the Tchebyshev inequality and amplification, this process is correct with probability of at least $(1 - \frac{2B^2}{3Q})^n$ for n being the number of coefficients of the secrets. Hence when $Q \geq nB^2$, i.e., the size of the secrets is much smaller than the square root number of oracle queries, a total recovery is possible in linear time. The probability density of the candidate estimator is superposed with the acceptance zone which is the segment of length 1 centered at the secret. This example indicates that the size of the MLWE secret and noise crucially depend on the accepted number of queries Q.

4.3.2 When AOM-MLWE is Plausibly Hard.

It is important to highlight that the attacks discussed in the previous examples are, in a sense, statistical in nature. These attacks rely solely on the linear combination of the secrets or the vectors but do not utilize any information about the specific MLWE sample $\mathbf{T} = \mathbf{AS}+\mathbf{E}$. Indeed, when we delve into the assessment of the hardness of the "selective" AOM-MLWE problem, we observe that these statistical attacks represent the sole advantage an adversary possesses compared to MLWE and MSIS. To put it differently, once we configure the parameters in such a way that the aforementioned statistical attacks are no longer viable, the only viable approach to compromise sel-AOM-MLWE is to break MLWE or MSIS. This insight serves as a cornerstone when evaluating the concrete hardness of AOM-MLWE using state-of-the-art cryptanalysis techniques.

In this work, we provide two reductions from MLWE or MSIS to sel-AOM-MLWE. Conceptually, the two reductions embed, in a different way, a single MLWE instance $\mathbf{t}^* \in \mathcal{R}_q^k$ into the Q MLWE instances $\mathbf{T} \in \mathcal{R}_q^{k\times Q}$ provided by the sel-AOM-MLWE game. It is worth noting that in the non-algebraic setting, it is unclear whether even the selective variant is implied by any standard assumptions. We thus believe that there is a fundamental gap between the hardness of the AOM-MLWE problem and its non-algebraic variant and view this indication as an evidence for the hardness of AOM-MLWE.

In the following, we present only one of the reductions, capturing the parameter setting of the threshold signature we construct in Section 5. This will be our main focus of cryptanalysis. The second reduction, which we call the "alternative" reduction, captures a parameter setting not used in this work and is presented in Appendix B.

4.4 MSIS and MLWE Imply Selective AOM-MLWE

In this section, we embed $\mathbf{t}^* = \mathbf{As}^* + \mathbf{e}^*$ in one of the columns of $\mathbf{T} = \mathbf{AS} + \mathbf{E}$ and define the accepted linear combinations $\mathcal{L}$ so that $\mathbf{t}^*$ remains a hard MLWE instance even after the adversary obtains the hints $(\mathbf{SD}, \mathbf{ED})$. Without loss of generality, we set the first column $(\mathbf{S}, \mathbf{E})$ to be $(\mathbf{s}^*, \mathbf{e}^*)$. Moreover, for simplicity, we focus on the uniform secret sel-AOM-UMLWE, establishing the hardness of sel-AOM-MLWE. Note that, in this section, we only consider the slightly restricted AOM-UMLWE problem whose matrix $\mathbf{A}$ contains an invertible submatrix over $\mathcal{R}_q$ (cf. Section 4.2.3). For readability, we only make this restriction explicit when it is required.

4.4.1 Constraints and Parameter Selection.

As discussed, the parameters for which (selective) AOM-UMLWE is hard need to be chosen meticulously. We provide the set of parameters for which we establish hardness of $\mathsf{sel\text{-}AOM\text{-}UMLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{e}}}$. Below, on first glance, the condition on the accepted linear combinations $\mathcal{L} \subseteq \mathcal{R}_q^{Q\times(Q-1)}$ may seem contrived and it is not immediately clear how one sets $\mathcal{L}$ in practice. A concrete example of $\mathcal{L}$ satisfying such constraints is provided in Section 4.5. Looking ahead, this is exactly the same $\mathcal{L}$ appearing in the proof of our threshold signature scheme.

Constraints on Parameters. We first define the following intermediate variables that will be used in the proof:

- $\mathcal{D}_1$ is defined as $2 \cdot \mathcal{D}_{\sigma_1} := \{2 \cdot x \mid x \xleftarrow{\$} \mathcal{D}_{\sigma_1}\}$, where $\mathcal{D}_{\sigma_1}$ is a discrete Gaussian distribution $\mathcal{D}_{\sigma_1}$ with width $\sigma_1 > 0$.
- $\mathcal{D}_i$ for $i \in [2 : Q]$ is a discrete Gaussian distribution $\mathcal{D}_{\sigma_i}$ with width $\sigma_i > 0$, where we denote $\sigma^* = \min_{i \in [2:Q]} \sigma_i$.
- Accepted linear combinations $\mathcal{L} \subseteq \mathcal{R}_q^{Q \times Q-1}$ satisfy that for any matrix $\mathbf{D} = \begin{bmatrix} \mathbf{d}^\top \\ \underline{\mathbf{D}} \end{bmatrix} \in \mathcal{L}$, where $\mathbf{d}^\top$ is the first row of $\mathbf{D}$, $\underline{\mathbf{D}}$ is invertible over $\mathcal{R}_q$.[7]
- $\gamma_{\mathcal{L}} > 0$ is a bound w.r.t. $\mathcal{L}$ such that for any element $\mathbf{D} \in \mathcal{L}$ as above, we have $\gamma_{\mathcal{L}} \geq \|u_i\|_2$ for all $i \in [Q-1]$, where $\mathbf{u} = 2 \cdot \mathbf{d}^\top \underline{\mathbf{D}}^{-1} \in \mathcal{R}_q^{1 \times (Q-1)}$, and u_i is the i-th entry of $\mathbf{u}$.[8]
- $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}^{\mathsf{UMLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}$ for Lemma 4.7, where n is the dimension of $\mathcal{R}_q$ and $\mathcal{B}$ and $\mathcal{B}'$ are constructed from the adversary $\mathcal{A}$ against the $\mathsf{AOM\text{-}UMLWE}$ problem.
- The order of the Rényi divergence $\alpha = \frac{\sigma^*}{\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n} \cdot \sqrt{\frac{-\log(\epsilon_{\mathsf{lattice}})}{Q \cdot k}} \geq 2$ and $\sigma^* \geq \gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n \cdot \sqrt{Q \cdot k}$, chosen to minimize the overall advantage advantage in Lemma 4.7, over all possible choices of Rényi's orders.

We now list the constraints for the proof to hold:

- $\mathsf{UMLWE}_{q,\ell,k,\mathcal{D}_{\sigma_1}}$ is hard, implying $\mathsf{Adv}^{\mathsf{UMLWE}}_{\mathcal{B}}(1^\lambda) = \mathsf{negl}(\lambda)$. i.e., $\sigma_1 \geq \sqrt{\ell} \cdot \omega(\sqrt{\log n})$ using Lemma 3.17.
- $\mathsf{MSIS}_{q,\ell+1,k-\ell,B_{\mathcal{L}}+B_{\mathbf{e}}}$ is hard, implying $\mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) = \mathsf{negl}(\lambda)$. i.e., $q > (B_{\mathcal{L}} + B_{\mathbf{e}}) \cdot \sqrt{n(k-\ell)} \cdot \omega(\log(n(k-\ell)))$ using Lemma 3.18.
- $2^{-\frac{nk}{10}} = \mathsf{negl}(\lambda)$ to bound the norm of samples from discrete Gaussians using Lemma 3.2.

4.4.2 Candidate Asymptotic Parameters.

Finally, we give a set of asymptotic parameters which fit the above constraints. Below it is helpful to keep in mind that the number Q of UMLWE samples and the "quality" $\gamma_{\mathcal{L}}$ of the accepted linear combinations $\mathcal{L}$ dictate the parameters.

Definition 4.4 (Parameters Establishing Hardness of $\mathsf{sel\text{-}AOM\text{-}UMLWE}$). *We denote the set of following asymptotic parameters and conditions along with the restricted accepted linear combinations $\mathcal{L}$ explained above as* $\mathsf{hard\text{-}param}$.

- $n, \ell, k = \mathsf{poly}(\lambda)$ such that $n \geq \lambda$.
- $\mathcal{D}_1 = 2 \cdot \mathcal{D}_{\sigma_1}$ with $\sigma_1 = \sqrt{\ell} \cdot \log n$.
- $\mathcal{D}_i = \mathcal{D}_{\sigma_i}$ for $i \in [2 : Q]$ such that $\sigma^* = \min_{i \in [2:Q]} \sigma_i$.
- $\sigma^* = \gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n \cdot \sqrt{Q \cdot k}$.
- q is the smallest prime larger than $(B_{\mathcal{L}} + B_{\mathbf{e}}) \cdot \sqrt{n(k-\ell)} \cdot \log^2(n(k-\ell))$.
- Plugging in σ^*, $\alpha = \sqrt{-\log(\epsilon_{\mathsf{lattice}})}$ which is larger than 2 assuming hardness of UMLWE and MSIS.

[7] This and the following requirements come from the discussion regarding the last insecure example provided in Section 4.3.

[8] In general, we can allow $\mathbf{u} = v \cdot \mathbf{d}^\top \underline{\mathbf{D}}^{-1}$ for some fixed small polynomial v and replace the factor 2 in $\mathcal{D}_1$ by v. For simplicity, we use 2 as it is the only case relevant for our later instantiation of threshold signatures.

4.4.3 Reduction

Hardness of the "selective" sel-AOM-UMLWE problem is established for the selected parameters through a reduction from standard lattice problems, namely the MSIS and UMLWE problems.

The following is a proof outline of our main Theorem 4.5.

1. Instead of independently and uniformly sampling the $Q-1$ other secrets $\mathbf{s}_2, \ldots, \mathbf{s}_Q$, our initial transformation involves the challenger first uniformly sampling the answer $\mathbf{W}$ corresponding to $\mathbf{SD}$ and then reverse-engineering the corresponding other $Q-1$ secrets so that $\mathbf{W} = \mathbf{SD}$. Importantly, the first secret $\mathbf{s}_1$ is still uniformly sampled from $\mathcal{R}_q$, independent of $\mathbf{W}$.

2. Similarly, we next wish to retain the first error $\mathbf{e}_1$ as a valid error for UMLWE, answer $\mathbf{ED}$ by some $\mathbf{Y}$ sampled independently of $\mathbf{e}_1$, and then reverse-engineer the $Q-1$ other noises $\mathbf{e}_2, \ldots, \mathbf{e}_Q$ to keep the view of the adversary consistent. However, unlike the uniformly random secrets, the noises must follow a specific discrete Gaussian distribution. To this end, we use Rènyi divergence to carefully argue that this modification cannot be detected with overwhelming probability. This is the key step where we use the above restriction on the accepted linear combinations $\mathcal{L}$.[9]

3. Finally, the challenger replaces the construction of the first challenge $\mathbf{A}\mathbf{s}_1 + \mathbf{e}_1$ with a truly uniform element, which is an indistinguishable transformation according to the UMLWE assumption. We conclude the reduction by constructing a MSIS adversary based on an adversary in our modified game. This final step is where we use the condition that $\mathbf{A}$ contains an invertible submatrix (cf. Section 4.2.3).

Theorem 4.5 (UMLWE and MSIS imply sel-AOM-UMLWE with an invertible submatrix). *If there exists an adversary $\mathcal{A}$ against the* $\mathsf{sel\text{-}AOM\text{-}UMLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{e}}}$ *problem with an invertible submatrix (cf. Section 4.2.3), defined with respect to the* hard-param *parameters in Definition 4.4, then we can construct adversaries $\mathcal{B}$ and $\mathcal{B}'$ against the* $\mathsf{UMLWE}_{q,\ell,k,\mathcal{D}_{\sigma_1}}$ *and* $\mathsf{MSIS}_{q,\ell+1,k-\ell,B_{\mathcal{L}}+B_{\mathbf{e}}}$ *problems such that*

$$\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}UMLWE}}_{\mathcal{A}}(1^\lambda) \leq \epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-Q \cdot k \cdot \log(\epsilon_{\mathsf{lattice}})} \cdot \frac{\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n}{\sigma^*}\right) + 2^{-\frac{nk}{10}}.$$

where $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}^{\mathsf{UMLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}$ *and* $\mathsf{Time}(\mathcal{B}), \mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A})$.

Concretely, plugging in hard-param *and assuming the hardness of* UMLWE *and* MSIS*, we have*

$$\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}UMLWE}}_{\mathcal{A}}(1^\lambda) = \mathsf{negl}(\lambda).$$

Proof. Let $\mathcal{A}$ be an adversary against the sel-AOM-UMLWE problem. Below, we consider a sequence of games where the first game is the original game and the last is a game that can be reduced from the MSIS problem. The detail of each game is provided in Fig. 7. We denote $\mathsf{Adv}^{\mathsf{Game}_i}_{\mathcal{A}}(1^\lambda)$ as the advantage of $\mathcal{A}$ in Game_i.

Game_1: This is the real sel-AOM-UMLWE game.

Game_2: In this game, the challenger modifies how the UMLWE secrets except for the first $\mathbf{s}_1$ is set. By the restriction on $\mathcal{L}$, $\underline{\mathbf{D}} \in \mathcal{R}_q^{(Q-1)\times(Q-1)}$ is invertible over $\mathcal{R}_q$. Therefore, since $\mathbf{W}$ is uniform random, the secrets are $[\mathbf{s}_2 \mid \cdots \mid \mathbf{s}_Q]$ independently and uniformly distributed. Moreover, $\mathbf{SD} = \mathbf{W}$ by construction. Thus, we have

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda) = \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda).$$

[9] We can alternatively use the Hint-MLWE assumption by Kim et al. [KLSS23] instead of relying on the Rényi divergence. While we did not opt to do so since the efficiency gain seemed limited, it could be worthwhile to investigate this in more detail in future work.

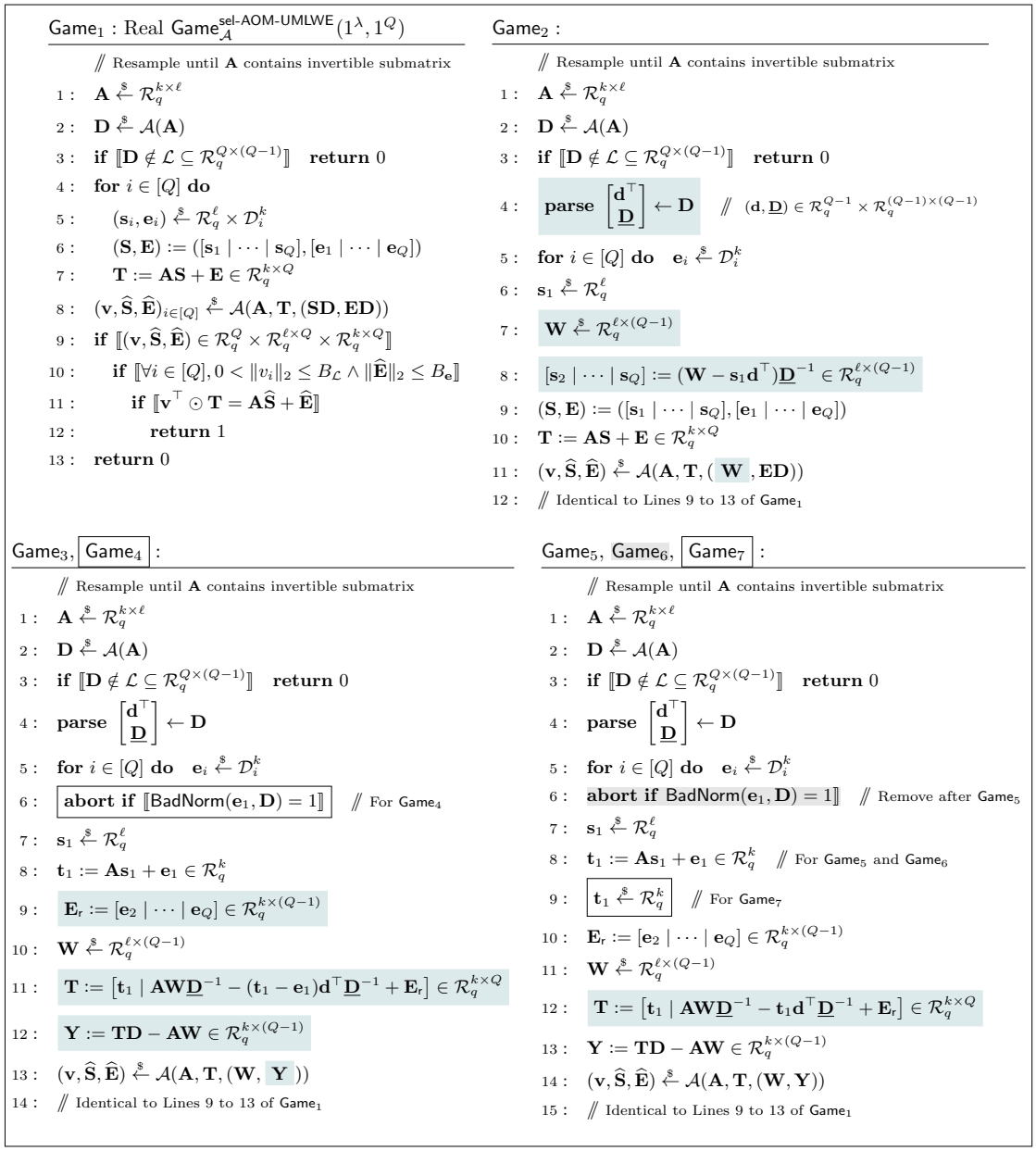

Figure 7: Hybrid games for the proof of Theorem 4.5. Recall the game restricts the adversary to output $\mathbf{D}$ such that $\underline{\mathbf{D}}$ is invertible. Game_4 is the same as Game_3 except that it adds an **abort** condition (i.e., outputs 0 if the condition holds). Game_6 is the same as Game_5 except that it removes the **abort** condition. Game_7 is the same as Game_6 except that it samples $\mathbf{t}_1$ randomly.

Game_3**:** In this game, the challenger computes $\mathbf{T}$ using $(\mathbf{t}_1, \mathbf{e}_1)$ without explicitly using $\mathbf{s}_1$. In particular, this is identical to the previous game by noting the following equality:

$$\begin{aligned}\mathbf{A}[\mathbf{s}_2 \mid \cdots \mid \mathbf{s}_Q] &= \mathbf{A}(\mathbf{W} - \mathbf{s}_1\mathbf{d}^\top)\underline{\mathbf{D}}^{-1} \\ &= \mathbf{A}\mathbf{W}\underline{\mathbf{D}}^{-1} - (\mathbf{A}\mathbf{s}_1)\mathbf{d}^\top\underline{\mathbf{D}}^{-1} \\ &= \mathbf{A}\mathbf{W}\underline{\mathbf{D}}^{-1} - (\mathbf{t}_1 - \mathbf{e}_1)\mathbf{d}^\top\underline{\mathbf{D}}^{-1}\end{aligned}$$

Finally, we use the equality $\mathbf{TD} = \mathbf{AW} + \mathbf{ED}$ to compute the error term $\mathbf{Y}$ provided to the adversary, where recall $\mathbf{W}$ was identical to $\mathbf{SD}$. Thus, we have

$$\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda) = \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda).$$

Game_4**:** In this game, the challenger adds a check on the size of the noise term $\mathbf{e}_1$. Let us define the function $\mathsf{BadNorm}(\mathbf{e}_1, \mathbf{D})$ that outputs 1 if and only if $\|u_i \cdot \widetilde{\mathbf{e}}_1\|_2 \geq e^{1/4}\|u_i\|_1 \cdot \sigma_1 \cdot \sqrt{nk}$ for any $i \in [Q-1]$, where $\widetilde{\mathbf{u}}^\top = \mathbf{d}^\top\underline{\mathbf{D}}^{-1} \in \mathcal{R}_q^{1\times(Q-1)}$, u_i is the i-th elements of $2 \cdot \widetilde{\mathbf{u}}$, and $\mathbf{e}_1 := 2 \cdot \widetilde{\mathbf{e}}_1$. The challenger then aborts the game if $\mathsf{BadNorm}(\mathbf{e}_1, \mathbf{D}) = 1$. Due to Lemma 3.2 and $\widetilde{\mathbf{e}}_1 \xleftarrow{\$} \mathcal{D}^k_{\sigma_1}$, we have

$$\left|\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda) - \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(1^\lambda)\right| \leq 2^{-\frac{nk}{10}}.$$

Game_5**:** In this game, the challenger removes the noise term $\mathbf{e}_1$ when preparing the UMLWE challenge $\mathbf{T}$. We use the Rényi divergence to relate the advantage of this game to the previous game. While the advantages differ non-negligibly, the difference is polynomially related, which suffices for our purpose.[10]

Let us set $\widetilde{\mathbf{u}}^\top = \mathbf{d}^\top\underline{\mathbf{D}}^{-1} \in \mathcal{R}_q^{1\times(Q-1)}$, $\mathbf{u} = 2 \cdot \widetilde{\mathbf{u}}$, and define two distributions $\mathcal{D}_{\vec{\sigma},\widetilde{\mathbf{u}}} = \{2 \cdot \widetilde{\mathbf{e}}_1\widetilde{\mathbf{u}}^\top + \mathbf{E}_r \mid \widetilde{\mathbf{e}}_1 \xleftarrow{\$} \mathcal{D}^k_{\sigma_1}, \mathbf{E}_r \xleftarrow{\$} \prod_{i\in[2:Q]}\mathcal{D}^k_i\}$ and $\mathcal{D}_{\vec{\sigma}} = \{\mathbf{E}_r \mid \mathbf{E}_r \xleftarrow{\$} \prod_{i\in[2:Q]}\mathcal{D}^k_i\}$. Here note that $\mathbf{e}_1 := 2 \cdot \widetilde{\mathbf{e}}_1$ for $\widetilde{\mathbf{e}}_1 \xleftarrow{\$} \mathcal{D}^k_{\sigma_1}$ is the same distribution as $\mathbf{e}_1 \xleftarrow{\$} \mathcal{D}^k_1$. Since the only difference between Game_4 and Game_5 is whether $2 \cdot \widetilde{\mathbf{e}}_1 \cdot \widetilde{\mathbf{u}}$ is used or not, we have the following:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(1^\lambda) &\leq \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot R_\alpha(\mathcal{D}_{\vec{\sigma},\mathbf{u}}; \mathcal{D}_{\vec{\sigma}}) \\ &\leq \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \prod_{i\in[2:Q]} \exp\left(\frac{\alpha\|u_{i-1} \cdot \widetilde{\mathbf{e}}_1\|_2^2}{2\sigma_i^2}\right) \\ &\leq \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \prod_{i\in[2:Q]} \exp\left(\frac{\alpha \cdot \left(e^{1/4}\|u_{i-1}\|_1 \cdot \sigma_1 \cdot \sqrt{nk}\right)^2}{2\sigma_i^2}\right) \\ &\leq \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \exp\left(\frac{Q \cdot \alpha \cdot k \cdot (\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n)^2}{\sigma^{*2}}\right). \qquad (4)\end{aligned}$$

The first inequality follows from Lemma 3.5, Items 1 and 2, the second follows from Lemma 3.5, Item 3 and Lemma 3.6, the third follows from Lemma 3.2 and the **abort** condition we added in Game_4, and the last follows from the definitions of σ^* and $\gamma_{\mathcal{L}}$ and the facts $\|a\|_1 \leq \sqrt{n} \cdot \|a\|_2$ for $a \in \mathbb{R}_q$ and $\sqrt{e}/2 < 1$. Here, note that we can properly invoke Lemma 3.5, Item 3 since each entry of $\mathcal{D}_{\vec{\sigma},\widetilde{\mathbf{u}}}$ are distributed independently once $2 \cdot \widetilde{\mathbf{e}}_1 \cdot \widetilde{\mathbf{u}}$ is fixed. We proceed with the hybrid games to prove that $\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(1^\lambda)$ and $\mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)$ are polynomially related for our selection of α.

Game_6**:** In this game, the challenger undo the abort check added in Game_4. Following the same argument, we have

$$\left|\mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda) - \mathsf{Adv}^{\mathsf{Game}_6}_{\mathcal{A}}(1^\lambda)\right| \leq 2^{-\frac{nk}{10}}.$$

[10] From a theoretical perspective, we can rely on the statistical distance by simply assuming $\mathcal{D}_2$ is a discrete Gaussian with width super-polynomially larger than the size of $\mathbf{e}_1\mathbf{u}$. For concrete efficiency, we rely on a more fine-grained analysis by using the upper bound Q and the Rényi divergence.

Game_7**:** Lastly, in this game, the challenger samples $\mathbf{t}_1$ uniformly random over $\mathcal{R}_q^k$ instead of setting it as a valid UMLWE sample. Note that the challenger no longer requires knowledge of the secret and noise $(\mathbf{s}_1, \mathbf{e}_1)$, and in particular, run the game only using $\mathbf{t}_1$. Moreover, due to the modification we made in Game_6, $\mathbf{e}_1$ is distributed exactly as in a valid UMLWE sample, multiplied by 2. Since q is odd, it is easy to check that we can construct an UMLWE adversary $\mathcal{B}$ that internally runs $\mathcal{A}$ solving the (decisional) $\mathsf{UMLWE}_{q,\ell,k,\mathcal{D}_{\sigma_1}}$ problem such that

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_6}(1^\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_7}(1^\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{UMLWE}}(1^\lambda).$$

In the above, an attentive reader may have noticed that we assumed the hardness of UMLWE where $\mathbf{A}^\top \xleftarrow{\$} \mathsf{GL}_\ell(\mathcal{R}_q) \times \mathcal{R}_q^{\ell\times(k-\ell)}$, rather than the standard UMLWE where $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$. However, this is without loss of generality since the hardness of the latter implies the hardness of the former assuming the probability of a random matrix sampled from $\mathcal{R}_q^{\ell\times\ell}$ is invertible with non-negligible probability.

We show in Lemma 4.6 that there we can construct an MSIS adversary $\mathcal{B}'$ that internally runs $\mathcal{A}$ solving the $\mathsf{MSIS}_{q,\ell+1,k-\ell,B_{\mathcal{L}}+B_{\mathbf{e}}}$ problem such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_7}(1^\lambda) \leq \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{MSIS}}(1^\lambda).$$

Before providing the proof of Lemma 4.6, we finish the proof of Theorem 4.5.

Collecting the bounds, we obtain

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(1^\lambda) \leq \left(\mathsf{Adv}_{\mathcal{B}}^{\mathsf{UMLWE}}(1^\lambda) + \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{MSIS}}(1^\lambda) + 2^{-\frac{nk}{10}}\right)^{\frac{\alpha-1}{\alpha}} \\ \cdot \exp\left(\frac{Q\cdot\alpha\cdot k\cdot(\gamma_{\mathcal{L}}\cdot\sigma_1\cdot n)^2}{\sigma^{*2}}\right) + 2^{-\frac{nk}{10}}.\end{aligned}$$

Plugging our choices of parameters $\mathsf{hard\text{-}param}$ (remark here that the choice of α was made to minimize the latter expression), we obtain

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(1^\lambda) \leq \epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-Q\cdot k\cdot\log(\epsilon_{\mathsf{lattice}})}\cdot\frac{\gamma_{\mathcal{L}}\cdot\sigma_1\cdot n}{\sigma^*}\right) + 2^{-\frac{nk}{10}},$$

where $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}_{\mathcal{B}}^{\mathsf{UMLWE}}(1^\lambda) + \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{MSIS}}(1^\lambda) + 2^{-\frac{nk}{10}}$. We finally show in Lemma 4.7 that the right hand side is negligible, assuming the hardness of the UMLWE and MSIS problem. This completes the proof of Theorem 4.5. □

It remains to prove the following two Lemmata 4.6 and 4.7.

Lemma 4.6. *There exists an adversary $\mathcal{B}'$ that internally runs $\mathcal{A}$ solving the $\mathsf{MSIS}_{q,\ell+1,k-\ell,B_{\mathcal{L}}+B_{\mathbf{e}}}$ problem such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_7}(1^\lambda) \leq \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{MSIS}}(1^\lambda).$$

Moreover, we have $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A})$.

Proof. Let $\mathcal{A}$ be an adversary against the $\mathsf{sel\text{-}AOM\text{-}UMLWE}$ problem in Game_7. We construct an adversary $\mathcal{B}'$ solving the MSIS problem having the same advantage as $\mathcal{A}$. Assume $\mathcal{B}'$ is given $\mathbf{M} = [\mathbf{h} \mid \overline{\mathbf{M}}] \in \mathcal{R}_q^{(k-\ell)\times(\ell+1)}$ as the MSIS problem where $\mathbf{h} \in \mathcal{R}_q^{k-\ell}$. It then samples a random $(\mathbf{A}', \mathbf{t}') \xleftarrow{\$} \mathsf{GL}_\ell(\mathcal{R}_q) \times \mathcal{R}_q^{k-\ell}$ and sets

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}' \\ -\overline{\mathbf{M}}\mathbf{A}'\end{bmatrix} \in \mathcal{R}_q^{k\times\ell} \quad \text{and} \quad \mathbf{t}_1 = \begin{bmatrix}\mathbf{t}' \\ \mathbf{h} - \overline{\mathbf{M}}\mathbf{t}'\end{bmatrix} \in \mathcal{R}_q^k. \tag{5}$$

It then simulates Game_7 to $\mathcal{A}$, where the only difference is that it uses the above computed $(\mathbf{A}, \mathbf{t}_1)$ rather than sampling them. At the end of the game, $\mathcal{A}$ outputs an approximate UMLWE solution $(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}})$. $\mathcal{B}'$

then sets $(v_1, \widehat{\mathbf{s}}, \widehat{\mathbf{e}}) \in \mathcal{R}_q \times \mathcal{R}_q^\ell \times \mathcal{R}_q^k$ as the first entry and columns of $\mathbf{v}$ and $(\widehat{\mathbf{S}}, \widehat{\mathbf{E}})$, respectively, and outputs $\mathbf{s}^* = \begin{bmatrix} v_1 \\ -\widehat{\mathbf{e}} \end{bmatrix} \in \mathcal{R}_q^{k+1}$ as the MSIS solution.

Let us analyze the success probability of $\mathcal{B}'$. Clearly, we have $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A})$. Moreover, since $\mathbf{A}'$ is invertible, the instance given to $\mathcal{A}$ is identical to those provided in the $\mathsf{sel\text{-}AOM\text{-}UMLWE}$ game in Game_7.[11] Now, if $\mathcal{A}$ breaks $\mathsf{sel\text{-}AOM\text{-}UMLWE}$ in Game_7, we have $v_1 \cdot \mathbf{t}_1 = \mathbf{A}\widehat{\mathbf{s}} + \widehat{\mathbf{e}}$. Due to Eq. (5), if we left multiply $[\overline{\mathbf{M}} \mid \mathbf{I}]$ to the equation, we have

$$v_1 \cdot \mathbf{h} - [\overline{\mathbf{M}} \mid \mathbf{I}]\widehat{\mathbf{e}} = [\mathbf{h} \mid \overline{\mathbf{M}} \mid \mathbf{I}]\mathbf{s}^* = \mathbf{0}.$$

Due to $\mathcal{A}$'s winning condition, we have $v_1 \neq 0$. This implies that $\mathbf{s}^* \neq \mathbf{0}$ as desired. Moreover, we can bound the size of $\mathbf{s}^*$ as $\|\mathbf{s}^*\|_2 \leq B_{\mathcal{L}} + B_{\mathbf{e}}$. This completes the proof. □

Lemma 4.7. *Under the assumption that* $\mathsf{UMLWE}_{q,\ell,k,\mathcal{D}_1}$ *and* $\mathsf{MSIS}_{q,\ell+1,k-\ell,B_{\mathcal{L}}+B_{\mathbf{e}}}$ *are hard, we have the following plugging in our parameter selection* $\mathsf{hard\text{-}param}$*:*

$$\epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-Q \cdot k \cdot \log(\epsilon_{\mathsf{lattice}})} \cdot \frac{\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n}{\sigma^*}\right) = \mathsf{negl}(\lambda).$$

Proof. Plugging in our choice of σ^* to the right hand side, we have $\epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-\log(\epsilon_{\mathsf{lattice}})}\right)$. It can be checked that for any negligible function f, $f \cdot \exp(\sqrt{-\log(f)}) = \mathsf{negl}(\lambda)$. Hence, assuming the hardness of the UMLWE and MSIS problem, we conclude that the term is indeed negligible as desired. □

4.5 Example of Accepted Linear Combination $\mathcal{L} = \mathcal{L}_{\mathsf{TS}}$

Lastly, we provide a concrete example of an accepted linear combination $\mathcal{L}$ satisfying the constraints in Section 4.4. The $\mathcal{L}$ we consider in this section appears in the threshold signature scheme presented in Section 5. We define $\mathcal{L}_{\mathsf{TS}} := \mathcal{L}$. More specifically, when we reduce the unforgeability of our threshold signature scheme from the $\mathsf{AOM\text{-}MLWE}$ problem, the set of vectors the reduction queries to the MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$ is guaranteed to be in $\mathcal{L}_{\mathsf{TS}}$. Below, $\mathcal{L}_{\mathsf{TS}}$ is defined by two sets $\mathcal{C}$ and $\mathbb{T}$, where $\mathcal{C}$ is the so-called challenge set of the threshold signature scheme consisting of $\{-1, 0, 1\}$-coefficient polynomials with fixed hamming weight $W > 0$ and $\mathbb{T}$ is the set of signed monomials. Recall Section 3.2 for their definitions. Formally, $\mathcal{L}_{\mathsf{TS}}$ is defined as follows.

Definition 4.8 (Accepted Linear Combinations $\mathcal{L}_{\mathsf{TS}}$ for Threshold Signature). *Let $\mathcal{C}$ and $\mathbb{T}$ be the sets defined in Section 3.2. Let τ and Q' be integers such that $\tau \geq 2$ and set $Q = \tau \cdot Q' + 1$. Let $\mathcal{P}_k$ be the set of permutation matrices of size $k > 0$. Define two sets $\mathcal{C}_{\mathsf{TS}}$ and $\mathcal{B}_{\mathsf{TS}}$ as follows:*

- $\mathcal{C}_{\mathsf{TS}} = \left\{[c, c', 0, \cdots, 0]^\top \in \mathcal{R}_q^\tau \mid c, c' \in \mathcal{C} \cup \{0\}\right\}$. *I.e., a set of row vectors where the first two entries are in $\mathcal{C} \cup \{0\}$ and the remaining $\tau - 2$ entries are zero.*
- $\mathcal{B}_{\mathsf{TS}} = \left\{ \begin{bmatrix} 1 & 1 & & & \\ b_1 & b_1' & & & \\ b_2 & b_2' & 1 & & \\ \vdots & \vdots & & \ddots & \\ b_{\tau-1} & b_{\tau-1}' & & & 1 \end{bmatrix} \in \mathcal{R}_q^{\tau \times \tau} \;\middle|\; \begin{array}{c} \forall i \in [\tau - 1], (b_i, b_i') \in \mathbb{T}^2 \\ \wedge\ b_1 \neq b_1' \end{array} \right\}$. *I.e., a set of invertible matrices where the first two columns consist of entries in $\mathbb{T}$, the first two rows are full-rank, and the remaining entries consist of an identity matrix of dimension $\tau - 2$.*

[11] This is the critical step where we rely on the restriction that $\mathbf{A}$ contains an invertible submatrix of size $(\ell \times \ell)$. If $\mathbf{A}$ did not contain such a submatrix, we will no longer be able to argue above that $\overline{\mathbf{M}}\mathbf{A}'$ is uniform random over $\mathcal{R}_q^{(k-\ell)\times\ell}$.

Then, define the set of accepted linear combinations $\mathcal{L}_{\mathsf{TS}}$ *as follows:*

$$\mathcal{L}_{\mathsf{TS}} = \left\{ \begin{bmatrix} 1 & \\ & \mathbf{P}_{\mathsf{row}} \end{bmatrix} \begin{bmatrix} \mathbf{c}_1^\top & \mathbf{c}_2^\top & \cdots & \mathbf{c}_{Q'}^\top \\ \mathbf{B}_1 & & & \\ & \mathbf{B}_2 & & \\ & & \ddots & \\ & & & \mathbf{B}_{Q'} \end{bmatrix} \cdot \mathbf{P}_{\mathsf{column}} \subset \mathcal{R}_q^{Q\times(Q-1)} \;\middle|\; \begin{array}{l} \forall i \in [Q'], (\mathbf{c}_i, \mathbf{B}_i) \in \mathcal{C}_{\mathsf{TS}} \times \mathcal{B}_{\mathsf{TS}}, \\ (\mathbf{P}_{\mathsf{row}}, \mathbf{P}_{\mathsf{column}}) \in \mathcal{P}_{Q-1}^2 \end{array} \right\}.$$

The following shows that $\mathcal{L}_{\mathsf{TS}}$ satisfies the condition required to establish the hardness of sel-AOM-MLWE problem via Theorem 4.5.

Lemma 4.9. *The set of accepted linear combinations* $\mathcal{L}_{\mathsf{TS}}$ *defined in Definition 4.8 satisfies the condition imposed by* hard-param *defined in Definition 4.4, where* $\gamma_{\mathcal{L}_{\mathsf{TS}}} = 2 \cdot W\sqrt{n}$.

Concretely, for any matrix $\mathbf{D} = \begin{bmatrix} \mathbf{d}^\top \\ \underline{\mathbf{D}} \end{bmatrix} \in \mathcal{L}_{\mathsf{TS}}$, *where* $\mathbf{d}^\top$ *is the first row of* $\mathbf{D}$, $\underline{\mathbf{D}}$ *is invertible over* $\mathcal{R}_q$. *Moreover, we have* $\gamma_{\mathcal{L}_{\mathsf{TS}}} \geq \|u_i\|_2$ *for all* $i \in [Q-1]$, *where* $\mathbf{u} = 2 \cdot \mathbf{d}^\top \underline{\mathbf{D}}^{-1} \in \mathcal{R}_q^{1\times(Q-1)}$ *and* u_i *is the* i*-th entry of* $\mathbf{u}$.

Proof. By Lemma 3.1, any matrix $\mathbf{B} \in \mathcal{B}_{\mathsf{TS}}$ is invertible. Specifically, we have

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & & \\ b_1 & b_1' & & \\ b_2 & b_2' & 1 & \\ \vdots & \vdots & & \ddots \\ b_{\tau-1} & b_{\tau-1}' & & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B}^{-1} = (b_1' - b_1)^{-1} \cdot \begin{bmatrix} b_1' & -1 & & & \\ -b_1 & 1 & & & \\ * & * & 1 & & \\ * & * & & \ddots & \\ * & * & & & 1 \end{bmatrix}, \tag{6}$$

where note that $2(b_1' - b_1)$ is invertible by Lemma 3.1 and $*$ denotes an arbitrary element in $\mathcal{R}_q$. Since $\underline{\mathbf{D}}$ is a matrix that can be obtained by applying a row and column permutations to a block diagonal matrix with entries in $\mathcal{B}_{\mathsf{TS}}$, $\underline{\mathbf{D}}$ is invertible as desired.

It remains to check the L_2-norm of $\mathbf{u} = 2 \cdot \mathbf{d}^\top \underline{\mathbf{D}}^{-1}$ is small. First notice that permutations do not alter the size of the vector, hence we can ignore them without loss of generality. Moreover, since $\underline{\mathbf{D}}$ is block diagonal, we can focus on the case $Q' = 1$. That is, it remains to establish that $\gamma_{\mathcal{L}_{\mathsf{TS}}} \geq \|u_i\|_2$ for all $i \in [\tau]$, where $\mathbf{u}' = 2 \cdot \mathbf{c}^\top \mathbf{B}^{-1}$ for any $(\mathbf{c}, \mathbf{B}) \in \mathcal{C}_{\mathsf{TS}} \times \mathcal{B}_{\mathsf{TS}}$ and u_i is the i-th entry. Let $\mathbf{c} = [c, c', 0, \cdots 0]$ for $c, c' \in \mathcal{C} \cup \{0\}$. Then, plugging in Eq. (6),

$$2 \cdot \mathbf{c}^\top \mathbf{B}^{-1} = [2(b_1' - b_1)^{-1} \cdot (cb_1' - c'b_1), 2(b_1' - b_1)^{-1} \cdot (-c + c'), 0, \cdots, 0] \in \mathcal{R}_q^\tau. \tag{7}$$

Using Lemma 3.1 and the fact that b_1, b_1' are monomials, we can bound the first entry using $\|2(b_1' - b_1)^{-1} \cdot (cb_1')\|_2 \leq \|2(b_1 - b_1')^{-1}\|_2 \cdot \|cb_1'\|_1 \leq W\sqrt{n} = 2^{-1} \cdot \gamma_{\mathcal{L}_{\mathsf{TS}}}$, where we use the Minkowski inequality. The same bound holds for the second entry. This completes the proof. □

Remark 4.10. By considering our specific $\mathcal{L}_{\mathsf{TS}}$ in the proof of Theorem 4.5, we can obtain better asymptotic parameters than those in Definition 4.4. Specifically, we can take σ^* to $\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n \cdot \sqrt{2Q' \cdot k}$ instead of $\gamma_{\mathcal{L}} \cdot \sigma_1 \cdot n \cdot \sqrt{\tau Q' \cdot k}$. This is because we can bound the Rényi divergence in Eq. (4) independently of τ. Recall that u_i in Eq. (4) is i-th entry of $\mathbf{u} = 2\mathbf{d}^\top \underline{\mathbf{D}}^{-1} \in \mathcal{R}_q^{\tau Q'}$. There are at most $2Q'$ non-zero entries in $\mathbf{u}$ since there are at most 2 non-zero entries in $2 \cdot \mathbf{c}^\top \mathbf{B}^{-1}$ for any $(\mathbf{c}, \mathbf{B}) \in \mathcal{C}_{\mathsf{TS}} \times \mathcal{B}_{\mathsf{TS}}$, as shown in Eq. (7). Thus, we can obtain a tighter upper bound.

5 Construction of Our Two-Round Threshold Signature

In this section, we present our two-round threshold signature scheme $\mathsf{TS}_{\text{2-round}}$.

Parameters. For reference, we provide in Table 1 the parameters used in the scheme. Parameters related to the security proof are provided in Section 6.

Parameter	Explanation
$\mathcal{R}_q$	Polynomial ring $\mathcal{R}_q = \mathbb{Z}[X]/(q, X^n + 1)$
(k, ℓ)	Dimension of public matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$
$(\mathcal{D}_{\mathbf{t}}, \sigma_{\mathbf{t}})$	Gaussian distribution with width $\sigma_{\mathbf{t}}$ used for the verification key $\mathbf{t}$
$(\mathcal{D}_{\mathbf{w}}, \sigma_{\mathbf{w}})$	Gaussian distribution with width $\sigma_{\mathbf{w}}$ used for the commitment $\mathbf{w}$
$\nu_{\mathbf{t}}$	Amount of bit dropping performed on verification key
$\nu_{\mathbf{w}}$	Amount of bit dropping performed on (aggregated) commitment
$(q_{\nu_{\mathbf{t}}}, q_{\nu_{\mathbf{w}}})$	Rounded moduli satisfying $(q_{\nu_{\mathbf{t}}}, q_{\nu_{\mathbf{w}}}) := (\lfloor q/2^{\nu_{\mathbf{t}}} \rfloor, \lfloor q/2^{\nu_{\mathbf{w}}} \rfloor) = (\lfloor q/2^{\nu_{\mathbf{t}}} \rceil, \lfloor q/2^{\nu_{\mathbf{w}}} \rceil)$
$\mathbb{T} \subset \mathcal{R}_q$	Set of signed monomials (see Section 3)
rep	An integer s.t. $\lvert\mathbb{T}\rvert^{\mathsf{rep}-1} \geq 2^\lambda$
$(\mathcal{C} \subset \mathcal{R}_q, W)$	Challenge set $\{c \in \mathcal{R}_q \mid \lVert c\rVert_\infty = 1 \wedge \lVert c\rVert_1 = W\}$ s.t. $\lvert\mathcal{C}\rvert \geq 2^\lambda$
B	Two-norm bound on the signature

Table 1: Overview of parameters used in our two-round threshold signature.

Construction. The construction of our two-round threshold signature $\mathsf{TS}_{\text{2-round}}$ is provided in Fig. 8. Our scheme uses two hash functions modeled as a random oracle in the security proof. $\mathsf{G} : \{0,1\}^* \to \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$ is used to aggregate the *individual* commitments into one commitment; that is, each user outputs rep commitments in the pre-processing phase and G is used to aggregate them. $\mathsf{H} : \{0,1\}^* \to \mathcal{C}$ is used to generate the random challenge polynomial for which the users reply with a response. Note that H is the typical hash function that appears in Fiat-Shamir based signatures. While we define G and H to take an arbitrary bit string as input, it is understood that in practice, we check the format of these inputs. Moreover, as standard practice, the two hash functions can be derived from a single hash function using appropriate domain separation.

A peculiarity of our construction is that the verification key $\mathbf{t}$ is generated using $2 \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ rather than the more conventional $\mathbf{A} \cdot \mathbf{s} + \mathbf{e}$. While from an algorithmic point of view, this has almost no impact on the signing and verification algorithms, it is vital when establishing security based on our AOM-MLWE assumption. Specifically, this is used to invoke Lemma 4.9, establishing that the adversary's queries fall into the accepted linear combinations $\mathcal{L}_{\mathsf{TS}}$ required to argue hardness of AOM-MLWE. It is not clear whether this is an artifact of our proof and we leave it as an interesting problem to remove the factor 2 from our construction.

Remark 5.1 (Avoiding sending the row masks). In the previous version of our scheme [EKT24], the row mask $\mathbf{m}_i$ was explicitly sent to other users. Very recently, Katsumata et al. [KRT24] provided a way to avoid this. Specifically, instead of adding only the column mask $\mathbf{m}_i^*$ to the response, $\mathbf{m}_i^* - \mathbf{m}_i$ is added as $\mathbf{z}_i = c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}_i^* - \mathbf{m}_i$. Since $\sum_{i \in \mathsf{SS}} (\mathbf{m}_i^* - \mathbf{m}_i) = \sum_{i \in \mathsf{SS}} \mathbf{m}_i^* - \sum_{i \in \mathsf{SS}} \mathbf{m}_i = 0$ holds, the masking terms disappear when responses $\mathbf{z}_i$ for all users in SS are summed up in the aggregation algorithm. Thus, each user no longer needs to output the row mask. Our scheme in Fig. 8 is obtained by applying this optimization to the previous version.

The following establishes the correctness of our scheme.

Theorem 5.2 (Correctness). *The two-round threshold signature* $\mathsf{TS}_{\text{2-round}}$ *in Fig. 8 is correct if* $(W \cdot 2^{\nu_{\mathbf{t}}} + 2^{\nu_{\mathbf{w}}}) \cdot \sqrt{nk} + e^{1/4} \cdot (2W \cdot \sigma_{\mathbf{t}} + \sigma_{\mathbf{w}} \cdot \sqrt{\mathsf{rep} \cdot N}) \cdot \sqrt{n} \cdot (\sqrt{k} + \sqrt{\ell}) \leq B$, $\sigma_{\mathbf{w}} > \sqrt{\frac{\log(2nk)+\lambda}{\pi}}$, *and assuming* $(q, \nu_{\mathbf{t}}, \nu_{\mathbf{w}})$ *satisfies the condition in Table 1.*

Proof. It is clear that when the signatures are generated honestly the check $[\![c = c']\!]$ inside the verification algorithm always holds. We thus focus on the check on the L_2-norm. Below, we will be precise on where each element lives and be explicit about our use of the lift notation, i.e., $\bar{x} \in [0, 1, \cdots, q-1]$ for $x \in \mathcal{R}_q$ (see Section 3.7.2 for more detail).

$\mathsf{TS.Setup}(1^\lambda, N, T)$

$\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$
$\mathsf{tspar} := (\mathbf{A}, N, T)$
return tspar

$\mathsf{TS.KeyGen}(\mathsf{tspar})$

parse $(\mathbf{A}, N, T) \leftarrow \mathsf{tspar}$
$(\mathbf{s}, \mathbf{e}) \xleftarrow{\$} \mathcal{D}_{\mathbf{t}}^\ell \times \mathcal{D}_{\mathbf{t}}^k$
$\mathbf{t} := \lfloor 2 \cdot (\mathbf{A}\mathbf{s} + \mathbf{e}) \rceil_{\nu_{\mathbf{t}}} \in \mathcal{R}_{q_{\nu_{\mathbf{t}}}}^k$
for $(i, j) \in [N] \times [N]$ **do**
  $\mathsf{seed}_{i,j} \xleftarrow{\$} \{0,1\}^\lambda$
$\vec{P} \xleftarrow{\$} \mathcal{R}_q^\ell[X]$ with $\deg(\vec{P}) = T-1, \vec{P}(0) = 2\cdot\mathbf{s}$
$(\mathbf{s}_i)_{i\in[N]} := (\vec{P}(i))_{i\in[N]}$
$\mathsf{vk} := (\mathsf{tspar}, \mathbf{t})$
$(\mathsf{sk}_i)_{i\in[N]} := \Big((\mathbf{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[N]})\Big)_{i\in[N]}$
return $(\mathsf{vk}, (\mathsf{sk}_i)_{i\in[N]})$

$\mathsf{TS.Agg}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, (\widehat{\mathsf{sig}}_j)_{j\in\mathsf{SS}})$

parse $(\mathsf{tspar}, \mathbf{t}) \leftarrow \mathsf{vk}$
parse $(\mathbf{A}, N, T) \leftarrow \mathsf{tspar}$
parse $(\mathbf{w}_j, \mathbf{z}_j)_{j\in\mathsf{SS}} \leftarrow (\widehat{\mathsf{sig}}_j)_{j\in\mathsf{SS}}$
$\mathbf{w} := \left\lfloor \sum_{j\in\mathsf{SS}} \mathbf{w}_j \right\rceil_{\nu_{\mathbf{w}}} \in \mathcal{R}_{q_{\nu_{\mathbf{w}}}}^k$
$\mathbf{z} := \sum_{j\in\mathsf{SS}} \mathbf{z}_j \in \mathcal{R}_q^\ell$
$c := \mathsf{H}(\mathsf{vk}, \mathsf{M}, \mathbf{w})$
$\mathbf{y} := \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} \cdot c \cdot \mathbf{t} \rceil_{\nu_{\mathbf{w}}} \in \mathcal{R}_{q_{\nu_{\mathbf{w}}}}^k$
$\mathbf{h} := \mathbf{w} - \mathbf{y} \in \mathcal{R}_{q_{\nu_{\mathbf{w}}}}^k$
return $\mathsf{sig} := (c, \mathbf{z}, \mathbf{h})$

$\mathsf{TS.Verify}(\mathsf{vk}, \mathsf{M}, \mathsf{sig})$

parse $(c, \mathbf{z}, \mathbf{h}) \leftarrow \mathsf{sig}$
$c' := \mathsf{H}(\mathsf{vk}, \mathsf{M}, \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} \cdot c \cdot \mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h})$
if $[\![c = c']\!] \wedge [\![\|(\mathbf{z}, 2^{\nu_{\mathbf{w}}} \cdot \mathbf{h})\|_2 \le B]\!]$ **then**
  return 1
return 0

$\mathsf{TS.PP}(\mathsf{vk}, i, \mathsf{sk}_i, \mathsf{st}_i)$

parse $(\mathsf{tspar}, \mathbf{t}) \leftarrow \mathsf{vk}$
parse $(\mathbf{A}, N, T) \leftarrow \mathsf{tspar}$
for $b \in [\mathsf{rep}]$ **do**
  $(\mathbf{r}_{i,b}, \mathbf{e}'_{i,b}) \xleftarrow{\$} \mathcal{D}_{\mathbf{w}}^\ell \times \mathcal{D}_{\mathbf{w}}^k$
  $\mathbf{w}_{i,b} := \mathbf{A}\mathbf{r}_{i,b} + \mathbf{e}'_{i,b} \in \mathcal{R}_q^k$
$\vec{\mathbf{w}}_i := [\mathbf{w}_{i,1} \mid \cdots \mid \mathbf{w}_{i,\mathsf{rep}}]$
$\mathsf{pp}_i := \vec{\mathbf{w}}_i$
$\mathsf{st}_i \leftarrow \mathsf{st}_i \cup \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$
return $(\mathsf{pp}_i, \mathsf{st}_i)$

$\mathsf{TS.Sign}(\mathsf{vk}, \mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}}, \mathsf{sk}_i, \mathsf{st}_i)$

parse $\big(\mathbf{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[N]}\big) \leftarrow \mathsf{sk}_i$
req $[\![\mathsf{SS} \subseteq [N]]\!] \wedge [\![i \in \mathsf{SS}]\!] \wedge [\![(\mathsf{pp}_i, \cdot) \in \mathsf{st}_i]\!]$
parse $(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}\setminus\{i\}} \leftarrow (\mathsf{pp}_j)_{j\in\mathsf{SS}}$
pick $(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})$ **from** st_i with $\mathsf{pp}_i = \vec{\mathbf{w}}_i$
$\mathsf{ctnt} := \mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$
$(\beta_b)_{b\in[\mathsf{rep}]} := \mathsf{G}(\mathsf{vk}, \mathsf{ctnt}) \quad /\!/\ \beta_1 = 1, \beta_b \in \mathbb{T}$
for $j \in \mathsf{SS}$ **do**
  parse $[\mathbf{w}_{j,1} \mid \cdots \mid \mathbf{w}_{j,\mathsf{rep}}] \leftarrow \vec{\mathbf{w}}_j$
  $\mathbf{w}_j := \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{w}_{j,b} \in \mathcal{R}_q^k$
$\mathbf{w} := \left\lfloor \sum_{j\in\mathsf{SS}} \mathbf{w}_j \right\rceil_{\nu_{\mathbf{w}}} \in \mathcal{R}_{q_{\nu_{\mathbf{w}}}}^k$
$c := \mathsf{H}(\mathsf{vk}, \mathsf{M}, \mathbf{w}) \quad /\!/\ c \in \mathcal{C}$
$\mathbf{m}_i := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt}) \in \mathcal{R}_q^\ell$
$\mathbf{m}_i^* := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt}) \in \mathcal{R}_q^\ell$
$\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}_i^* - \mathbf{m}_i \in \mathcal{R}_q^\ell$
$\mathsf{st}_i \leftarrow \mathsf{st}_i \setminus \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$
$\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$
return $(\widehat{\mathsf{sig}}_i, \mathsf{st}_i)$

Figure 8: Our two-round threshold signature $\mathsf{TS}_{\text{2-round}}$. In the above, $L_{\mathsf{SS},i}$ denotes the Lagrange coefficient of user i in the set $\mathsf{SS} \subseteq [N]$ (see Section 3.4 for the definition). **pick** X **from** Y denotes the process of picking an element X from the set Y.

Fix any $(N, T, \mathsf{SS} \subset [N])$ such that $|\mathsf{SS}| = T$. Then, it can be checked that

$$\begin{aligned}\mathbf{z} &= \sum_{j \in \mathsf{SS}} \mathbf{z}_j \\ &= \sum_{j \in \mathsf{SS}} (c \cdot L_{\mathsf{SS},j} \cdot \mathbf{s}_j + \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{j,b} + \mathbf{m}^*_j - \mathbf{m}_j) \\ &= 2 \cdot c \cdot \mathbf{s} + \sum_{j \in \mathsf{SS}} \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{j,b} \in \mathcal{R}_q^\ell,\end{aligned}$$

where the last equality follows from the correctness of the linear Shamir secret sharing scheme and the fact that $\sum_{j \in \mathsf{SS}} \mathbf{m}^*_j = \sum_{j \in \mathsf{SS}} \mathbf{m}_j$. We then have

$$\begin{aligned}\mathbf{y} &= \left\lfloor \mathbf{Az} - 2^{\nu_\mathbf{t}} \cdot c \cdot \bar{\mathbf{t}} \right\rceil_{\nu_\mathbf{w}} \\ &= \left\lfloor 2 \cdot c \cdot \mathbf{As} + \sum_{j \in \mathsf{SS}} \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{Ar}_{j,b} - 2^{\nu_\mathbf{t}} \cdot c \cdot \mathbf{t} \right\rceil_{\nu_\mathbf{w}} \\ &= \left\lfloor 2 \cdot c \cdot (\bar{\mathbf{t}} - \mathbf{e}) + \sum_{j \in \mathsf{SS}} (\mathbf{w}_j - \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{e}'_{j,b}) - 2^{\nu_\mathbf{t}} \cdot c \cdot \mathbf{t} \right\rceil_{\nu_\mathbf{w}} \\ &= \left\lfloor \bar{\mathbf{w}} + c \cdot \underbrace{(2 \cdot \bar{\mathbf{t}} - 2^{\nu_\mathbf{t}} \cdot \lfloor 2 \cdot \bar{\mathbf{t}} \rceil_{\nu_\mathbf{t}})}_{=:\boldsymbol{\alpha}_\mathbf{t} \in \mathcal{R}_q^k} - \underbrace{\left(2 \cdot c \cdot \mathbf{e} + \sum_{j \in \mathsf{SS}} \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{e}'_{j,b}\right)}_{=:\boldsymbol{\alpha} \in \mathcal{R}_q^k} \right\rceil_{\nu_\mathbf{w}},\end{aligned}$$

where $\bar{\mathbf{t}} = \mathbf{As} + \mathbf{e} \in \mathcal{R}_q^k$, $\bar{\mathbf{w}} = \sum_{j \in \mathsf{SS}} \mathbf{w}_j \in \mathcal{R}_q^k$, and note that $\mathbf{t} = \lfloor 2 \cdot \bar{\mathbf{t}} \rceil_{\nu_\mathbf{t}} \in \mathcal{R}_{q_{\nu_\mathbf{t}}}^k$ and $\mathbf{w} = \lfloor \bar{\mathbf{w}} \rceil_{\nu_\mathbf{w}} \in \mathcal{R}_{q_{\nu_\mathbf{w}}}^k$.

Plugging the above $\mathbf{y}$ and using Lemma 3.14 Eq. (3), we have

$$\begin{aligned}\|2^{\nu_\mathbf{w}} \cdot \bar{\mathbf{h}} \mod q\|_2 &= \|2^{\nu_\mathbf{w}} \cdot \overline{\mathbf{w} - \mathbf{y}} \mod q\|_2 \\ &= \|2^{\nu_\mathbf{w}} \cdot \overline{\lfloor \bar{\mathbf{w}} \rceil_{\nu_\mathbf{w}} - \lfloor \bar{\mathbf{w}} + c \cdot \boldsymbol{\alpha}_\mathbf{t} - \boldsymbol{\alpha} \rceil_{\nu_\mathbf{w}}} \mod q\|_2 \\ &\leq \|-c \cdot \boldsymbol{\alpha}_\mathbf{t} + \boldsymbol{\alpha} \mod q\|_2 + \sqrt{nk} \cdot 2^{\nu_\mathbf{w}}.\end{aligned}$$

Using Lemma 3.13, we further have $\|-c \cdot \boldsymbol{\alpha}_\mathbf{t} + \boldsymbol{\alpha} \mod q\|_2 \leq \|c \cdot \boldsymbol{\alpha}_\mathbf{t} \mod q\|_2 + \|\boldsymbol{\alpha} \mod q\|_2$. Using the Minkowski inequality and Lemma 3.14, Eq. (2), we have $\|c \cdot \boldsymbol{\alpha}_\mathbf{t} \mod q\|_2 \leq W \cdot \sqrt{nk} \cdot (2^{\nu_\mathbf{t}} - 1)$, where recall $W = \|c\|_1$. Moreover, we have $\|\boldsymbol{\alpha} \mod q\|_2 \leq e^{1/4} \cdot (2W \cdot \sigma_\mathbf{t} + \sigma_\mathbf{w} \cdot \sqrt{\mathsf{rep} \cdot |\mathsf{SS}|}) \cdot \sqrt{nk}$ with overwhelming probability from Lemma 3.2 and Lemma 3.3, where note that we use the fact $\beta_b \in \mathbb{T}$ for the latter. Lastly, with the same argument, we have $\|\mathbf{z} \mod q\|_2 \leq e^{1/4} \cdot (2W \cdot \sigma_\mathbf{t} + \sigma_\mathbf{w} \cdot \sqrt{\mathsf{rep} \cdot |\mathsf{SS}|}) \cdot \sqrt{n\ell}$. Combining all the bounds results in the desired bound. □

6 Security of Our two-round Threshold Signature

In this section, we prove the unforgeability of our two-round threshold signature scheme $\mathsf{TS}_{\mathsf{2\text{-}round}}$. We will rely on the hardness of the AOM-MLWE problem with parameters based on those proposed in Sections 4.4 and 4.5. Before providing the proof of unforgeability, we first give asymptotic parameters for which our scheme is provably secure. A concrete parameter selection along with an efficiency analysis is provided in Section 7.

6.1 Asymptotic Parameters

We will first be explicit on how we establish the parameters for the hardness of the AOM-MLWE problem. We begin by choosing the parameters for which the *selective* AOM-UMLWE problem with an invertible submatrix is hard as in Theorem 4.5. We then use the equivalence between this slightly restricted sel-AOM-UMLWE and the standard sel-AOM-MLWE in Corollary 4.3 (see Section 4.2.3). Lastly, as discussed in Section 4, we assume that AOM-MLWE is as hard as its selective variant. The final step is the only step for which we do not have a supporting security reduction. Concretely, we rely on the following:

1. Let us define $\mathsf{sel\text{-}AOM\text{-}UMLWE}_{q,\ell,k+\ell,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L}_{\mathsf{TS}},B_{\mathcal{L}_{\mathsf{TS}}},\max\{B_{\mathbf{s}},B_{\mathbf{e}}\}}$ with an invertible submatrix, where
   - $Q = \mathsf{rep}\cdot Q_{\mathsf{S}} + 1$,
   - $\mathcal{D}_1 = 2\cdot\mathcal{D}_{\mathbf{t}}$ and $\mathcal{D}_i = \mathcal{D}_{\mathbf{w}}$ for $i \in [2, \mathsf{rep}\cdot Q_{\mathsf{S}}+1]$,
   - $\mathcal{L}_{\mathsf{TS}}$ is the accepted linear combinations defined in Definition 4.8,
   - $B_{\mathbf{s}} = 8e^{1/4}\cdot\left(W^2\cdot\sigma_{\mathbf{t}} + W\cdot\sigma_{\mathbf{w}}\right)\cdot\sqrt{n\ell} + 4B$,
   - $B_{\mathcal{L}_{\mathsf{TS}}} = 4\sqrt{W}$,
   - $B_{\mathbf{e}} = \left(2^{\nu_{\mathbf{w}}+3} + W\cdot 2^{\nu_{\mathbf{t}}+2} + 8e^{1/4}\cdot\left(W^2\cdot\sigma_{\mathbf{t}} + W\cdot\sigma_{\mathbf{w}}\right)\right)\cdot\sqrt{nk} + 4B$.

   By setting the parameters $(q, n, \ell, \mathcal{D}_{\mathbf{t}} = \mathcal{D}_{\sigma_1}, \mathcal{D}_{\mathbf{w}} = \mathcal{D}_{\sigma^*})$ showing up in our two-round threshold signature scheme (c.f., from Table 1) according to hard-param in Definition 4.4 (and Remark 4.10) and setting the set of accepted linear combinations $\mathcal{L}_{\mathsf{TS}}$ according to Definition 4.8, we have $\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}UMLWE}}_{\mathcal{B}_1}(1^\lambda) = \mathsf{negl}(\lambda)$ for any efficient adversary $\mathcal{B}_1$.

2. Let us define $\mathsf{sel\text{-}AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L}_{\mathsf{TS}},B_{\mathcal{L}_{\mathsf{TS}}},B_{\mathbf{s}},B_{\mathbf{e}}}$ with the same parameters as above. Then, from Corollary 4.3, assuming the hardness of sel-AOM-UMLWE with an invertible submatrix, we have $\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}MLWE}}_{\mathcal{B}_2}(1^\lambda) = \mathsf{negl}(\lambda)$ for any efficient adversary $\mathcal{B}_2$ (see Section 4.2.3).

3. Lastly, let us define $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L}_{\mathsf{TS}},B_{\mathcal{L}_{\mathsf{TS}}},B_{\mathbf{s}},B_{\mathbf{e}}}$ with the same parameters as above. Here, assuming that any adaptive adversary $\mathcal{B}$ against the AOM-MLWE problem can perform no better than a selective adversary $\mathcal{B}_2$ against the sel-AOM-MLWE problem defined above, we have $\mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda) = \mathsf{negl}(\lambda)$. This is the assumption our two-round threshold signature is based on.

Candidate Asymptotic Parameters. We give a set of asymptotic parameters that fit the above constraints and the correctness condition in Theorem 5.2. Note that Q_{S} denotes the maximum signature query an adversary can perform.

- $n, \ell, k = \mathsf{poly}(\lambda)$ such that $n \geq \lambda$.
- $(\sigma_{\mathbf{t}}, \sigma_{\mathbf{w}}) = \left(\sqrt{\ell}\cdot\log n, 2W\cdot\sigma_{\mathbf{t}}\cdot n^{1.5}\cdot\sqrt{2Q_{\mathsf{S}}\cdot k}\right)$.
- $\nu_{\mathbf{t}}, \nu_{\mathbf{w}} = O(\log\lambda)$.
- $\mathsf{rep} = \omega(\lambda/\log\lambda)$ for $|\mathbb{T}| \geq 2^\lambda$.
- $\sqrt{\mathsf{rep}}\cdot\sigma_{\mathbf{w}} > 2n\cdot q^{\frac{1}{k+\ell}+\frac{2}{n\ell}}$ and $\nu_{\mathbf{w}} < \log(q) - 1$ for Lemma 3.20, where the lower bound on $\sigma_{\mathbf{w}}$ is subsumed by above.
- $W = \omega(1)$ for $|\mathcal{C}| \geq 2^\lambda$.
- $B = (W\cdot 2^{\nu_{\mathbf{t}}} + 2^{\nu_{\mathbf{w}}})\cdot\sqrt{nk} + e^{1/4}\cdot(2W\cdot\sigma_{\mathbf{t}} + \sigma_{\mathbf{w}}\cdot\sqrt{\mathsf{rep}\cdot N})\cdot\sqrt{n}\cdot(\sqrt{k}+\sqrt{\ell})$.
- q is the smallest prime larger than $(B_{\mathcal{L}_{\mathsf{TS}}} + B_{\mathbf{e}})\cdot\sqrt{n(k-\ell)}\cdot\log^2(n(k-\ell))$ such that $(q, \nu_{\mathbf{t}}, \nu_{\mathbf{w}})$ satisfies the condition in Table 1, where $B_{\mathcal{L}_{\mathsf{TS}}} = 4\sqrt{W}$ and $B_{\mathbf{e}} = (2^{\nu_{\mathbf{w}}+3} + W\cdot 2^{\nu_{\mathbf{t}}+2} + 8e^{1/4}\cdot\left(W^2\cdot\sigma_{\mathbf{t}} + W\cdot\sigma_{\mathbf{w}}\right))\cdot\sqrt{nk} + 4B$.

6.2 Main Theorem

The following is the main theorem establishing the unforgeability of our two-round threshold signature scheme. The statement assumes the asymptotic parameter selections in Section 6.1.

Theorem 6.1. *The two-round threshold signature* $\mathsf{TS}_{\text{2-round}}$ *in Fig. 8 is unforgeable under the* $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q_\mathsf{G}]},\mathcal{L}_\mathsf{TS},B}$ *assumption and the pseudorandomness of* PRF.

Formally, for any N and T with $T \le N$ and an adversary $\mathcal{A}$ against the unforgeability game making at most Q_H, Q_G, and Q_S queries to the random oracles H and G, and the signing oracle, respectively, there exists adversaries $\mathcal{B}$ and $\mathcal{B}'$ against the $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q_\mathsf{G}]},\mathcal{L}_\mathsf{TS},B_{\mathcal{L}_\mathsf{TS}},B_\mathbf{s},B_\mathbf{e}}$ *problem and pseudorandomness of* PRF *such that*

$$\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\text{2-round}},\mathcal{A}}(1^\lambda,N,T) \le \sqrt{Q_\mathsf{RO}\cdot\mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + N^2\cdot\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}'}(1^\lambda) + \frac{Q_\mathsf{S}^2}{2^{n-1}} + \mathsf{negl}(\lambda),$$

where $Q_\mathsf{RO} = Q_\mathsf{H} + 2Q_\mathsf{G} + 2Q_\mathsf{S} + 1$, $\mathsf{Time}(\mathcal{B}) \approx 2\cdot\mathsf{Time}(\mathcal{A})$, *and* $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A})$.

Overview. Before providing the full proof, we provide a brief overview. The proof consists of two parts. The first half consists of carefully crafting a sequence of games so that the reduction can simulate the game using only knowledge of the signing key $\mathsf{sk} = \mathbf{s}$, implicitly defined by the partial signing keys $\mathbf{s}_i$ included in the secret key shares sk_i. From a bird's eye view, this is similar to what was done in [PKM+24], and an intuition of the idea is given in Appendix A. At a lower level, as explained in Section 2, the difference lies in how we generate the masks $\mathbf{m}_i$ and $\mathbf{m}_i^*$. We no longer rely on session unique identifiers $\mathsf{sid} \in \{0,1\}^*$ and standard signatures to explicitly authenticate the signers' views. Instead, we replace sid with $\mathsf{ctnt} := \mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$ and the signature by viewing the masks as an implicit MAC on the "message" ctnt. The reduction then consists of a careful bookkeeping of the signers that have signed with respect to ctnt.

The second half consists of constructing an $\mathsf{AOM\text{-}MLWE}$ adversary $\mathcal{B}$ using the adversary $\mathcal{A}$ against the unforgeability game. $\mathcal{B}$ is given $\mathbf{T} = \mathbf{AS} + \mathbf{E}$ as the problem instance, where $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$, $(\mathbf{s},\mathbf{e}) \xleftarrow{\$} \mathcal{D}_\mathbf{t}^\ell \times \mathcal{D}_\mathbf{t}^k$, $(\widehat{\mathbf{r}}_{i,b}, \widehat{\mathbf{e}}'_{i,b}) \xleftarrow{\$} \mathcal{D}_\mathbf{w}^\ell \times \mathcal{D}_\mathbf{w}^k$ for $(i,b) \in [Q_\mathsf{S}]\times[\mathsf{rep}]$, and $(\mathbf{S},\mathbf{E}) = \big([2\cdot\mathbf{s} \mid \widehat{\mathbf{r}}_{1,1} \mid \widehat{\mathbf{r}}_{1,2} \mid \cdots \mid \widehat{\mathbf{r}}_{Q_\mathsf{S},\mathsf{rep}}], [2\cdot\mathbf{e} \mid \widehat{\mathbf{e}}'_{1,1} \mid \widehat{\mathbf{e}}'_{1,2} \mid \cdots \mid \widehat{\mathbf{e}}'_{Q_\mathsf{S},\mathsf{rep}}]\big) \in \mathcal{R}_q^{\ell\times(\mathsf{rep}\cdot Q_\mathsf{S}+1)} \times \mathcal{R}_q^{\ell\times(\mathsf{rep}\cdot Q_\mathsf{S}+1)}$. It embeds the first column $\mathbf{t}_1$ of $\mathbf{T}$ into the verification key $\mathbf{t}$, and the rest is used to simulate the pre-processing token pp_i of the honest signers. Due to the above modification, whenever the reduction needs to simulate a partial response $\mathbf{z}_i$, they will be of the form $\mathbf{z}_i = c\cdot\mathbf{s} + \sum_{b\in[\mathsf{rep}]}\beta_b\cdot\mathbf{r}_{i,b} + (\text{public vector})$ or $\mathbf{z}_i = \sum_{b\in[\mathsf{rep}]}\beta_b\cdot\mathbf{r}_{i,b} + (\text{public vector})$. Thus, it can simulate them by querying the MLWE solving oracle on the respective coefficients (i.e., linear combination). The bulk of the proof consists of checking that all the queries fall into the accepted linear combinations $\mathcal{L}_\mathsf{TS}$, required by the winning condition of the $\mathsf{AOM\text{-}MLWE}$ problem. This check is non-trivialized by the fact that the adversary is rewound to extract from the forgery an MLWE solution with respect to the verification key $\mathbf{t}$.

Proof. Let $\mathcal{A}$ be an adversary against the unforgeability game. We consider a sequence of games where the first hybrid is the original game and the last is a game that can be reduced from the $\mathsf{AOM\text{-}MLWE}$ problem. We relate the advantage of $\mathcal{A}$ for each adjacent games, where ϵ_i denotes the advantage of $\mathcal{A}$ in Game_i.

Game_1**:** This is the real unforgeability game. Formally, this is depicted in Fig. 9. By definition, we have

$$\epsilon_1 := \mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\text{2-round}},\mathcal{A}}(1^\lambda,N,T).$$

Game_2**:** In this game, the challenger modifies how it maintains the random oracles G and H. This is depicted in Fig. 10. Specifically, when G is queried on a pair $(\mathsf{vk},\mathsf{ctnt})$ such that ctnt correctly parses as $\mathsf{SS}\|\mathsf{M}\|(\vec{\mathbf{w}}_j)_{j\in[\mathsf{SS}]}$, it computes the aggregated commitment $\mathbf{w}$ and sets $\mathsf{H}(\mathsf{vk},\mathsf{M},\mathbf{w}) \leftarrow c$ for a random $c \xleftarrow{\$} \mathcal{C}$ if it hasn't been set yet. Since the time on which H is set cannot be detected by $\mathcal{A}$, the two games are identical. Thus, we have

$$\epsilon_2 = \epsilon_1.$$

Looking ahead, this modification is useful when invoking the forking lemma to extract an MLWE solution from the forgery (see Lemma 6.2 for the detail).

$\mathsf{Game}_1 := \mathsf{Game}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\text{2-round}},\mathcal{A}}(1^\lambda, N, T)$

$\mathsf{Q_M} := \emptyset, \mathsf{Q_H}[\cdot] := \bot, \mathsf{Q_G}[\cdot] := \bot$
$\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$
$(\mathsf{CS}, \mathsf{st}_\mathcal{A}) \xleftarrow{\$} \mathcal{A}^{\mathsf{H},\mathsf{G}}(\mathbf{A}, N, T)$
req $[\![\mathsf{CS} \subseteq [N]]\!] \wedge [\![|\mathsf{CS}| \le T-1]\!]$
$\mathsf{HS} := [N] \backslash \mathsf{CS}$
for $i \in \mathsf{HS}$ **do** $\mathsf{st}_i := \emptyset$
$(\mathbf{s}, \mathbf{e}) \xleftarrow{\$} \mathcal{D}_{\mathbf{t}}^\ell \times \mathcal{D}_{\mathbf{t}}^k$
$\mathbf{t} := \lfloor 2 \cdot (\mathbf{A}\mathbf{s} + \mathbf{e}) \rceil_{\nu_\mathbf{t}} \in \mathcal{R}_{q_{\nu_\mathbf{t}}}^k$
for $(i,j) \in [N] \times [N]$ **do**
  $\mathsf{seed}_{i,j} \xleftarrow{\$} \{0,1\}^\lambda$
$\vec{P} \xleftarrow{\$} \mathcal{R}_q^\ell[X]$ with $\deg(\vec{P}) = T-1, \vec{P}(0) = 2 \cdot \mathbf{s}$
$(\mathbf{s}_i)_{i\in[N]} := (\vec{P}(i))_{i\in[N]}$
$\mathsf{vk} := (\mathsf{tspar}, \mathbf{t})$
$(\mathsf{sk}_i)_{i\in[N]} := \Big(\big(\mathbf{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[N]}\big)\Big)_{i\in[N]}$
$(\mathsf{sig}^*, \mathsf{M}^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{TS.PP}}, \mathcal{O}_{\mathsf{TS.Sign}}, \mathsf{H}, \mathsf{G}}(\mathsf{vk}, (\mathsf{sk}_i)_{i\in\mathsf{CS}}, \mathsf{st}_\mathcal{A})$
if $[\![\mathsf{M}^* \in \mathsf{Q_M}]\!]$ **then return** 0
return $\mathsf{TS.Verify}(\mathsf{tspar}, \mathsf{vk}, \mathsf{M}^*, \mathsf{sig}^*)$

$\mathcal{O}_{\mathsf{TS.PP}}(i)$

req $[\![i \in \mathsf{HS}]\!]$
for $b \in [\mathsf{rep}]$ **do**
  $(\mathbf{r}_{i,b}, \mathbf{e}'_{i,b}) \xleftarrow{\$} \mathcal{D}_\mathbf{w}^\ell \times \mathcal{D}_\mathbf{w}^k$
  $\mathbf{w}_{i,b} := \mathbf{A}\mathbf{r}_{i,b} + \mathbf{e}'_{i,b} \in \mathcal{R}_q^k$
$\vec{\mathbf{w}}_i := [\mathbf{w}_{i,1} \mid \cdots \mid \mathbf{w}_{i,\mathsf{rep}}]$
$\mathsf{pp}_i := \vec{\mathbf{w}}_i$
$\mathsf{st}_i \leftarrow \mathsf{st}_i \cup \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$
return pp_i

$\mathsf{H}(\mathsf{vk}, \mathsf{M}, \mathbf{w})$

if $[\![\mathsf{Q_H}[\mathsf{vk}, \mathsf{M}, \mathbf{w}] = \bot]\!]$ **then**
  $c \xleftarrow{\$} \mathcal{C}$
  $\mathsf{Q_H}[\mathsf{vk}, \mathsf{M}, \mathbf{w}] \leftarrow c$
return $\mathsf{Q_H}[\mathsf{vk}, \mathsf{M}, \mathbf{w}]$

$\mathsf{G}(\mathsf{vk}, \mathsf{ctnt})$

if $[\![\mathsf{Q_G}[\mathsf{vk}, \mathsf{ctnt}] = \bot]\!]$ **then**
  $(\beta_b)_{b\in[2,\mathsf{rep}]} \xleftarrow{\$} \mathbb{T}^{\mathsf{rep}-1}$
  $\mathsf{Q_G}[\mathsf{vk}, \mathsf{ctnt}] \leftarrow (1, (\beta_b)_{b\in[2,\mathsf{rep}]})$
return $\mathsf{Q_G}[\mathsf{vk}, \mathsf{ctnt}]$

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}})$

req $[\![\mathsf{SS} \subseteq [N]]\!] \wedge [\![i \in \mathsf{HS} \cap \mathsf{SS}]\!] \wedge [\![(\mathsf{pp}_i, \cdot) \in \mathsf{st}_i]\!]$
parse $(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}\backslash\{i\}} \leftarrow (\mathsf{pp}_j)_{j\in\mathsf{SS}}$
pick $(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})$ **from** st_i with $\mathsf{pp}_i = \vec{\mathbf{w}}_i$
$\mathsf{ctnt} := \mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$
$(\beta_b)_{b\in[\mathsf{rep}]} := \mathsf{G}(\mathsf{vk}, \mathsf{ctnt})$
for $j \in \mathsf{SS}$ **do**
  parse $[\mathbf{w}_{j,1} \mid \cdots \mid \mathbf{w}_{j,\mathsf{rep}}] \leftarrow \vec{\mathbf{w}}_j$
  $\mathbf{w}_j := \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{w}_{j,b} \in \mathcal{R}_q^k$
$\mathbf{w} := \left\lfloor \sum_{j\in\mathsf{SS}} \mathbf{w}_j \right\rceil_{\nu_\mathbf{w}} \in \mathcal{R}_{q_{\nu_\mathbf{w}}}^k$
$c := \mathsf{H}(\mathsf{vk}, \mathsf{M}, \mathbf{w})$
$\mathbf{m}_i := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt}) \in \mathcal{R}_q^\ell$
$\mathbf{m}_i^* := \sum_{j\in\mathsf{SS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt}) \in \mathcal{R}_q^\ell$
$\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}_i^* - \mathbf{m}_i \in \mathcal{R}_q^\ell$
$\mathsf{st}_i \leftarrow \mathsf{st}_i \backslash \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$
$\mathsf{Q_M} := \mathsf{Q_M} \cup \{\mathsf{M}\}$
return $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Figure 9: The first game, identical to the real unforgeability game.

Game_3**:** This game, depicted in Fig. 11, is merely a syntactical modification to aid readability. In particular, we divide the signer set SS into the set of honest users $\mathsf{sHS} := \mathsf{SS} \cap \mathsf{HS}$ and corrupt users $\mathsf{sCS} := \mathsf{SS} \cap \mathsf{CS}$, and define intermediate masking terms $(\mathbf{m}_{i,\mathsf{sHS}}, \mathbf{m}^*_{\mathsf{sHS},i}, \mathbf{m}_{i,\mathsf{sCS}}, \mathbf{m}^*_{\mathsf{sCS},i})$. The advantage remains the same and we have

$$\epsilon_3 = \epsilon_2.$$

Game_2:

$\mathsf{G}(\mathsf{vk}, \mathsf{ctnt})$

1 : **if** $[\![\mathsf{Q}_\mathsf{G}[\mathsf{vk}, \mathsf{ctnt}] = \bot]\!]$ **then**
2 : $\quad (\beta_b)_{b\in[2,\mathsf{rep}]} \xleftarrow{\$} \mathbb{T}^{\mathsf{rep}-1}$
3 : $\quad \mathsf{Q}_\mathsf{G}[\mathsf{vk}, \mathsf{ctnt}] := (1, (\beta_b)_{b\in[2,\mathsf{rep}]})$
4 : $\quad$ **if** $[\![\mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}} \leftarrow \mathsf{ctnt}$ correctly parses$]\!]$ **then**
5 : $\quad\quad$ **for** $j \in \mathsf{SS}$ **do**
6 : $\quad\quad\quad$ **parse** $[\mathbf{w}_{j,1} \mid \cdots \mid \mathbf{w}_{j,\mathsf{rep}}] \leftarrow \vec{\mathbf{w}}_j$
7 : $\quad\quad\quad \mathbf{w}_j := \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{w}_{j,b} \in \mathcal{R}_q^\ell$
8 : $\quad\quad \mathbf{w} := \left\lfloor \sum_{j\in\mathsf{SS}} \mathbf{w}_j \right\rceil_{\nu_\mathbf{w}} \in \mathcal{R}_{q\nu_\mathbf{w}}^k$
9 : $\quad\quad$ **if** $[\![\mathsf{Q}_\mathsf{H}[\mathsf{vk}, \mathsf{M}, \mathbf{w}] \neq \bot]\!]$ **then**
10 : $\quad\quad\quad c \xleftarrow{\$} \mathcal{C},\ \mathsf{Q}_\mathsf{H}[\mathsf{vk}, \mathsf{M}, \mathbf{w}] \leftarrow c$
11 : **return** $\mathsf{Q}_\mathsf{G}[\mathsf{vk}, \mathsf{ctnt}]$

Figure 10: The second game. The only difference between Game_1 is how the random oracle G is simulated. This is highlighted in blue. In above, we assume ctnt can be checked to have a correct encoding: a signer set $\mathsf{SS} \in [N]$, a message, and $|\mathsf{SS}|(= T)$ commitments.

Game_4**:** In this game, the challenger samples a random mask $\mathbf{m}_{i,j} \xleftarrow{\$} \mathcal{R}_q^\ell$ instead of computing $\mathbf{m}_{i,j} = \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$ when $i, j \in \mathsf{HS}$. This is depicted in Fig. 11. In particular, the challenger prepares an empty list $\mathsf{Rand}[\cdot]$ at the beginning of the game and assigns $\mathsf{Rand}[\mathsf{ctnt}, i, j]$ random masks. Since $\mathsf{seed}_{i,j}$ is never revealed to the adversary $\mathcal{A}$ when $i, j \in \mathsf{HS}$, we can go over at most $(N-1)^2$ hybrids using the pseudorandomness of the PRF to establish indistinguishability of the two games. That is, there exists an adversary $\mathcal{B}'$ against the pseudorandomness of PRF such that $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A})$ and

$$|\epsilon_4 - \epsilon_3| \leq N^2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}'}(1^\lambda).$$

Game_5**:** In this game, the challenger adds an abort condition when queried the signing oracle $\mathcal{O}_{\mathsf{TS.Sign}}$. This is depicted in Fig. 12. The challenger first prepares an empty list $\mathsf{Signed}[\cdot]$ at the beginning of the game. When the adversary $\mathcal{A}$ queries the signing oracle $\mathcal{O}_{\mathsf{TS.Sign}}$ on user $i \in \mathsf{SS}$, it sets $\mathsf{ctnt} := \mathsf{SS}||\mathsf{M}||(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$ and checks if $\mathsf{Signed}[\mathsf{ctnt}, i] = \bot$, that is, the challenger checks whether user i has already signed with ctnt. If so, it aborts the game and otherwise, it proceeds identically to Game_4 and the challenger updates $\mathsf{Signed}[\mathsf{ctnt}, i] \leftarrow \top$.

Let us bound the probability that an honest user i signs the same ctnt more than once. Notice that ctnt includes $\vec{\mathbf{w}}_i$; the vector of commitments that user i generated in the pre-processing phase. By construction, $\vec{\mathbf{w}}_i$ is stored in st_i, which is discarded once user i signs with ctnt. In particular, for an honest user i to have signed on the same ctnt more than twice, then it must have generated the same vector of commitments $\vec{\mathbf{w}}_i$ in the pre-processing phase. Since these commitments are generated honestly, the probability of such an event occurring can be bounded by 2^{-n+1} with overwhelming

Game_3:

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}})$

// Identical to Lines 1 to 10 of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1

11 : $\mathbf{m}_{i,\mathsf{sCS}} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

12 : $\mathbf{m}^*_{\mathsf{sCS},i} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

13 : $\mathbf{m}_{i,\mathsf{sHS}} := \sum_{j\in\mathsf{sHS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

14 : $\mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

15 : $\mathbf{m}_i := \mathbf{m}_{i,\mathsf{sHS}} + \mathbf{m}_{i,\mathsf{sCS}} \in \mathcal{R}_q^\ell$

16 : $\mathbf{m}^*_i := \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} \in \mathcal{R}_q^\ell$

17 : $\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_i - \mathbf{m}_i \in \mathcal{R}_q^\ell$

18 : $\mathsf{st}_i \leftarrow \mathsf{st}_i \backslash \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$

19 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$

20 : **return** $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Game_4:

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}})$

// Identical to Lines 1 to 10 of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1

11 : $\mathbf{m}_{i,\mathsf{sCS}} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

12 : $\mathbf{m}^*_{\mathsf{sCS},i} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

13 : **for** $j \in \mathsf{sHS}$ **do**

14 : **if** $[\![\mathsf{Rand}[\mathsf{ctnt}, i, j] = \bot]\!]$ **then**

15 : $\mathbf{m}_{i,j} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Rand}[\mathsf{ctnt}, i, j] \leftarrow \mathbf{m}_{i,j}$

16 : **if** $[\![\mathsf{Rand}[\mathsf{ctnt}, j, i] = \bot]\!]$ **then**

17 : $\mathbf{m}_{j,i} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Rand}[\mathsf{ctnt}, j, i] \leftarrow \mathbf{m}_{j,i}$

18 : $\mathbf{m}_{i,\mathsf{sHS}} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, i, j]$

19 : $\mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, j, i]$

20 : $\mathbf{m}_i := \mathbf{m}_{i,\mathsf{sHS}} + \mathbf{m}_{i,\mathsf{sCS}} \in \mathcal{R}_q^\ell$

21 : $\mathbf{m}^*_i := \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} \in \mathcal{R}_q^\ell$

22 : $\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_i - \mathbf{m}_i \in \mathcal{R}_q^\ell$

23 : $\mathsf{st}_i \leftarrow \mathsf{st}_i \backslash \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$

24 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$

25 : **return** $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Figure 11: The third and fourth games. The changes between the previous game are highlighted in blue. For readability, we omit the lines that are identical to those of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1. We assume Game_4 initializes an empty list $\mathsf{Rand}[\cdot] := \bot$ at the beginning of the game. Lastly, $\mathsf{sHS} := \mathsf{SS} \cap \mathsf{HS}$ and $\mathsf{sCS} := \mathsf{SS} \cap \mathsf{CS}$.

Game_5:

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}})$

// Identical to Lines 1 to 10 of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1

11 : $\mathbf{m}_{i,\mathsf{sCS}} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

12 : $\mathbf{m}^*_{\mathsf{sCS},i} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

13 : **abort if** $[\![\mathsf{Signed}[\mathsf{ctnt}, i] = \top]\!]$

14 : **for** $j \in \mathsf{sHS}$ **do**

15 : **if** $[\![\mathsf{Rand}[\mathsf{ctnt}, i, j] = \bot]\!]$ **then**

16 : $\mathbf{m}_{i,j} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Rand}[\mathsf{ctnt}, i, j] \leftarrow \mathbf{m}_{i,j}$

17 : **if** $[\![\mathsf{Rand}[\mathsf{ctnt}, j, i] = \bot]\!]$ **then**

18 : $\mathbf{m}_{j,i} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Rand}[\mathsf{ctnt}, j, i] \leftarrow \mathbf{m}_{j,i}$

19 : $\mathbf{m}_{i,\mathsf{sHS}} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, i, j]$

20 : $\mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, j, i]$

21 : $\mathsf{Signed}[\mathsf{ctnt}, i] \leftarrow \top$

22 : $\mathbf{m}_i := \mathbf{m}_{i,\mathsf{sHS}} + \mathbf{m}_{i,\mathsf{sCS}} \in \mathcal{R}_q^\ell$

23 : $\mathbf{m}^*_i := \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} \in \mathcal{R}_q^\ell$

24 : $\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_i - \mathbf{m}_i \in \mathcal{R}_q^\ell$

25 : $\mathsf{st}_i \leftarrow \mathsf{st}_i \backslash \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$

26 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$

27 : **return** $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Game_6:

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, i, (\mathsf{pp}_j)_{j\in\mathsf{SS}\backslash\{i\}})$

// Identical to Lines 1 to 10 of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1

11 : $\mathbf{m}_{i,\mathsf{sCS}} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

12 : $\mathbf{m}^*_{\mathsf{sCS},i} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

13 : **abort if** $[\![\mathsf{Signed}[\mathsf{ctnt}, i] = \top]\!]$

14 : $\mathbf{m}_{i,\mathsf{sHS}} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m} \leftarrow \mathbf{m}_{i,\mathsf{sHS}}$

15 : **if** $[\![\forall j \in \mathsf{sHS}\backslash\{i\}, \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m}^* \neq \bot]\!]$ **then**

16 : $\mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m} - \sum_{j\in\mathsf{sHS}\backslash\{i\}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m}^*$

17 : **else**

18 : $\mathbf{m}^*_{\mathsf{sHS},i} \xleftarrow{\$} \mathcal{R}_q^\ell$

19 : $\mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m}^* \leftarrow \mathbf{m}^*_{\mathsf{sHS},i}$

20 : $\mathsf{Signed}[\mathsf{ctnt}, i] \leftarrow \top$

21 : $\mathbf{m}_i := \mathbf{m}_{i,\mathsf{sHS}} + \mathbf{m}_{i,\mathsf{sCS}} \in \mathcal{R}_q^\ell$

22 : $\mathbf{m}^*_i := \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} \in \mathcal{R}_q^\ell$

23 : $\mathbf{z}_i := c \cdot L_{\mathsf{SS},i} \cdot \mathbf{s}_i + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_i - \mathbf{m}_i \in \mathcal{R}_q^\ell$

24 : $\mathsf{st}_i \leftarrow \mathsf{st}_i \backslash \{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$

25 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$

26 : **return** $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Figure 12: The fifth and sixth games. The changes between the previous game are highlighted in blue. We assume both games initialize an empty list $\mathsf{Signed}[\cdot] := \bot$ and an empty object $\mathsf{Mask}[\cdot] := \bot$ at the beginning of the game. $\mathsf{Mask}[\cdot]$ contains two fields: $\mathsf{Mask}[\cdot].\mathbf{m}$ to store row masks and $\mathsf{Mask}[\cdot].\mathbf{m}^*$ to store column masks.

probability using Lemma 3.20. Thus, we have

$$|\epsilon_5 - \epsilon_4| \leq \frac{Q_\mathsf{S}^2}{2^{n-1}} + \mathsf{negl}(\lambda).$$

Game_6: In this game, the challenger changes how it generates the intermediate masking terms of the honest users $(\mathbf{m}_{i,\mathsf{sHS}}, \mathbf{m}^*_{\mathsf{sHS},i})$. This is depicted in Fig. 12. Throughout the proof, we will call $\mathbf{m}_{i,\mathsf{sHS}}$ and $\mathbf{m}^*_{\mathsf{sHS},i}$ the *row* and *column* masks, respectively. Viewing $(\mathbf{m}_{i,j})_{i,j\in\mathsf{sHS}}$ from Game_5 as a matrix, $\mathbf{m}_{i,\mathsf{sHS}}$ and $\mathbf{m}^*_{\mathsf{sHS},i}$ are indeed sum of the row and column entries, respectively. Concretely, in this game, the challenger prepares a new object $\mathsf{Mask}[\cdot]$ containing two fields: $\mathsf{Mask}[\cdot].\mathbf{m}$ to store row masks and $\mathsf{Mask}[\cdot].\mathbf{m}^*$ to store column masks. The challenger directly generates the row and column masks $\mathbf{m}_{i,\mathsf{sHS}}$ and $\mathbf{m}^*_{\mathsf{sHS},i}$, rather than generating the individual masks $(\mathbf{m}_{i,j})_{i,j\in\mathsf{sHS}}$, and stores them in $\mathsf{Mask}[\cdot].\mathbf{m}$

and $\mathsf{Mask}[\cdot].\mathbf{m}^*$.

We prove the view of the two games are identically distributed to the adversary $\mathcal{A}$. Assume $\mathcal{A}$ queries the signing oracle on user $i \in \mathsf{HS}$ with signer set SS. We first focus on the row masks. Since user i has never signed with $\mathsf{ctnt} = \mathsf{SS}\|\mathsf{M}\|(\mathbf{w}_j)_{j\in\mathsf{SS}}$ (i.e., $\mathsf{Signed}[\mathsf{ctnt}, i] = \bot$), we have $\mathsf{Rand}[\mathsf{ctnt}, i, i] = \bot$ in Game_5. This implies $\mathbf{m}_{i,i} \xleftarrow{\$} \mathcal{R}_q^\ell$, and in particular, $\mathbf{m}_{i,\mathsf{sHS}} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, i, j]$ is distributed uniformly random over $\mathcal{R}_q^\ell$ in Game_5. On the other hand, $\mathbf{m}_{i,\mathsf{sHS}} := \mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m}$ in Game_6 is also distributed uniformly random over $\mathcal{R}_q^\ell$ since $\mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m} = \bot$ if $\mathsf{Signed}[\mathsf{ctnt}, i] = \bot$. Therefore, the view of $\mathcal{A}$ remains identical in both games.

Next, we look at the column masks. We first assume user $i \in \mathsf{sHS}$ is not the last user to sign with ctnt. In this case, there exists at least one $j \in \mathsf{sHS}$ distinct from i for which $\mathsf{Rand}[\mathsf{ctnt}, j, i]$ is not set yet. This implies $\mathbf{m}_{j,i} \xleftarrow{\$} \mathcal{R}_q^\ell$, and in particular, $\mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{Rand}[\mathsf{ctnt}, j, i]$ is distributed uniformly random over $\mathcal{R}_q^\ell$ in Game_5. Similarly to the previous argument, $\mathbf{m}^*_{\mathsf{sHS},i} := \mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m}^*$ in Game_6 is also distributed uniformly random over $\mathcal{R}_q^\ell$, and thus, the view of $\mathcal{A}$ remains identical in both games.

Lastly, let us assume user $i \in \mathsf{sHS}$ is the last user to sign with ctnt. That is, $\mathsf{Signed}[\mathsf{ctnt}, j] = \top$ for all $j \in \mathsf{sHS}\backslash\{i\}$.[12] In Game_5, this implies that everything except $\mathsf{Rand}[\mathsf{ctnt}, i, i]$ is already set. Namely, after $\mathbf{m}_{i,i}$ is sampled, the set $(\mathbf{m}_{i,j})_{i,j\in\mathsf{sHS}}$ is fully determined. Moreover, by construction, we have

$$\sum_{j\in\mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} = \sum_{j\in\mathsf{sHS}} \mathbf{m}^*_{\mathsf{sHS},j}.$$

Combining the arguments, the column mask $\mathbf{m}^*_{\mathsf{sHS},i}$ of the final user is uniquely defined as

$$\mathbf{m}^*_{\mathsf{sHS},i} = \sum_{j\in\mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} - \sum_{j\in\mathsf{sHS}\backslash\{i\}} \mathbf{m}^*_{\mathsf{sHS},j}.$$

This is identical to how $\mathbf{m}^*_{\mathsf{sHS},i}$ is set in Game_6. Therefore, the view of $\mathcal{A}$ remains identical in both games. We conclude that,

$$\epsilon_6 = \epsilon_5.$$

Game_7**:** In this game, the challenger modifies how it computes the responses $\mathbf{z}_i$. This is depicted in Fig. 13. When the adversary $\mathcal{A}$ queries the signing oracle on user $i \in \mathsf{sHS}$, the challenger constructs $\mathsf{ctnt} := \mathsf{SS}\|\mathsf{M}\|(\vec{\mathbf{w}}_j)_{j\in\mathsf{sHS}}$ and checks whether user i is the last user in sHS to sign with ctnt. If not, the challenger removes the partial secret key $\mathsf{sk}_i = \mathbf{s}_i$ from the response $\mathbf{z}_i$. Otherwise, if user i is the last user, then it uses $2 \cdot \mathbf{s} - \sum_{j\in\mathsf{sCS}} L_{\mathsf{SS},j} \cdot \mathbf{s}_j$ in place of the partial secret key $\mathbf{s}_i$ to generate the response $\mathbf{z}_i$.

We show the view of $\mathcal{A}$ remains identical to the previous game. The key observation is that up until the last user in sHS, denoted as i^*, signs with ctnt, the row and column masks $(\mathbf{m}_{j,\mathsf{sHS}}, \mathbf{m}^*_{\mathsf{sHS},j})_{j\in\mathsf{sHS}\backslash\{i^*\}}$ are independently and uniformly distributed over $\mathcal{R}_q^\ell$. Moreover, until i^* signs with ctnt, all the column masks $(\mathbf{m}^*_{\mathsf{sHS},j})_{j\in\mathsf{sHS}\backslash\{i^*\}}$ remain information theoretically hidden from $\mathcal{A}$.

With this observation, we can equally define the challenger of Game_7 to first sample $\widetilde{\mathbf{m}}^*_{\mathsf{sHS},j} \xleftarrow{\$} \mathcal{R}_q^\ell$ and set the column mask as $\mathbf{m}^*_{\mathsf{sHS},j} := c \cdot L_{\mathsf{SS},j} \cdot \mathbf{s}_j + \widetilde{\mathbf{m}}^*_{\mathsf{sHS},j}$ for all users $j \in \mathsf{sHS}\backslash\{i^*\}$. This induces the same distribution as simply sampling $\mathbf{m}^*_{\mathsf{sHS},j} \xleftarrow{\$} \mathcal{R}_q^\ell$. Then, we can rewrite the response $\mathbf{z}_j$ as

$$\mathbf{z}_j := c \cdot L_{\mathsf{SS},j} \cdot \mathbf{s}_j + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{j,b} + \widetilde{\mathbf{m}}^*_{\mathsf{sHS},j} + \mathbf{m}^*_{\mathsf{sCS},j} - \mathbf{m}_i.$$

[12] Note that the adversary $\mathcal{A}$ may never invoke the last user. For instance, $\mathcal{A}$ can query user $i \in \mathsf{sHS}$ with $\mathsf{ctnt}' := \mathsf{SS}\|\mathsf{M}'\|(\vec{\mathbf{w}}_j)_{j\in\mathsf{SS}}$, in which case $\mathsf{Signed}[\mathsf{ctnt}, i]$ will never be set $\top$ due to the argument we made in Game_4: user i never samples the same $\vec{\mathbf{w}}_i$.

Game_7:

$\mathcal{O}_{\mathsf{TS.Sign}}(\mathsf{SS}, \mathsf{M}, (\mathsf{pp}_j)_{j\in\mathsf{SS}}, i)$

// Identical to Lines 1 to 10 of $\mathcal{O}_{\mathsf{TS.Sign}}$ in Game_1

11 : $\mathbf{m}_{i,\mathsf{sCS}} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{ctnt})$

12 : $\mathbf{m}^*_{\mathsf{sCS},i} := \sum_{j\in\mathsf{sCS}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{ctnt})$

13 : **abort if** $[\![\mathsf{Signed}[\mathsf{ctnt}, i] = \top]\!]$

14 : $\mathbf{m}_{i,\mathsf{sHS}} \xleftarrow{\$} \mathcal{R}_q^\ell,\ \mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m} \leftarrow \mathbf{m}_{i,\mathsf{sHS}}$

15 : $\mathbf{m}_i := \mathbf{m}_{i,\mathsf{sHS}} + \mathbf{m}_{i,\mathsf{sCS}}$

16 : **if** $[\![\forall j \in \mathsf{sHS}\backslash\{i\}, \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m}^* \neq \bot]\!]$ **then**

17 : $\quad \mathbf{m}^*_{\mathsf{sHS},i} := \sum_{j\in\mathsf{sHS}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m} - \sum_{j\in\mathsf{sHS}\backslash\{i\}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m}^*$

18 : $\quad \mathbf{z}_i := 2\cdot c\cdot \mathbf{s} - c\sum_{j\in\mathsf{sCS}} L_{\mathsf{SS},j}\cdot \mathbf{s}_j + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} - \mathbf{m}_i \in \mathcal{R}_q^\ell$

19 : **else**

20 : $\quad \mathbf{m}^*_{\mathsf{sHS},i} \xleftarrow{\$} \mathcal{R}_q^\ell$

21 : $\quad \mathbf{z}_i := \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i,b} + \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} - \mathbf{m}_i \in \mathcal{R}_q^\ell$

22 : $\mathsf{Mask}[\mathsf{ctnt}, i].\mathbf{m}^* \leftarrow \mathbf{m}^*_{\mathsf{sHS},i}$

23 : $\mathsf{Signed}[\mathsf{ctnt}, i] \leftarrow \top$

24 : $\mathsf{st}_i \leftarrow \mathsf{st}_i\backslash\{(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b\in[\mathsf{rep}]})\}$

25 : $\mathsf{Q}_\mathsf{M} := \mathsf{Q}_\mathsf{M} \cup \{\mathsf{M}\}$

26 : **return** $\widehat{\mathsf{sig}}_i := (\mathbf{w}_i, \mathbf{z}_i)$

Figure 13: The seventh game. The changes between the previous game are highlighted in blue.

Since $\widetilde{\mathbf{m}}^*_{\mathsf{sHS},j}$ is distributed identically to the column mask of user j sampled in Game_6, the response $\mathbf{z}_j$ for $j \in \mathsf{sHS}\backslash\{i^*\}$ is identically distributed to Game_6.

It remains to analyze the response $\mathbf{z}_{i^*}$ for the last user i^*. First, notice that the column mask $\mathbf{m}^*_{\mathsf{sHS},i}$ of user i^* can be rewritten as

$$\begin{aligned}
\mathbf{m}^*_{\mathsf{sHS},i} &= \sum_{j\in\mathsf{sHS}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m} - \sum_{j\in\mathsf{sHS}\backslash\{i^*\}} \mathsf{Mask}[\mathsf{ctnt}, j].\mathbf{m}^* \\
&= \sum_{j\in\mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} - \sum_{j\in\mathsf{sHS}\backslash\{i^*\}} \mathbf{m}^*_{\mathsf{sHS},j} \\
&= \sum_{j\in\mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} - \sum_{j\in\mathsf{sHS}\backslash\{i^*\}} \big(c\cdot L_{\mathsf{SS},j}\cdot \mathbf{s}_j + \widetilde{\mathbf{m}}^*_{\mathsf{sHS},j}\big).
\end{aligned}$$

Plugging this into $\mathbf{z}_{i^*}$, we have

$$\mathbf{z}_{i^*} = 2\cdot c\cdot \mathbf{s} - c\sum_{j\in\mathsf{sCS}} L_{\mathsf{SS},j}\cdot \mathbf{s}_j + \sum_{b\in[\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i^*,b} + \mathbf{m}^*_{\mathsf{sHS},i^*} + \mathbf{m}^*_{\mathsf{sCS},i^*} - \mathbf{m}_{i^*}$$

$$= c \cdot \left(2 \cdot \mathbf{s} - \sum_{j \in \mathsf{sCS} \cup \mathsf{sHS} \setminus \{i^*\}} L_{\mathsf{SS},j} \cdot \mathbf{s}_j\right) + \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i^*,b} + \sum_{j \in \mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} - \sum_{j \in \mathsf{sHS} \setminus \{i^*\}} \widetilde{\mathbf{m}}^*_{\mathsf{sHS},j} + \mathbf{m}^*_{\mathsf{sCS},i^*} - \mathbf{m}_{i^*}$$
$$= c \cdot L_{\mathsf{SS},i^*} \cdot \mathbf{s}_{i^*} + \sum_{b \in [\mathsf{rep}]} \beta_b \cdot \mathbf{r}_{i^*,b} + \sum_{j \in \mathsf{sHS}} \mathbf{m}_{j,\mathsf{sHS}} - \sum_{j \in \mathsf{sHS} \setminus \{i^*\}} \widetilde{\mathbf{m}}^*_{\mathsf{sHS},j} + \mathbf{m}^*_{\mathsf{sCS},i^*} - \mathbf{m}_{i^*},$$

where the second equality comes from the correctness of the linear Shamir secret sharing scheme. It is easy to check that this is exactly how $\mathbf{z}_{i^*}$ is generated in Game_6, where recall we define $\widetilde{\mathbf{m}}^*_{\mathsf{sHS},j}$ for $j \in \mathsf{sHS} \setminus \{i^*\}$ as the column masks of user j sampled in Game_6.

Combining all the arguments, the response $\mathbf{z}_j$ for all users $j \in \mathsf{sHS}$ are distributed identically in both games. Thus, we have

$$\epsilon_7 = \epsilon_6.$$

Game_8**:** In this final game, the challenger modifies how it generates the commitment $\vec{\mathbf{w}}_i$ in the pre-processing oracle $\mathcal{O}_{\mathsf{TS.PP}}$ and how it generates the partial secret keys $(s_i)_{i \in [N]}$. This is depicted in Fig. 14. The challenger first prepares an empty list $\mathsf{Com}[\cdot]$ and a counter $\mathsf{ctr}_{\vec{\mathbf{w}}} := 1$ at the beginning of the game. It then only generates secret shares for the corrupted signers $(\mathbf{s}_i)_{i \in \mathsf{CS}}$ by uniformly sampling $\mathbf{s}_i \xleftarrow{\$} \mathcal{R}_q^\ell$. Since $\mathbf{s}_i$ of the linear Shamir secret sharing scheme is uniformly distributed over $\mathcal{R}_q^\ell$, the view of $\mathcal{A}$ remains identical to the previous game. Furthermore, it generates $\vec{\widehat{\mathbf{w}}}_i$ and $(\widehat{\mathbf{r}}_{i,b})_{b \in [\mathsf{rep}]}$ for all $i \in [Q_{\mathsf{S}}]$ and stores them in $\mathsf{Com}[\cdot]$ at the beginning of the game. When $\mathcal{A}$ queries the pre-processing oracle $\mathcal{O}_{\mathsf{TS.PP}}$ for the $\mathsf{ctr}_{\vec{\mathbf{w}}}$-th time, it uses $(\vec{\mathbf{w}}_i, (\mathbf{r}_{i,b})_{b \in [\mathsf{rep}]}) := \mathsf{Com}[\mathsf{ctr}_{\vec{\mathbf{w}}}]$ and increments $\mathsf{ctr}_{\vec{\mathbf{w}}}$. Since the timing on which $\vec{\mathbf{w}}_i$ is generated by the challenger is unnoticeable from $\mathcal{A}$, the view of $\mathcal{A}$ remains identical to the previous game. We thus have

$$\epsilon_8 = \epsilon_7.$$

Using Lemma 6.2, which we will prove in Section 6.3, there exists an adversary $\mathcal{B}$ against the $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}},B_{\mathbf{s}},B_{\mathbf{e}}}$ problem that internally runs $\mathcal{A}$ against the security game in Game_8 such that

$$\epsilon_8 \leq \sqrt{Q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + \mathsf{negl}(\lambda),$$

where $Q_{\mathsf{RO}} = Q_{\mathsf{H}} + 2Q_{\mathsf{G}} + 2Q_{\mathsf{S}} + 1$ and $Q = \mathsf{rep} \cdot Q_{\mathsf{S}} + 1$. Moreover, we have $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$. Collecting the bounds, we obtain

$$\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\mathsf{2\text{-}round}},\mathcal{A}}(1^\lambda, N, T) \leq \sqrt{Q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + N^2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}'}(1^\lambda) + \frac{Q_{\mathsf{S}}^2}{2^{n-1}} + \mathsf{negl}(\lambda).$$

This completes the proof. □

6.3 Proof of Lemma 6.2

This section provides the proof of Lemma 6.2, formally stated below.

Lemma 6.2. *If there exist an adversary $\mathcal{A}$ against the security game in Game_8 with advantage ϵ_8, then we can construct an adversary $\mathcal{B}$ against the $\mathsf{AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L},B_{\mathcal{L}_{\mathsf{TS}}},B_{\mathbf{s}},B_{\mathbf{e}}}$ problem such that*

$$\epsilon_8 \leq \sqrt{Q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + \mathsf{negl}(\lambda),$$

where $Q_{\mathsf{RO}} = Q_{\mathsf{H}} + 2Q_{\mathsf{G}} + 2Q_{\mathsf{S}} + 1$ and the parameters from Section 6.1. Moreover, we have $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$.

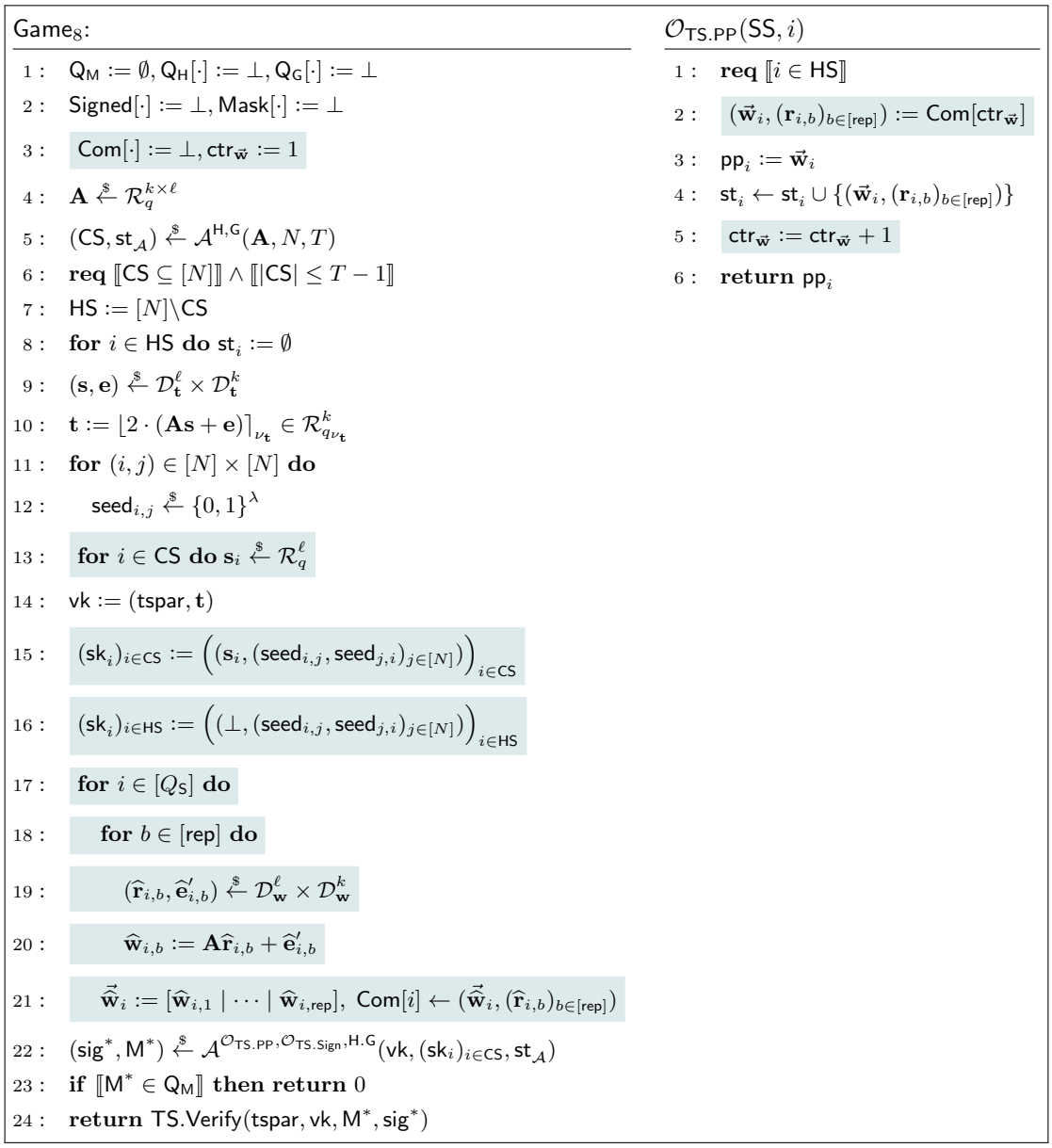

Figure 14: The final eighth game. The changes between the previous game are highlighted in blue.

Proof. To show this lemma, we construct an adversary $\mathcal{B}$ solving the AOM-MLWE problem which internally runs the adversary $\mathcal{A}$ against the security game in Game_8. $\mathcal{B}$ is given $(\mathbf{A}, \mathbf{T}) \in \mathcal{R}_q^{k\times\ell} \times \mathcal{R}_q^{k\times(\mathsf{rep}\cdot Q_{\mathsf{S}}+1)}$ as the problem instance, where $Q = \mathsf{rep} \cdot Q_{\mathsf{S}} + 1$. At a high level, $\mathcal{B}$ invokes the forking lemma and simulates the view of $\mathcal{A}$ using at most $Q - 1 = \mathsf{rep} \cdot Q_{\mathsf{S}}$ queries to its MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$. It then extracts the Q-th solution from the two forgeries it obtains from $\mathcal{A}$ to prepare the Q MLWE solutions. The proof consists of five parts: (1) we first explain the necessary algorithms to invoke the forking lemma; (2) we then explain how $\mathcal{B}$ invokes the forking lemma; (3) we explain the necessary conditions for $\mathcal{B}$ to be able to extract the Q MLWE solutions; (4) we explain how $\mathcal{B}$ extracts the Q MLWE solutions; and lastly, (5) explain the solutions are indeed valid.

(1) Algorithms $(\mathcal{C}_{\mathsf{TS}}, \mathcal{O})$ required to invoke the Forking Lemma. We first define the algorithm $\mathcal{C}_{\mathsf{TS}}$ and $\mathcal{O}$ to be used in the forking algorithm $\mathsf{Fork}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}_{\mathcal{C}_{\mathsf{TS}}}(\mathsf{par})$ of the (oracle-aided) forking lemma in Lemma 3.21. Looking ahead, $\mathcal{C}_{\mathsf{TS}}$ is almost identical to the challenger in Game_8 and $\mathcal{O}(\overline{\mathsf{par}},\cdot)$ will be set to the MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$ with some additional book keeping.

Let us first define the input generator IG, the set $\mathcal{H}$, and the integer q_{Fork} in Lemma 3.21. IG first samples $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$, $(\mathbf{s},\mathbf{e}) \xleftarrow{\$} \mathcal{D}_{\mathbf{t}}^{\ell} \times \mathcal{D}_{\mathbf{t}}^{k}$, and $(\widehat{\mathbf{r}}_{i,b}, \widehat{\mathbf{e}}'_{i,b}) \xleftarrow{\$} \mathcal{D}_{\mathbf{w}}^{\ell} \times \mathcal{D}_{\mathbf{w}}^{k}$ for $(i,b) \in [Q_{\mathsf{S}}] \times [\mathsf{rep}]$. It then sets $(\mathbf{S},\mathbf{E}) = \left([2\cdot\mathbf{s} \mid \widehat{\mathbf{r}}_{1,1} \mid \widehat{\mathbf{r}}_{1,2} \mid \cdots \mid \widehat{\mathbf{r}}_{Q_{\mathsf{S}},\mathsf{rep}}], \left[2\cdot\mathbf{e} \mid \widehat{\mathbf{e}}'_{1,1} \mid \widehat{\mathbf{e}}'_{1,2} \mid \cdots \mid \widehat{\mathbf{e}}'_{Q_{\mathsf{S}},\mathsf{rep}}\right] \right) \in \mathcal{R}_q^{\ell\times(\mathsf{rep}\cdot Q_{\mathsf{S}}+1)} \times \mathcal{R}_q^{k\times(\mathsf{rep}\cdot Q_{\mathsf{S}}+1)}$ and $\mathbf{T} = \mathbf{AS}+\mathbf{E}$. Finally, it outputs $(\mathsf{par},\overline{\mathsf{par}}) = \big((\mathbf{A},\mathbf{T}),(\mathbf{S},\mathbf{E})\big)$. We define $\mathcal{H} := \mathcal{C} \times \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$. For $h \in \mathcal{H}$, we write $h = (c, (\beta_b)_{b\in[\mathsf{rep}]})$, where $\beta_1 = 1$. Lastly, we set $q_{\mathsf{Fork}} := Q_{\mathsf{RO}} := Q_{\mathsf{H}} + 2Q_{\mathsf{G}} + 2Q_{\mathsf{S}} + 1$.

The description of $\mathcal{O}(\overline{\mathsf{par}},\cdot)$ is simple. When $\mathcal{O}$ is queried on $\mathbf{d} = [d_0 \mid \cdots \mid d_{\mathsf{rep}\cdot Q_{\mathsf{S}}}]$, it outputs $\perp$ if $\mathbf{d} \notin \mathcal{L}_{\mathsf{TS}}$. Here, note that membership of $\mathcal{L}_{\mathsf{TS}}$ is efficiently checkable. Otherwise, it simply outputs $(\mathbf{Sd},\mathbf{Ed})$. Notice that $\mathcal{O}(\overline{\mathsf{par}},\cdot)$ is deterministic as required.

We next define $\mathcal{C}_{\mathsf{TS}}$, given as input par and $\vec{h} \in \mathcal{H}^{Q_{\mathsf{RO}}}$. $\mathcal{C}_{\mathsf{TS}}$ first simulates the Game_8 challenger, except for three modifications. The first modification is that it uses $\mathbf{A}$ provided in par and generates $\mathbf{t}$ and $(\vec{\widehat{\mathbf{w}}}_i)_{i\in[Q_{\mathsf{S}}]}$ from $\mathbf{T}$ in par. That is, it parses $[\mathbf{t}^\star_0 \mid \cdots \mid \mathbf{t}^\star_{\mathsf{rep}\cdot Q_{\mathsf{S}}}] \leftarrow \mathbf{T}$ and sets $\mathbf{t} := \lfloor \mathbf{t}^\star_0 \rceil_{\nu_{\mathbf{t}}}$ and $\vec{\widehat{\mathbf{w}}}_i := [\mathbf{t}^\star_{\mathsf{rep}\cdot(i-1)+1} \mid \cdots \mid \mathbf{t}^\star_{\mathsf{rep}\cdot i}]$ for $i \in [Q_{\mathsf{S}}]$. Since $\mathcal{D}_1 = 2\cdot\mathcal{D}_{\mathbf{t}}$ and $\mathcal{D}_i = \mathcal{D}_{\mathbf{w}}$ for $i \in [2, \mathsf{rep}\cdot Q_{\mathsf{S}}+1]$, these are identically distributed to those of Game_8.

The second modification is that it answers the random oracle queries using $\vec{h}$ with $\left|\vec{h}\right| = Q_{\mathsf{RO}}$, instead of sampling them on its own. More concretely, it initializes a counter ctr to 1 and increments it any time a new entry in $\mathsf{Q}_{\mathsf{H}}[\cdot]$ or $\mathsf{Q}_{\mathsf{G}}[\cdot]$ needs to be defined. Moreover, when a new entry of $\mathsf{Q}_{\mathsf{H}}[\cdot]$ (resp. $\mathsf{Q}_{\mathsf{G}}[\cdot]$) needs to be defined, it parses $(c_{\mathsf{ctr}}, (\beta_{\mathsf{ctr},b})_{b\in[\mathsf{rep}]}) \leftarrow h_{\mathsf{ctr}} \in \mathcal{H}$ and uses c_{ctr} (resp. $(\beta_{\mathsf{ctr},b})_{b\in[\mathsf{rep}]}$), where the unused $(\beta_{\mathsf{ctr},b})_{b\in[\mathsf{rep}]}$ (resp. c_{ctr}) is discarded. This change is only conceptual since every h_{ctr} is uniformly chosen from $\mathcal{H}$. Here, recall an adversary $\mathcal{A}$ against the unforgeability of the threshold signature scheme makes at most Q_{H}, Q_{G}, and Q_{S} queries to the random oracles H, G, and signing oracle TS.Sign, respectively. Therefore, $\vec{h}$ with $\left|\vec{h}\right| = Q_{\mathsf{RO}} = Q_{\mathsf{H}} + 2Q_{\mathsf{G}} + 2Q_{\mathsf{S}} + 1$ suffices.

The third modification is that when $\mathcal{C}_{\mathsf{TS}}$ needs to generate a response $\mathbf{z}_i$ during answering queries to $\mathcal{O}_{\mathsf{TS.Sign}}$, it does so by accessing $\mathcal{O}(\overline{\mathsf{par}},\cdot)$. Recall $\overline{\mathsf{par}}$ includes all the secret of the challenger in Game_8. More concretely, there are two types of responses $\mathbf{z}_i$ to consider. One is the response made by the last user in sHS (i.e., Line 18 of Fig. 13) and the other is the response made by the other users (i.e., Line 21 of Fig. 13). We show how to generate $\mathbf{z}_i$ for both types of responses. Assume $\mathcal{O}_{\mathsf{TS.Sign}}$ is queried on user i, signer set SS, and $\vec{\mathbf{w}}_i$ is user i's commitment stored in st_i. Further, assume $\vec{\mathbf{w}}_i = \vec{\widehat{\mathbf{w}}}_\kappa$ for some $\kappa \in [Q_{\mathsf{S}}]$, where the existence of such κ is guaranteed by the construction and the uniqueness is (implicitly) guaranteed from the argument in Game_5.

Now, for the first type of response, it executes the same procedure until Line 18 of Fig. 13 and then prepares $\mathbf{d}_\kappa := [d_{\kappa,0} \mid \cdots \mid d_{\kappa,\mathsf{rep}\cdot Q_{\mathsf{S}}}]$, where $d_{\kappa,0} = c$, $(d_{\kappa,j})_{j\in[\mathsf{rep}\cdot(\kappa-1)+1,\mathsf{rep}\cdot\kappa]} = (\beta_{\kappa,b})_{b\in[\mathsf{rep}]}$, and $d_{\kappa,j} = 0$ for all other $j \in [0, \mathsf{rep}\cdot Q_{\mathsf{S}}]$. It then queries $\mathbf{d}_\kappa$ to $\mathcal{O}(\overline{\mathsf{par}},\cdot)$, receives $(\mathbf{s}_\kappa,\mathbf{e}_\kappa)$, and computes

$$\mathbf{z}_i := \mathbf{s}_\kappa - c\sum_{j\in\mathsf{sCS}} L_{\mathsf{SS},j}\cdot\mathbf{s}_j + \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} - \mathbf{m}_i.$$

This is identical to the partial response generated in Game_8 since we have $\mathbf{s}_\kappa = 2\cdot c\cdot\mathbf{s} + \sum_{b\in[\mathsf{rep}]}\beta_{\kappa,b}\cdot\widehat{\mathbf{r}}_{\kappa,b}$, where recall $\vec{\widehat{\mathbf{w}}}_\kappa$ is $\vec{\mathbf{w}}_i$. For the second type of response, it executes the same procedure until Line 21 of Fig. 13 and then prepares $\mathbf{d}_\kappa := [d_{\kappa,0} \mid \cdots \mid d_{\kappa,\mathsf{rep}\cdot Q_{\mathsf{S}}}]$, where $(d_{\kappa,j})_{j\in[\mathsf{rep}\cdot(\kappa-1)+1,\mathsf{rep}\cdot\kappa]} = (\beta_{\kappa,b})_{b\in[\mathsf{rep}]}$, and $d_{\kappa,j} = 0$ for all other $j \in [0, \mathsf{rep}\cdot Q_{\mathsf{S}}]$. It then queries $\mathbf{d}_\kappa$ to $\mathcal{O}(\overline{\mathsf{par}},\cdot)$, receives $(\mathbf{s}_\kappa,\mathbf{e}_\kappa)$, and computes

$$\mathbf{z}_i := \mathbf{s}_\kappa + \mathbf{m}^*_{\mathsf{sHS},i} + \mathbf{m}^*_{\mathsf{sCS},i} - \mathbf{m}_i.$$

Similarly to the above argument, this is identical to the partial response in Game_8.

It remains to explain what $\mathcal{C}_{\mathsf{TS}}$ does after it simulates the modified Game_8 challenger. At the end of the simulation, $\mathcal{C}_{\mathsf{TS}}$ is given the forgery $(\mathsf{sig}^* = (c^*,\mathbf{z}^*,\mathbf{h}^*), \mathsf{M}^*)$ from $\mathcal{A}$. It first checks the validity of the

forgery as in Game_8 and returns $(0, \bot)$ if the forgery is invalid. Otherwise, because the forgery is valid, there must exist an index $I^* \in [Q_{\mathsf{RO}}]$ for which $c_{I^*} = \mathsf{H}(\mathsf{vk}, \mathsf{M}^*, \lfloor \mathbf{A}\mathbf{z}^* - 2^{\nu_{\mathbf{t}}} \cdot c^* \cdot \mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^*)$, where $c_{I^*} \in h_{I^*}$ and $c^* = c_{I^*}$. In this case, $\mathcal{C}_{\mathsf{TS}}$ outputs (I^*, sig^*). Since $\mathcal{C}_{\mathsf{TS}}$ simulates Game_8 perfectly, we have

$$\mathsf{acc} = \Pr\left[(I^*, \mathsf{sig}^*) \xleftarrow{\$} \mathcal{C}_{\mathsf{TS}}^{\mathcal{O}(\overline{\mathsf{par}}, \cdot)}(\mathsf{par}, \vec{h}) : I^* \geq 1\right] = \epsilon_8.$$

(2) $\mathcal{B}$ invoking the Forking algorithm $\mathsf{Fork}_{\mathcal{C}_{\mathsf{TS}}}^{\mathcal{O}(\overline{\mathsf{par}}, \cdot)}(\mathsf{par})$. We now explain how $\mathcal{B}$ invokes the forking algorithm $\mathsf{Fork}_{\mathcal{C}_{\mathsf{TS}}}^{\mathcal{O}(\overline{\mathsf{par}}, \cdot)}(\mathsf{par})$ and prepares its state to extract the Q MLWE solutions. Recall $\mathcal{B}$ is given $(\mathbf{A}, \mathbf{T}) \in \mathcal{R}_q^{k \times \ell} \times \mathcal{R}_q^{k \times (\mathsf{rep} \cdot Q_{\mathsf{S}} + 1)}$ as the AOM-MLWE problem and has access to an MLWE solving oracle $\mathcal{O}_{\mathsf{solve}}$.

Instead of running IG, $\mathcal{B}$ sets $\mathsf{par} = (\mathbf{A}, \mathbf{T})$ and (perfectly) simulates $\mathcal{O}(\overline{\mathsf{par}}, \cdot)$ using $\mathcal{O}_{\mathsf{solve}}(\cdot)$. It then executes $\mathsf{Fork}_{\mathcal{C}_{\mathsf{TS}}}^{\mathcal{O}_{\mathsf{solve}}(\cdot)}(\mathsf{par})$ which behaves identically to $\mathsf{Fork}_{\mathcal{C}_{\mathsf{TS}}}^{\mathcal{O}(\overline{\mathsf{par}}, \cdot)}(\mathsf{par})$. From the forking lemma (cf. Lemma 3.21), $\mathcal{B}$ obtains two vaild forgeries sig_1^* and sig_2^*, which are involved in the same input of H, with probability

$$\mathsf{frk} \geq \mathsf{acc} \cdot \left(\frac{\mathsf{acc}}{Q_{\mathsf{RO}}} - \frac{1}{h}\right) = \frac{\epsilon_8^2}{Q_{\mathsf{RO}}} - \mathsf{negl}(\lambda), \tag{8}$$

where the inequality follows from $h = |\mathcal{H}| = |\mathcal{C}| \cdot |\mathbb{T}|^{\mathsf{rep}-1}$ and $|\mathcal{C}| \geq 2^\lambda$. Moreover, throughout the execution, $\mathcal{B}$ maintains a list $\mathsf{Ans}[\cdot]$, initially empty. Whenever $\mathcal{O}(\overline{\mathsf{par}}, \cdot)$ is queried on $\mathbf{d} \in \mathcal{L}_{\mathsf{TS}}$, it first parses $[d_0 \mid \cdots \mid d_{\mathsf{rep} \cdot Q_{\mathsf{S}}}] \leftarrow \mathbf{d}$, where by the definition of $\mathcal{C}_{\mathsf{TS}}$, we have $d_0 \in \mathcal{C} \cup \{0\}$ and for some $\kappa \in [Q_{\mathsf{S}}]$, $(d_j)_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]} \in \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$, and $d_j = 0$ for all other $j \in [\mathsf{rep} \cdot Q_{\mathsf{S}}]$. It then stores the reply from $\mathcal{O}(\overline{\mathsf{par}}, \cdot)$ as $\mathsf{Ans}[\kappa] \leftarrow \mathsf{Ans}[\kappa] \cup \{(\mathbf{d}, \mathbf{Sd}, \mathbf{Ed})\}$. Here, note that $\mathsf{Ans}[\kappa]$ can store up to 2 tuples. This is because by definition of $\mathcal{C}_{\mathsf{TS}}$, every $\mathbf{d}$ has a corresponding user commitment $\vec{\mathbf{w}}_i$ and the κ defined via $\mathbf{d}$ satisfies $\vec{\mathbf{w}}_i = \vec{\widehat{\mathbf{w}}}_\kappa$. Therefore, since the same user commitment $\vec{\mathbf{w}}_i$ cannot be reused during a single invocation of $\mathcal{C}_{\mathsf{TS}}$, the same κ can appear at most twice.

(3) Conditions on which the extraction by $\mathcal{B}$ succeeds. Before explaining how $\mathcal{B}$ extracts the Q MLWE solutions, we explain two bad conditions that must not occur for $\mathcal{B}$ to be able to extract, and prove that these two bad conditions can happen with only negligible probability. The first condition is standard in Fiat-Shamir type proofs: we denote Collc as the event when the two challenges (c_1^*, c_2^*) included in $\mathsf{sig}_1^* = (c_1^*, \mathbf{z}_1^*, \mathbf{h}_1^*)$ and $\mathsf{sig}_2^* = (c_2^*, \mathbf{z}_2^*, \mathbf{h}_2^*)$ are identical. That is, $c_1^* = c_2^*$. Since both challenges are uniformly random over $\mathcal{C}$, it is immediate that $\Pr[\mathsf{Collc}] = 1/|\mathcal{C}| = \mathsf{negl}(\lambda)$.

The second condition is unique to threshold signatures. For every $\kappa \in [Q_{\mathsf{S}}]$ such that $|\mathsf{Ans}[\kappa]| = 2$, let $\{(\mathbf{d}_\kappa, \mathbf{s}_\kappa, \mathbf{e}_\kappa), (\mathbf{d}'_\kappa, \mathbf{s}'_\kappa, \mathbf{e}'_\kappa)\} \leftarrow \mathsf{Ans}[\kappa]$. By definition, we can parse $[d_{\kappa,0} \mid \cdots \mid d_{\kappa, \mathsf{rep} \cdot Q_{\mathsf{S}}}] \leftarrow \mathbf{d}_\kappa$, where $d_{\kappa,0} \in \mathcal{C} \cup \{0\}$, $(d_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]} \in \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$, and $d_{\kappa,j} = 0$ for all other $j \in [\mathsf{rep} \cdot Q_{\mathsf{S}}]$. We parse $\mathbf{d}'_\kappa$ similarly. We then denote $\mathsf{BadQuery}$ as the event that there exists $\kappa \in [Q_{\mathsf{S}}]$ such that $d_{\kappa,0} \neq d'_{\kappa,0}$ but $(d_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]} = (d'_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]}$. To bound $\Pr[\mathsf{BadQuery}]$, recall that $\mathbf{d}_\kappa$ (resp. $\mathbf{d}'_\kappa$) is defined in the first (resp. second) invocation of $\mathcal{C}_{\mathsf{TS}}$. Moreover, by how $\mathbf{d}_\kappa$ is defined, there exists corresponding c and $(\beta_b)_{b \in [\kappa]}$ with $c = d_{\kappa,0}$ and $(\beta_b)_{b \in [\kappa]} = (d_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]}$ such that $(c, (\beta_b)_{b \in [\kappa]}) \in (h_{I_c}, h_{I_\beta})$ for some $h_{I_c}, h_{I_\beta} \in \vec{h}$ and distinct $I_c, I_\beta \in [Q_{\mathsf{RO}}]$. In other words, $d_{\kappa,0}$ and $(d_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]}$ are the outputs of the random oracles H and G, respectively. Similarly, we define the indices $I'_c, I'_\beta \in [Q_{\mathsf{RO}}]$ induced by $\mathbf{d}'_\kappa$.

First, assume $I^* < I_\beta$ or $I^* < I'_\beta$. Then, since either h_{I_β} or $h'_{I'_\beta}$ are sampled uniformly and independently after the forking point I^*, the probability that $(d_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]} = (d'_{\kappa,j})_{j \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]}$ (i.e., $(\beta_b)_{b \in [\kappa]} = (\beta'_b)_{b \in [\kappa]}$) is bounded by $\frac{1}{|\mathbb{T}^{\mathsf{rep}-1}|} = \mathsf{negl}(\lambda)$. Following a similar argument, we cannot have $I_\beta \neq I'_\beta$ when $I_\beta, I'_\beta < I^*$ with all but a negligible probability. Moreover, we have $I_\beta \neq I^*$ since $c_{I^*} \in h_{I^*}$ is used; that is, the β-terms in h_{I^*} are discarded. Therefore, we can assume $I_\beta = I'_\beta < I^*$ if $\mathsf{BadQuery}$ occurs.

We next have $I_c, I'_c \neq I^*$ since due to the winning condition of the unforgeability game, the adversary cannot query the signing oracle $\mathcal{O}_{\mathsf{TS.Sign}}$ on M^*. Lastly, due to the modification we made in Game_2, we have either $I_c < I_\beta$ or $I_c = I_\beta + 1$ and either $I'_c < I_\beta$ or $I'_c = I_\beta + 1$. Combining $I_c, I'_c \neq I^*$ and $I_\beta = I'_\beta < I^*$, and the fact that the behavior of $\mathcal{C}_{\mathsf{TS}}$ is identical in the two runs before the rewinding point,

we must have $I_c = I'_c$. However, this implies that $\mathsf{BadQuery}$ does not occur. To summarize, we can bound $\Pr[\mathsf{BadQuery}] \leq \frac{2 \cdot Q_{\mathsf{S}}}{|\mathbb{T}^{\mathsf{rep}-1}|} = \mathsf{negl}(\lambda)$ by taking the union bound over all $\kappa \in [Q_{\mathsf{S}}]$.

As we show in Items (4) and (5) below, when CollC and $\mathsf{BadQuery}$ do not occur, $\mathcal{B}$ will be able to extract Q MLWE solutions to break the AOM-MLWE problem. Therefore, we conclude

$$\begin{aligned}\mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda) &= \Pr\left[(b, \mathsf{sig}^*_1, \mathsf{sig}^*_2) \leftarrow \mathsf{Fork}^{\mathcal{O}(\overline{\mathsf{par}},\cdot)}_{\mathcal{C}_{\mathsf{TS}}}(\mathsf{par}) : b = 1 \wedge \neg\mathsf{CollC} \wedge \neg\mathsf{BadQuery}\right] \\ &\geq \mathsf{frk} - \Pr[\mathsf{CollC}] - \Pr[\mathsf{BadQuery}] \\ &\geq \frac{\epsilon_8^2}{Q_{\mathsf{RO}}} - \mathsf{negl}(\lambda),\end{aligned}$$

where the final inequality follows from Eq. (8) and the bounds we established for $\Pr[\mathsf{CollC}]$ and $\Pr[\mathsf{BadQuery}]$. We can rewrite the bound to arrive at the following statement in the lemma:

$$\epsilon_8 \leq \sqrt{Q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + \mathsf{negl}(\lambda).$$

(4) $\mathcal{B}$ extracting the Q MLWE solutions. To prove Lemma 6.2, it remains to show how $\mathcal{B}$ extracts the solution of the AOM-MLWE problem from sig^*_1, sig^*_2, and the list $\mathsf{Ans}[\cdot]$, conditioned on CollC and $\mathsf{BadQuery}$ not occurring. Here, we first focus on how $\mathcal{B}$ extracts a solution and postpone verifying that the solution is valid to Item (5). Below, we will be precise on where each elements live and be explicit about our use of the the lift notation, i.e., $\bar{x} \in [0, 1, \cdots, q-1]$ for $x \in \mathcal{R}_q$ (see Section 3.7.2 for more detail).

First, $\mathcal{B}$ extracts $(v_0, \widehat{\mathbf{s}}_0, \widehat{\mathbf{e}}_0)$ satisfying $v_0 \cdot \mathbf{t}^\star_0 = \mathbf{A} \cdot \widehat{\mathbf{s}}_0 + \widehat{\mathbf{e}}_0 \in R^k_q$, i.e., an *approximate* secret key associated to the verification key $\mathbf{t} = \lfloor \mathbf{t}^\star_0 \rceil_{\nu_{\mathbf{t}}} \in R^k_{q_{\nu_{\mathbf{t}}}}$. Due to the forking lemma, the challenges c^*_1 and c^*_2 are generated on the same input to the random oracle H. Therefore, $\mathsf{sig}^*_1 = (c^*_1, \mathbf{z}^*_1, \mathbf{h}^*_1)$ and $\mathsf{sig}^*_2 = (c^*_2, \mathbf{z}^*_2, \mathbf{h}^*_2)$ satisfy

$$\left\lfloor \mathbf{A}\mathbf{z}^*_1 - 2^{\nu_{\mathbf{t}}} \cdot c^*_1 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^*_1 = \left\lfloor \mathbf{A}\mathbf{z}^*_2 - 2^{\nu_{\mathbf{t}}} \cdot c^*_2 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^*_2 \mod q_{\nu_{\mathbf{w}}}.$$

Let us consider taking the lift of both sides:

$$\overline{\left\lfloor \mathbf{A}\mathbf{z}^*_1 - 2^{\nu_{\mathbf{t}}} \cdot c^*_1 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^*_1} = \overline{\left\lfloor \mathbf{A}\mathbf{z}^*_2 - 2^{\nu_{\mathbf{t}}} \cdot c^*_2 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^*_2}.$$

Then, since the output of $\lfloor \cdot \rceil_{\nu_{\mathbf{w}}}$ is over $\mathcal{R}_{q_{\nu_{\mathbf{w}}}}$ and $\mathbf{h}^*_1, \mathbf{h}^*_2$ are also over $\mathcal{R}_{q_{\nu_{\mathbf{w}}}}$, there exists a unique vector $\boldsymbol{\delta}_1 \in \mathcal{R}^k$ with $\|\boldsymbol{\delta}_1\|_\infty \leq 2$ such that

$$\overline{\left\lfloor \mathbf{A}\mathbf{z}^*_1 - 2^{\nu_{\mathbf{t}}} \cdot c^*_1 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}}} + \overline{\mathbf{h}^*_1} = \overline{\left\lfloor \mathbf{A}\mathbf{z}^*_2 - 2^{\nu_{\mathbf{t}}} \cdot c^*_2 \cdot \overline{\mathbf{t}} \right\rceil_{\nu_{\mathbf{w}}}} + \overline{\mathbf{h}^*_2} + q_{\nu_{\mathbf{w}}} \cdot \boldsymbol{\delta}_1. \tag{9}$$

Clearly, the equality also holds over modulo q. Let us now multiply both sides by $2^{\nu_{\mathbf{w}}}$ and apply Lemma 3.14, Eq. (2) as follows:

$$\mathbf{A}\mathbf{z}^*_1 - c^*_1 \cdot \mathbf{t}^\star_0 + 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}^*_1} = \mathbf{A}\mathbf{z}^*_2 - c^*_2 \cdot \mathbf{t}^\star_0 + 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}^*_2} + 2^{\nu_{\mathbf{w}}} \cdot q_{\nu_{\mathbf{w}}} \cdot \boldsymbol{\delta}_1 + (\boldsymbol{\delta}_2 + c^*_1 \cdot \boldsymbol{\delta}_3 + c^*_2 \cdot \boldsymbol{\delta}_4) \mod q, \tag{10}$$

where $\boldsymbol{\delta}_2, \boldsymbol{\delta}_3, \boldsymbol{\delta}_4 \in \mathcal{R}^k_q$ satisfy $\|\boldsymbol{\delta}_2\|_\infty \leq 2 \cdot (2^{\nu_{\mathbf{w}}} - 1)$ and $\|\boldsymbol{\delta}_3\|_\infty, \|\boldsymbol{\delta}_4\|_\infty \leq 2^{\nu_{\mathbf{t}}} - 1$. In particular, $\boldsymbol{\delta}_2, \boldsymbol{\delta}_3, \boldsymbol{\delta}_4$ are the noise incurred by invoking Lemma 3.14, Eq. (2). Rearranging the terms, we obtain the following:

$$\underbrace{(c^*_1 - c^*_2)}_{=:v_0} \cdot \mathbf{t}^\star_0 = \mathbf{A} \cdot \underbrace{(\mathbf{z}^*_1 - \mathbf{z}^*_2)}_{=:\widehat{\mathbf{s}}_0} + \underbrace{(2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}^*_1} - 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}^*_2} - (q_{\mathsf{bot}} \cdot \boldsymbol{\delta}_1 + \boldsymbol{\delta}_2 + c^*_1 \cdot \boldsymbol{\delta}_3 + c^*_2 \cdot \boldsymbol{\delta}_4)}_{=:\widehat{\mathbf{e}}_0} \mod q. \tag{11}$$

Here, we use the fact that we can be uniquely expressed as $q = 2^{\nu_{\mathbf{w}}} q_{\nu_{\mathbf{w}}} + q_{\mathsf{bot}}$ with $q_{\mathsf{bot}} \in [0, 2^{\nu_{\mathbf{w}}-1} - 1]$ to swap $2^{\nu_{\mathbf{w}}} q_{\nu_{\mathbf{w}}}$ with q_{bot} (see also Lemma 3.14). Finally, $\mathcal{B}$ sets $(v_0, \widehat{\mathbf{s}}_0, \widehat{\mathbf{e}}_0)$ as shown in Eq. (11). Since $c^*_1 \neq c^*_2$ holds when CollC does not occur, $v_0 \neq 0$ holds. As mentioned above, we postpone verifying the size bound of the solution to Item (5).

Next, for each $\kappa \in [Q_{\mathsf{S}}]$, $\mathcal{B}$ extracts $(v_i, \widehat{\mathbf{s}}_i, \widehat{\mathbf{e}}_i)_{i \in [\mathsf{rep} \cdot (\kappa-1)+1, \mathsf{rep} \cdot \kappa]}$ such that $v_i \cdot \mathbf{t}^\star_i = \mathbf{A} \cdot \widehat{\mathbf{s}}_i + \widehat{\mathbf{e}}_i$ from $\mathsf{Ans}[\kappa]$ and $(v_0, \widehat{\mathbf{s}}_0, \widehat{\mathbf{e}}_0)$. Below, for readability, we denote the MLWE solution $(\mathbf{S}, \mathbf{E}) \in \mathcal{R}_q^{\ell \times (\mathsf{rep} \cdot Q_{\mathsf{S}}+1)} \times \mathcal{R}_q^{k \times (\mathsf{rep} \cdot Q_{\mathsf{S}}+1)}$

used by the AOM-MLWE challenger as $\big((\mathbf{s}_0^\star, \mathbf{s}_1^\star, \cdots, \mathbf{s}_{\mathsf{rep}\cdot Q_\mathsf{S}}^\star), (\mathbf{e}_0^\star, \mathbf{e}_1^\star, \cdots, \mathbf{e}_{\mathsf{rep}\cdot Q_\mathsf{S}}^\star)\big)$, that is, $\mathbf{t}_i^\star = \mathbf{A}\cdot\mathbf{s}_i^\star + \mathbf{e}_i^\star$ for $i \in [\mathsf{rep}\cdot Q_\mathsf{S}+1]$.

To this end, $\mathcal{B}$ first performs a preparation step to "complete" the list $\mathsf{Ans}[\kappa]$. Recall $|\mathsf{Ans}[\kappa]| \le 2$. If $|\mathsf{Ans}[\kappa]| = 0$, then it defines $\mathbf{d}_\kappa \in \mathcal{L}_{\mathsf{TS}}$ to be any vector such that $d_{\kappa,0} = 0$, $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+1} = 1$, $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b} \in \mathbb{T}$ for $b \in [2,\mathsf{rep}]$, and $d_{\kappa,j} = 0$ for all other $j \in [\mathsf{rep}\cdot Q_\mathsf{S}]$. It also defines $\mathbf{d}'_\kappa$ to be identical to $\mathbf{d}_\kappa$, except that $d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+2} \in \mathbb{T}$ is distinct from $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+2}$. $\mathcal{B}$ then queries $\mathcal{O}_{\mathsf{solve}}$ on input $\mathbf{d}_\kappa$ and $\mathbf{d}'_\kappa$, and receives $(\mathbf{s}_\kappa, \mathbf{e}_\kappa)$ and $(\mathbf{s}'_\kappa, \mathbf{e}'_\kappa)$, respectively. Lastly, it updates $\mathsf{Ans}[\kappa] \leftarrow \mathsf{Ans}[\kappa] \cup \{(\mathbf{d}_\kappa, \mathbf{s}_\kappa, \mathbf{e}_\kappa), (\mathbf{d}'_\kappa, \mathbf{s}'_\kappa, \mathbf{e}'_\kappa)\}$. Next, if $|\mathsf{Ans}[\kappa]| = 1$, denote $\{(\mathbf{d}_\kappa, \mathbf{s}_\kappa, \mathbf{e}_\kappa)\} \leftarrow \mathsf{Ans}[\kappa]$. By definition, we can parse $[d_{\kappa,0} \mid \cdots \mid d_{\kappa,\mathsf{rep}\cdot Q_\mathsf{S}}] \leftarrow \mathbf{d}_\kappa$, where $d_{\kappa,0} \in \mathcal{C} \cup \{0\}$, $(d_{\kappa,j})_{j\in[\mathsf{rep}\cdot(\kappa-1)+1,\mathsf{rep}\cdot\kappa]} \in \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$, and $d_{\kappa,j} = 0$ for all other $j \in [\mathsf{rep}\cdot Q_\mathsf{S}]$. It also defines $\mathbf{d}'_\kappa$ to be identical to $\mathbf{d}_\kappa$, except that $d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+2} \in \mathbb{T}$ that is distinct from $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+2}$. $\mathcal{B}$ then queries $\mathcal{O}_{\mathsf{solve}}$ on input d'_κ, receives $(\mathbf{s}'_\kappa, \mathbf{e}'_\kappa)$, and finally updates $\mathsf{Ans}[\kappa] \leftarrow \mathsf{Ans}[\kappa] \cup \{(\mathbf{d}'_\kappa, \mathbf{s}'_\kappa, \mathbf{e}'_\kappa)\}$.

At the end of the preparation of $\mathsf{Ans}[\cdot]$, for all $\kappa \in [Q_\mathsf{S}]$, we have $|\mathsf{Ans}[\kappa]| = 2$. Since the number of elements in the list $\mathsf{Ans}[\cdot]$ is the number of times $\mathcal{B}$ queried $\mathcal{O}_{\mathsf{solve}}$, we know that $\mathcal{B}$ queried $\mathcal{O}_{\mathsf{solve}}$ $2\cdot Q_\mathsf{S}$ times so far. Moreover, for any $\{(\mathbf{d}_\kappa, \mathbf{s}_\kappa, \mathbf{e}_\kappa), (\mathbf{d}'_\kappa, \mathbf{s}'_\kappa, \mathbf{e}'_\kappa)\} \leftarrow \mathsf{Ans}[\kappa]$, we can parse $[d_{\kappa,0} \mid \cdots \mid d_{\kappa,\mathsf{rep}\cdot Q_\mathsf{S}}] \leftarrow \mathbf{d}_\kappa$, where $d_{\kappa,0} \in \mathcal{C} \cup \{0\}$, $(d_{\kappa,j})_{j\in[\mathsf{rep}\cdot(\kappa-1)+1,\mathsf{rep}\cdot\kappa]} \in \{1\} \times \mathbb{T}^{\mathsf{rep}-1}$, and $d_{\kappa,j} = 0$ for all other $j \in [Q_\mathsf{S}\cdot\mathsf{rep}]$. Similarly for $\mathbf{d}'_\kappa$. From how $\mathbf{d}_\kappa$ and $\mathbf{d}'_\kappa$ are defined via $\mathcal{C}_{\mathsf{TS}}$ and $\mathcal{B}$, we have the following

$$\mathbf{A}\mathbf{s}_\kappa + \mathbf{e}_\kappa = d_{\kappa,0}\cdot\mathbf{t}_0^\star + \mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + \sum_{b\in[2,\mathsf{rep}]} d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+b} \tag{12}$$

$$\mathbf{A}\mathbf{s}'_\kappa + \mathbf{e}'_\kappa = d'_{\kappa,0}\cdot\mathbf{t}_0^\star + \mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + \sum_{b\in[2,\mathsf{rep}]} d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+b}, \tag{13}$$

and

$$\begin{bmatrix}\mathbf{s}_\kappa\\ \mathbf{e}_\kappa\\ \mathbf{s}'_\kappa\\ \mathbf{e}'_\kappa\end{bmatrix} = \begin{bmatrix}d_{\kappa,0}\cdot\mathbf{s}_0^\star\\ d_{\kappa,0}\cdot\mathbf{e}_0^\star\\ d'_{\kappa,0}\cdot\mathbf{s}_0^\star\\ d'_{\kappa,0}\cdot\mathbf{e}_0^\star\end{bmatrix} + \begin{bmatrix}\mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\end{bmatrix} + \sum_{b\in[2,\mathsf{rep}]}\begin{bmatrix}d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+b}\\ d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+b}\\ d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+b}\\ d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\cdot\mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+b}\end{bmatrix}. \tag{14}$$

$\mathcal{B}$ next queries $\mathcal{O}_{\mathsf{solve}}$ additionally $(\mathsf{rep}-2)\cdot Q_\mathsf{S}$ times. Recall that conditioning on BadQuery not occurring, there exists at least one index $\alpha_\kappa \in [\mathsf{rep}\cdot(\kappa-1)+2, \mathsf{rep}\cdot\kappa]$ s.t. $d_{\kappa,\alpha_\kappa} \neq d'_{\kappa,\alpha_\kappa}$. For each $i \in [\mathsf{rep}\cdot(\kappa-1)+2, \mathsf{rep}\cdot\kappa]\backslash\{\alpha_\kappa\}$, $\mathcal{B}$ prepares $\mathbf{d}_\kappa^{(i)} := [d_{\kappa,0}^{(i)} \mid \cdots \mid d_{\kappa,\mathsf{rep}\cdot Q_\mathsf{S}}^{(i)}]$ such that $d_{\kappa,i}^{(i)} = 1$ and $d_{\kappa,j}^{(i)} = 0$ otherwise, and queries $\mathbf{d}_\kappa^{(i)}$ to $\mathcal{O}_{\mathsf{solve}}$. By definition, $\mathcal{B}$ receives $(\mathbf{s}_i^\star, \mathbf{e}_i^\star)$, i.e., the exact MLWE solution associated to $\mathbf{t}_i^\star$, and sets $(v_i, \widehat{\mathbf{s}}_i, \widehat{\mathbf{e}}_i) := (1, \mathbf{s}_i^\star, \mathbf{e}_i^\star)$. At this point, combining with the $2\cdot Q_\mathsf{S}$ queries it used to complete the list $\mathsf{Ans}[\kappa]$, the total number of $\mathcal{O}_{\mathsf{solve}}$ query performed by $\mathcal{B}$ is $\mathsf{rep}\cdot Q_\mathsf{S}$.

It remains for $\mathcal{B}$ to extract $(v_{\mathsf{rep}\cdot(\kappa-1)+1}, \widehat{\mathbf{s}}_{\mathsf{rep}\cdot(\kappa-1)+1}, \widehat{\mathbf{e}}_{\mathsf{rep}\cdot(\kappa-1)+1})$ and $(v_{\alpha_\kappa}, \widehat{\mathbf{s}}_{\alpha_\kappa}, \widehat{\mathbf{e}}_{\alpha_\kappa})$ for $\kappa \in [Q_\mathsf{S}]$ without making any more $\mathcal{O}_{\mathsf{solve}}$ query. Let us first define

$$\begin{bmatrix}\widetilde{\mathbf{s}}_\kappa\\ \widetilde{\mathbf{e}}_\kappa\\ \widetilde{\mathbf{s}}'_\kappa\\ \widetilde{\mathbf{e}}'_\kappa\end{bmatrix} = \begin{bmatrix}\mathbf{s}_\kappa\\ \mathbf{e}_\kappa\\ \mathbf{s}'_\kappa\\ \mathbf{e}'_\kappa\end{bmatrix} - \sum_{i\in[\mathsf{rep}\cdot(\kappa-1)+2,\mathsf{rep}\cdot\kappa]\backslash\{\alpha_\kappa\}} d_{\kappa,i}\cdot\begin{bmatrix}\widehat{\mathbf{s}}_i\\ \widehat{\mathbf{e}}_i\\ \widehat{\mathbf{s}}_i\\ \widehat{\mathbf{e}}_i\end{bmatrix}.$$

Plugging in Eq. (14) and using fact that $(v_i, \widehat{\mathbf{s}}_i, \widehat{\mathbf{e}}_i) := (1, \mathbf{s}_i^\star, \mathbf{e}_i^\star)$ for all $i \in [\mathsf{rep}\cdot(\kappa-1)+2, \mathsf{rep}\cdot\kappa]\backslash\{\alpha_\kappa\}$, we can compute

$$\begin{bmatrix}\widetilde{\mathbf{s}}_\kappa\\ \widetilde{\mathbf{e}}_\kappa\\ \widetilde{\mathbf{s}}'_\kappa\\ \widetilde{\mathbf{e}}'_\kappa\end{bmatrix} = \begin{bmatrix}d_{\kappa,0}\cdot\mathbf{s}_0^\star\\ d_{\kappa,0}\cdot\mathbf{e}_0^\star\\ d'_{\kappa,0}\cdot\mathbf{s}_0^\star\\ d'_{\kappa,0}\cdot\mathbf{e}_0^\star\end{bmatrix} + \begin{bmatrix}\mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\\ \mathbf{e}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\end{bmatrix} + \begin{bmatrix}d_{\kappa,\alpha_\kappa}\cdot\mathbf{s}^\star_{\alpha_\kappa}\\ d_{\kappa,\alpha_\kappa}\cdot\mathbf{e}^\star_{\alpha_\kappa}\\ d'_{\kappa,\alpha_\kappa}\cdot\mathbf{s}^\star_{\alpha_\kappa}\\ d'_{\kappa,\alpha_\kappa}\cdot\mathbf{e}^\star_{\alpha_\kappa}\end{bmatrix}, \tag{15}$$

which in particular implies

$$\mathbf{A}\widetilde{\mathbf{s}}_\kappa + \widetilde{\mathbf{e}}_\kappa = d_{\kappa,0}\cdot\mathbf{t}_0^\star + \mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + d_{\kappa,\alpha_\kappa}\cdot\mathbf{t}^\star_{\alpha_\kappa}$$
$$\mathbf{A}\widetilde{\mathbf{s}}'_\kappa + \widetilde{\mathbf{e}}'_\kappa = d'_{\kappa,0}\cdot\mathbf{t}_0^\star + \mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + d'_{\kappa,\alpha_\kappa}\cdot\mathbf{t}^\star_{\alpha_\kappa}.$$

By multiplying the above two equations by v_0 and plugging in $\mathbf{t}_0^\star$ from Eq. (11), we have

$$v_0\cdot\left(\mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + d_{\kappa,\alpha_\kappa}\cdot\mathbf{t}^\star_{\alpha_\kappa}\right) = \mathbf{A}\Big(\underbrace{v_0\cdot\widetilde{\mathbf{s}}_\kappa - d_{\kappa,0}\cdot\widehat{\mathbf{s}}_0}_{=:\widehat{\mathbf{s}}^*}\Big) + \Big(\underbrace{v_0\cdot\widetilde{\mathbf{e}}_\kappa - d_{\kappa,0}\cdot\widehat{\mathbf{e}}_0}_{=:\widehat{\mathbf{e}}^*}\Big) \tag{16}$$

$$v_0\cdot\left(\mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + d'_{\kappa,\alpha_\kappa}\cdot\mathbf{t}^\star_{\alpha_\kappa}\right) = \mathbf{A}\Big(\underbrace{v_0\cdot\widetilde{\mathbf{s}}'_\kappa - d'_{\kappa,0}\cdot\widehat{\mathbf{s}}_0}_{=:\widehat{\mathbf{s}}^{*\prime}}\Big) + \Big(\underbrace{v_0\cdot\widetilde{\mathbf{e}}'_\kappa - d'_{\kappa,0}\cdot\widehat{\mathbf{e}}_0}_{=:\widehat{\mathbf{e}}^{*\prime}}\Big) \tag{17}$$

By subtracting Eq. (16) from Eq. (17), we obtain

$$\underbrace{v_0\cdot(d_{\kappa,\alpha_\kappa} - d'_{\kappa,\alpha_\kappa})}_{=:v_{\alpha_\kappa}}\cdot\mathbf{t}^\star_{\alpha_\kappa} = \mathbf{A}\,\underbrace{(\widehat{\mathbf{s}}^* - \widehat{\mathbf{s}}^{*\prime})}_{=:\widehat{\mathbf{s}}_{\alpha_\kappa}} + \underbrace{(\widehat{\mathbf{e}}^* - \widehat{\mathbf{e}}^{*\prime})}_{=:\widehat{\mathbf{e}}_{\alpha_\kappa}}. \tag{18}$$

By multiplying Eq. (16) with $d'_{\kappa,\alpha_\kappa}$, Eq. (17) with d_{κ,α_κ}, and subtracting them, we obtain

$$\underbrace{v_0\cdot(d'_{\kappa,\alpha_\kappa} - d_{\kappa,\alpha_\kappa})}_{=:v_{\mathsf{rep}\cdot(\kappa-1)+1}}\cdot\mathbf{t}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} = \mathbf{A}\,\underbrace{(d'_{\kappa,\alpha_\kappa}\cdot\widehat{\mathbf{s}}^* - d_{\kappa,\alpha_\kappa}\cdot\widehat{\mathbf{s}}^{*\prime})}_{=:\widehat{\mathbf{s}}_{\mathsf{rep}\cdot(\kappa-1)+1}} + \underbrace{(d'_{\kappa,\alpha_\kappa}\cdot\widehat{\mathbf{e}}^* - d_{\kappa,\alpha_\kappa}\cdot\widehat{\mathbf{e}}^{*\prime})}_{=:\widehat{\mathbf{e}}_{\mathsf{rep}\cdot(\kappa-1)+1}}. \tag{19}$$

Finally, $\mathcal{B}$ sets $\mathbf{v} := (v_0,\ldots,v_{\mathsf{rep}\cdot Q_\mathsf{S}})$, $\widehat{\mathbf{S}} := [\widehat{\mathbf{s}}_0 \mid \cdots \mid \widehat{\mathbf{s}}_{\mathsf{rep}\cdot Q_\mathsf{S}}]$, and $\widehat{\mathbf{E}} := [\widehat{\mathbf{e}}_0 \mid \cdots \mid \widehat{\mathbf{e}}_{\mathsf{rep}\cdot Q_\mathsf{S}}]$ as the solution to the $\mathsf{AOM\text{-}MLWE}$ game.

(5) Checking the validity of $\mathcal{B}$'s solution. Finally, it remains to show that $\mathcal{B}$ satisfies the winning condition of the $\mathsf{AOM\text{-}MLWE}$ game. We first check that all the queries $(\mathbf{d}_i)_{i\in[\mathsf{rep}\cdot Q_\mathsf{S}]}$ made by $\mathcal{B}$ is in $\mathcal{L}_{\mathsf{TS}}$ defined in Definition 4.8. For $\kappa\in[Q_\mathsf{S}]$, there are two queries $\mathbf{d}_\kappa$ and $\mathbf{d}'_\kappa$ satisfying $d_{\kappa,0}, d'_{\kappa,0}\in\mathcal{C}\cup\{0\}$, $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+1} = d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+1} = 1$, $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}, d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}\in\mathbb{T}$ for $b\in[2,\mathsf{rep}]$, $d_{\kappa,\alpha_\kappa}\neq d'_{\kappa,\alpha_\kappa}$, and $d_{\kappa,j} = d'_{\kappa,j} = 0$ otherwise. Denote $(\kappa_1,\ldots,\kappa_{\mathsf{rep}-2}) = [\mathsf{rep}\cdot(\kappa-1)+2, \mathsf{rep}\cdot\kappa]\backslash\{\alpha_\kappa\}$. Then, there are $\mathsf{rep}-2$ queries $(\mathbf{d}_\kappa^{(\kappa_i)})_{i\in[\mathsf{rep}-2]}$ such that $d_{\kappa,\kappa_i}^{(\kappa_i)} = 1$ and $d_{\kappa,j}^{(\kappa_j)} = 0$ for $j\in[0,\mathsf{rep}\cdot Q_\mathsf{S}]$. Let us define a matrix $\mathbf{D}_\kappa := [\mathbf{d}_\kappa \mid \mathbf{d}'_\kappa \mid \mathbf{d}_\kappa^{(\kappa_1)} \mid \ldots \mid \mathbf{d}_\kappa^{(\kappa_{\mathsf{rep}-2})}] \in \mathcal{R}_q^{(\mathsf{rep}+1)\times\mathsf{rep}}$. Let $\mathbf{c}_\kappa$ be the first row of $\mathbf{D}_\kappa$ and $\underline{\mathbf{D}}_\kappa$ be a matrix in $\mathcal{R}_q^{\mathsf{rep}\times\mathsf{rep}}$ such that $\mathbf{D}_\kappa = \begin{bmatrix}\mathbf{c}_\kappa\\ \underline{\mathbf{D}}_\kappa\end{bmatrix}$. From the above described conditions of $\mathbf{d}_\kappa$, $\mathbf{d}'_\kappa$, and $(\mathbf{d}_\kappa^{(\kappa_i)})_{i\in[\mathsf{rep}-2]}$, we have $\mathbf{c}_\kappa\in\mathcal{C}_{\mathsf{TS}}$ and $\mathbf{P}_{\mathsf{row}}^{(\kappa)}\underline{\mathbf{D}}_\kappa\in\mathcal{B}_{\mathsf{TS}}$ where $\mathbf{P}_{\mathsf{row}}^{(\kappa)}$ is a permutation matrix in $\mathcal{P}_{\mathsf{rep}}$ (see Definition 4.8 for the definition of $\mathcal{C}_{\mathsf{TS}}$ and $\mathcal{B}_{\mathsf{TS}}$). Thus, we obtain

$$\begin{bmatrix}1 & & & & \\ & \mathbf{P}_{\mathsf{row}}^{(1)} & & & \\ & & \mathbf{P}_{\mathsf{row}}^{(2)} & & \\ & & & \ddots & \\ & & & & \mathbf{P}_{\mathsf{row}}^{(Q_\mathsf{S})}\end{bmatrix}\begin{bmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_{Q_\mathsf{S}}\\ \underline{\mathbf{D}}_1 & & & \\ & \underline{\mathbf{D}}_2 & & \\ & & \ddots & \\ & & & \underline{\mathbf{D}}_{Q_\mathsf{S}}\end{bmatrix}\in\mathcal{L}_{\mathsf{TS}}.$$

Therefore, the set of $\mathsf{rep}\cdot Q_\mathsf{S}$ queries is include in $\mathcal{L}_{\mathsf{TS}}$ as desired.

To complete the proof, it remains to prove that $(\mathbf{v},\widehat{\mathbf{S}},\widehat{\mathbf{E}})$ satisfies $0<\|v_i\|_2\le B_{\mathcal{L}_{\mathsf{TS}}}$ for all $i\in[0,\mathsf{rep}\cdot Q_\mathsf{S}]$, $\|\widehat{\mathbf{S}}\|_2\le B_\mathbf{s}$, and $\|\widehat{\mathbf{E}}\|_2\le B_\mathbf{e}$. First, we show $0<\|v_i\|_2\le B_{\mathcal{L}_{\mathsf{TS}}}$ for all $i\in[0,\mathsf{rep}\cdot Q_\mathsf{S}]$. Since $c_1^*\neq c_2^*$, we have $v_0\neq 0$. For $i\in[\mathsf{rep}\cdot Q_\mathsf{S}]$, v_i is 1 or $v_0(d-d')$ where $d,d'\in\mathbb{T}$ and $d\neq d'$. If $v_i = 1$, it is clear that $v_i\neq 0$. We now suppose that $v_i = v_0(d-d') = 0$. Then, since $(d-d')$ is invertible due to Lemma 3.1, we

have $v_0 = 0$. This is contradiction. Thus, we have $v_0(d - d') \neq 0$. Therefore, $v_i \neq 0$ for $i \in [0, \mathsf{rep} \cdot Q_{\mathsf{S}}]$. Moreover, $\|v_i\|_2$ is maximized when v_i is set as in Eqs. (18) and (19). Then, from $d_{\kappa,\alpha_\kappa}, d'_{\kappa,\alpha_\kappa} \in \mathbb{T}$, we have $\|v_i\|_2 \leq \|(d_{\kappa,\alpha_\kappa} - d'_{\kappa,\alpha_\kappa}) \cdot (c_1^* - c_2^*)\|_2 \leq 2\|c_1^*\|_2 + 2\|c_2^*\|_2$. Since $\|c\|_2 = \sqrt{W}$ for any $c \in \mathcal{C}$, we have $\|v_i\|_2 \leq 4\sqrt{W} = B_{\mathcal{L}_{\mathsf{TS}}}$. Therefore, we have $0 < \|v_i\|_2 \leq B_{\mathcal{L}}$ for all $i \in [0, \mathsf{rep} \cdot Q_{\mathsf{S}}]$.

Next, we show $\|\widehat{\mathbf{S}}\|_2 \leq B_{\mathbf{s}}$. Similarly to above, $\|\widehat{\mathbf{s}}_i\|_2$ is maximized when $\widehat{\mathbf{s}}_i$ is set as in Eqs. (18) and (19). Since $d_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b}, d'_{\kappa,\mathsf{rep}\cdot(\kappa-1)+b} \in \mathbb{T}$ for $b \in [2, \mathsf{rep}]$, it suffices to focus on Eq. (18). Then, for any $\kappa \in [Q_{\mathsf{S}}]$, we have

$$\begin{aligned}\|\widehat{\mathbf{s}}_{\alpha_\kappa}\|_2 &= \|\widehat{\mathbf{s}}^* - \widehat{\mathbf{s}}'^*\|_2 \\ &= \left\|v_0 \cdot (\widetilde{\mathbf{s}}_\kappa - \widetilde{\mathbf{s}}'_\kappa) - (d_{\kappa,0} - d'_{\kappa,0}) \cdot \widehat{\mathbf{s}}_0\right\|_2 \\ &\leq \|v_0 \cdot \widetilde{\mathbf{s}}_\kappa\|_2 + \|v_0 \cdot \widetilde{\mathbf{s}}'_\kappa\|_2 + 2\|\widehat{\mathbf{s}}_0\|_2.\end{aligned}$$

First, we derive an upper bound on $\|v_0 \cdot \widetilde{\mathbf{s}}_\kappa\|_2$. From Eq. (15), we have

$$\begin{aligned}\|v_0 \cdot \widetilde{\mathbf{s}}_\kappa\|_2 &= \left\|v_0 \cdot d_{\kappa,0} \cdot \mathbf{s}_0^\star + v_0 \cdot \mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1} + v_0 \cdot d_{\kappa,\alpha_\kappa} \cdot \mathbf{s}^\star_{\alpha_\kappa}\right\|_2 \\ &\leq \|v_0 \cdot d_{\kappa,0} \cdot \mathbf{s}_0^\star\|_2 + \|v_0 \cdot \mathbf{s}^\star_{\mathsf{rep}\cdot(\kappa-1)+1}\|_2 + \|v_0 \cdot d_{\kappa,\alpha_\kappa} \cdot \mathbf{s}^\star_{\alpha_\kappa}\|_2\end{aligned}$$

From Lemma 3.2 and $\mathbf{s}_0^\star \xleftarrow{\$} \mathcal{D}_1^\ell := 2 \cdot \mathcal{D}_{\mathbf{t}}^\ell$, we have $\|v_0 \cdot d_{\kappa,0} \cdot \mathbf{s}_0^\star\|_2 \leq 2 \cdot (e^{1/4} \cdot \|v_0\|_1 \cdot \sigma_{\mathbf{t}} \cdot \sqrt{n\ell})$ with overwhelming probability. Here, recall $d_{\kappa,0} \in \mathcal{C}$. Then, because $\|c \cdot c'\|_1 \leq \|c\|_1 \cdot \|c'\|_1 = W^2$ for any $c, c' \in \mathcal{C}$, we have $\|v_0 \cdot d_{\kappa,0}\|_1 \leq 2W^2$. Namely, we have

$$\|v_0 \cdot d_{\kappa,0} \cdot \mathbf{s}_0\|_2 \leq 4e^{1/4} \cdot W^2 \cdot \sigma_{\mathbf{t}} \cdot \sqrt{n\ell}.$$

Next, since $d_{\kappa,\alpha_\kappa} \in \mathbb{T}$, the distribution of $d_{\kappa,\alpha_\kappa} \cdot \mathbf{s}_{\alpha_\kappa}$ is identical to $\mathbf{s}_{\alpha_\kappa}$. Then, from Lemma 3.2, $\mathbf{s}_i \xleftarrow{\$} \mathcal{D}_{\mathbf{w}}^\ell$ for $i \in [\mathsf{rep} \cdot Q_{\mathsf{S}}]$, and $\|v_0\|_1 \leq 2W$, we have that $\|v_0 \cdot \mathbf{s}_{\mathsf{rep}\cdot(\kappa-1)+1}\|_2$ and $\|v_0 \cdot d_{\kappa,\alpha_\kappa} \cdot \mathbf{s}_{\alpha_\kappa}\|_2$ are bounded by $2e^{1/4} \cdot W \cdot \sigma_{\mathbf{w}} \cdot \sqrt{n\ell}$. Combining the two bounds, we arrive at

$$\|v_0 \cdot \widetilde{\mathbf{s}}_\kappa\|_2 \leq 4e^{1/4} \cdot (W^2 \cdot \sigma_{\mathbf{t}} + W \cdot \sigma_{\mathbf{w}}) \cdot \sqrt{n\ell}.$$

Similarly, $\|v_0 \cdot \mathbf{s}'_\kappa\|_2$ has the same upper bound as $\|v_0 \cdot \mathbf{s}_\kappa\|_2$. We also have $\|\widehat{\mathbf{s}}_0\|_2 = \|\mathbf{z}_1^* - \mathbf{z}_2^*\|_2 \leq 2B$ since $\|(\mathbf{z}_1^*, 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_1^*} \mod q)\|_2 \leq B$ and $\|(\mathbf{z}_2^*, 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_2^*} \mod q)\|_2 \leq B$ hold. Collecting all the bounds, we obtain our desired bound:

$$\|\widehat{\mathbf{s}}_{\alpha_\kappa}\|_2 \leq 8e^{1/4} \cdot (W^2 \cdot \sigma_{\mathbf{t}} + W \cdot \sigma_{\mathbf{w}}) \cdot \sqrt{n\ell} + 4B = B_{\mathbf{s}}.$$

Finally, we show $\|\widehat{\mathbf{E}}\|_2 \leq B_{\mathbf{e}}$. Similarly to above, we only focus on $\|\widehat{\mathbf{e}}_{\alpha_\kappa}\|_2$ in Eq. (18). Then, for any $\kappa \in [Q_{\mathsf{S}}]$, we have

$$\begin{aligned}\|\widehat{\mathbf{e}}_{\alpha_\kappa}\|_2 &= \|\widehat{\mathbf{e}}^* - \widehat{\mathbf{e}}'^*\|_2 \\ &= \left\|v_0 \cdot (\widetilde{\mathbf{e}}_\kappa - \widetilde{\mathbf{e}}'_\kappa) - (d_{\kappa,0} - d'_{\kappa,0}) \cdot \widehat{\mathbf{e}}_0\right\|_2 \\ &\leq \|v_0 \cdot \mathbf{e}_\kappa\|_2 + \|v_0 \cdot \mathbf{e}'_\kappa\|_2 + 2\|\widehat{\mathbf{e}}_0\|_2.\end{aligned}$$

By an almost identical argument to $\|\widehat{\mathbf{s}}_{\alpha_\kappa}\|_2$, we have

$$\|\widehat{\mathbf{e}}_{\alpha_\kappa}\|_2 \leq 8e^{1/4} \cdot \left(W^2 \cdot \sigma_{\mathbf{t}} + W \cdot \sigma_{\mathbf{w}}\right) \cdot \sqrt{nk} + 2\|\widehat{\mathbf{e}}_0\|_2.$$

It remains to derive an upper bound on $\|\widehat{\mathbf{e}}_0\|_2$. By the definition of $\widehat{\mathbf{e}}_0$ in Eq. (11),

$$\begin{aligned}\|\widehat{\mathbf{e}}_0\|_2 &= \|2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_1^*} - 2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_2^*} - (q_{\mathsf{bot}} \cdot \boldsymbol{\delta} + \boldsymbol{\delta}_2 + c_1^* \cdot \boldsymbol{\delta}_3 + c_2^* \cdot \boldsymbol{\delta}_4) \mod q\|_2 \\ &= \|2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_1^*} \mod q\|_2 + \|2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_2^*} \mod q\|_2 + \|q_{\mathsf{bot}} \cdot \boldsymbol{\delta}\|_2 + \|\boldsymbol{\delta}_2\|_2 + \|c_1^* \cdot \boldsymbol{\delta}_3\|_2 + \|c_2^* \cdot \boldsymbol{\delta}_4\|_2,\end{aligned}$$

where we have already established that $\|\boldsymbol{\delta}_1\|_\infty \leq 2$, $\|\boldsymbol{\delta}_2\|_\infty \leq 2 \cdot (2^{\nu_{\mathbf{w}}} - 1)$, and $\|\boldsymbol{\delta}_3\|_\infty, \|\boldsymbol{\delta}_4\|_\infty \leq 2^{\nu_{\mathbf{t}}} - 1$. By the verification bound, we have $\|2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_1^*} \mod q\|_2$, $\|2^{\nu_{\mathbf{w}}} \cdot \overline{\mathbf{h}_2^*} \mod q\|_2 \leq B$. Due to our choice of q, we have $q_{\mathsf{bot}} \in [0, 2^{\nu_{\mathbf{w}}-1} - 1]$. Combining the bounds, we have

$$\|\widehat{\mathbf{e}}_0\|_2 \leq (2^{\nu_{\mathbf{w}}+2} + W \cdot 2^{\nu_{\mathbf{t}}+1}) \cdot \sqrt{nk} + 2B.$$

Collecting all the bounds, we obtain our desired bound:

$$\|\widehat{\mathbf{e}}_{\alpha_\kappa}\|_2 \leq \left(2^{\nu_{\mathbf{w}}+3} + W \cdot 2^{\nu_{\mathbf{t}}+2} + 8e^{1/4} \cdot \left(W^2 \cdot \sigma_{\mathbf{t}} + W \cdot \sigma_{\mathbf{w}}\right)\right) \cdot \sqrt{nk} + 4B = B_{\mathbf{e}}.$$

In summary, $\mathcal{B}$'s output $(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}})$ is valid a solution to the AOM-MLWE problem. Since $\mathcal{B}$ runs $\mathcal{A}$ twice, we have $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$. This completes the proof. □

7 Cryptanalysis of AOM-MLWE

We now turn to the concrete security analysis of the AOM-MLWE problem and go beyond the reductions proposed in Section 4. To do so, we aim at giving the sharpest concrete reduction to AOM-MLWE from approx-SVP, which in turn can be solved using lattice reduction algorithms and converted in a bitsec estimate.

7.1 A First Naive Attempt by Linear Algebra

Let $\mathbf{T} = \mathbf{AS} + \mathbf{E}$ be an AOM-MLWE challenge with $Q - 1$ queries available. The first intuition we might have is that from a linear algebra perspective, each of the $Q - 1$ queries removes one degree of freedom in the module rank. Hence since our problem consists in Q independent MLWE instances (or equivalently one big instance in rank $Q\ell$) we can assume that after the query, the dimension of the resulting linear algebra problem is reduced to $Q - (Q - 1) = 1$, i.e. a single MLWE_i nstance. We can very easily formalize this intuition by querying first $\mathbf{e}_1, \mathbf{s}_1$, then $\mathbf{e}_2, \mathbf{s}_2$ and so on until $\mathbf{e}_{Q-1}, \mathbf{s}_{Q-1}$. We then solve the remaining instance $\mathbf{A}\mathbf{s}_Q + \mathbf{e}_Q$ to retrieve $\mathbf{s}_Q$ and $\mathbf{e}_Q$, completing the resolution. The resulting query matrix is an identity matrix with one extra zero column. One might argue, that because of the shape of the admissible query matrices, we cannot directly query the values of $\mathbf{e}_i$ and $\mathbf{s}_i$. However, we can always do an equivalent attack as the challenge space has sufficiently many invertible elements and uses basic Gaussian elimination with pivoting to *exactly* retrieves $Q - 1$ of the secrets. However, this attack is *far* from being optimal.

7.2 Solving AOM-MLWE with Selective Queries Better than Naively

We now show that we can diffuse a bit of the final secret in each of the queries so that we can generate a statical leak in addition to the previous recovery, making the final instance of MLWE much easier. As we saw in the preliminary discussion of Section 4, the query power gives a statistical advantage in breaking the final MLWE instance. The attack we proposed in Section 7.1 fully relies on lattice reduction and completely ignores the subtlety of the choices of the queries. Whereas the attack proposed in Section 4.3.1, we fully break the scheme with only a statistical recovery. For the hardest set of parameters, we can not do so. We however show that we can combine this statistical information with standard lattice reduction arguments to do better.

7.3 A Simple Example

Let us reuse the attack we already described in Section 4.3.1 and roughly analyze it, as it will give the main intuitions on how the leakage exploitation works.

We use the (selective) queries: $\mathbf{d}_i = (1, \ldots, 0, 1, 0, \ldots, 0)^T$ where the second 1 is in position i, for i ranging from 2 to Q. As in Section 4.3.1, we then get the sets of $Q - 1$ secrets $\mathbf{s}_1 + \mathbf{s}_i$ and noises $\mathbf{e}_1 + \mathbf{e}_i$. Computing the sum of all these values yields $\tilde{\mathbf{s}}_1 = (Q - 1)\mathbf{s}_1 + \eta$ and $\tilde{\mathbf{e}}_1 = (Q - 1)\mathbf{e}_1 + \epsilon$ where $(\eta, \epsilon) = \sum_{i=2}^{Q} (\mathbf{s}_i, \mathbf{e}_i)$. As we are working above the smoothing parameter of R—the functional conditions on the standard deviation

of the error and secret in our scheme are indeed *orders of magnitude* above even the crudest estimate of the smoothing of the cyclotomic ring, we are working in—, these two variables' distribution are indistinguishable from discrete Gaussians of width $Q-1$ times larger than the width of the $\mathbf{s}_i$ and $\mathbf{e}_i$. For the sake of simplicity of exposition, let us assume that both variables $\tilde{\mathbf{s}}_1, \tilde{\mathbf{e}}_1$ are both multiple of $Q-1$. We will see in a minute that we can similarly treat the general case; but for now write $\mathbf{s}'_1 = \frac{\tilde{\mathbf{s}}_1}{Q-1} = \mathbf{s}_1 + \eta'$ and $\mathbf{e}'_1 = \frac{\tilde{\mathbf{e}}_1}{Q-1} = \mathbf{e}_1 + \epsilon'$. Now we can write a new MLWE instance derived from the sample $\mathbf{t}_1 = \mathbf{A}\mathbf{s}_1 + \mathbf{e}_1$. To do so let expand $\mathbf{s}_1, \mathbf{e}_1$ using our approximations:

$$\mathbf{t}_1 = \mathbf{A}\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{A}(\mathbf{s}'_1 - \eta') + (\mathbf{e}_1{}' - \epsilon') = (\mathbf{A}\mathbf{s}'_1 + \mathbf{e}'_1) - (\mathbf{A}\eta' + \epsilon')$$

Since $\mathbf{A}\mathbf{s}'_1 + \mathbf{e}'_1$ is known by the attacker, we can set

$$\tau := -\mathbf{t}_1 + (\mathbf{A}\mathbf{s}'_1 + \mathbf{e}'_1) = \mathbf{A}\eta' + \epsilon'.$$

This new LWE instance $(\mathbf{A}, \tau)$ is now easier than the original one, as the absolute norms of the secret and noise are smaller by a factor $\frac{1}{\sqrt{Q-1}}$.

Remark 7.1. We can directly handle the general case where the divisibility condition $Q-1 | \tilde{\mathbf{s}}_1, \tilde{\mathbf{e}}_1$ is not fulfilled. For that, it suffices to scale the instance $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{e}_1$ by the factor $Q-1$. As such, we will end with a final instance of the shape $(\mathbf{A}, \mathbf{A}\eta + \epsilon \mod q(Q-1))$, where the information/noise ratio is the same as previously. Blowing up the lattice or dividing the errors by the same factor is, indeed, conceptually the same operation.

Once $\mathbf{s}_1, \mathbf{e}_1$ are recovered from η', ϵ', simply remark that each query result $\mathbf{s}_1 + \mathbf{s}_i, \mathbf{e}_1 + \mathbf{e}_i$ yield the *exact* values of $\mathbf{s}_i, \mathbf{e}_i$ by subtracting $\mathbf{s}_1, \mathbf{e}_1$. We then recover all the secrets and errors by exact linear algebra, completing the attack as in Section 7.1.

7.4 The General Case

The generic situation is quite similar but a bit more subtle. Indeed, we need to accommodate the specific shape imposed on the query matrix. For an AOM-MLWE challenge $\mathbf{T} = \mathbf{A}\mathbf{S} + \mathbf{E}$ with $Q-1$ queries collected in a matrix $\mathbf{D}$. The blueprint of the attack goes as follows:

1. Get the matrices of query answers: $\mathsf{Ans}_\mathbf{s} = \mathbf{S}\mathbf{D}$ and $\mathsf{Ans}_\mathbf{e} = \mathbf{E}\mathbf{D}$.
2. Decompose the Queries and Extraction of $\mathbf{e}_1$: separate the first line of $\mathbf{D}$ and perform linear algebra in order to rewrite $\mathbf{e}_1$ as a known target $\mathbf{e}'_1$ plus some *controlled* noise ϵ'. This localizes $\mathbf{e}_1$ in an ellipsoid centered at $\mathbf{e}'_1$ and of known parameters with overwhelming probability. Same goes for $\mathbf{s}_1$ decomposed as $\mathbf{s}'_1$ plus some noise η'.
3. Decode $\mathbf{e}_1$ by yet another MLWE instance: craft the following challenge $\tau := -\mathbf{t}_1 + (\mathbf{A}\mathbf{s}'_1 + \mathbf{e}'_1) = \mathbf{A}\eta' + \epsilon'$. The trick here is that both η' and ϵ' are elliptically distributed, which implies isotropizing the problem to solve it classically. This step induces a distortion of the space which we can quantify finely.
4. Recover η', ϵ' from lattice reduction and use linear algebra to recover $\mathbf{s}_1, \mathbf{e}_1$ and subsequently $\mathbf{s}_i, \mathbf{e}_i$.

We now turn to these steps in more detail.

Setup. Let $\mathbf{T} = \mathbf{A}\mathbf{S} + \mathbf{E}$ the AOM-MLWE challenge, with the secret matrix $\mathbf{S} \in \mathcal{R}_q^{\ell \times Q}$ and the error matrix $\mathbf{E} \in \mathcal{R}_q^{k \times Q}$. Suppose that the query is encoded in a matrix $\mathbf{D} \in \mathcal{R}_q^{Q \times Q-1}$, so that the challenger returns the matrices of answers $\mathsf{Ans}_\mathbf{s} = \mathbf{S}\mathbf{D}$ and $\mathsf{Ans}_\mathbf{e} = \mathbf{E}\mathbf{D}$.

Decomposing the Query and Extraction of $\mathbf{e}_1$. Up to permutation of the rows—and renaming of the secrets and errors, we can assume without loss of generality that we can decompose $\mathbf{D}$ as

$$\mathbf{D} = \begin{pmatrix} \mathbf{d}^\dagger \\ \overline{\mathbf{D}} \end{pmatrix} \text{ for a vector } \mathbf{d} \in \mathcal{R}_q^{Q-1} \text{ and } \overline{\mathbf{D}} \in \mathsf{GL}_{Q-1}(\mathcal{R}_q).$$

Similarly, we decompose the errors and secrets by their first columns: $\mathbf{E} = \left(\mathbf{e}_1 \,|\, \overline{\mathbf{E}}\right)$ and $\mathbf{S} = \left(\mathbf{s}_1 \,|\, \overline{\mathbf{S}}\right)$ Using these decompositions, we have by linear algebra:

$$\mathsf{Ans}_\mathbf{s} = \mathbf{s}_1\mathbf{d}^\dagger + \overline{\mathbf{S}\mathbf{D}} \quad \text{and} \quad \mathsf{Ans}_\mathbf{e} = \mathbf{e}_1\mathbf{d}^\dagger + \overline{\mathbf{E}\mathbf{D}}.$$

We now have an explicit dependency on $\mathbf{s}_1$ and $\mathbf{e}_1$, but only through the multiplication by $\mathbf{d}^\dagger$. To make this latter term disappear we use the pseudo-inversion trick: multiplying on the right by $\mathbf{d}$, then dividing by the totally positive element $\mathbf{d}^\dagger\mathbf{d}$ gives:

$$\mathbf{s}_1 = \underbrace{\frac{\mathsf{Ans}_\mathbf{s}\mathbf{d}}{\mathbf{d}^\dagger\mathbf{d}}}_{:=\mathbf{s}'_1} - \underbrace{\frac{\overline{\mathbf{S}\mathbf{D}}\mathbf{d}}{\mathbf{d}^\dagger\mathbf{d}}}_{:=\eta} \quad \text{and} \quad \mathbf{e}_1 = \underbrace{\frac{\mathsf{Ans}_\mathbf{e}\mathbf{d}}{\mathbf{d}^\dagger\mathbf{d}}}_{:=\mathbf{e}'_1} - \underbrace{\frac{\overline{\mathbf{E}\mathbf{D}}\mathbf{d}}{\mathbf{d}^\dagger\mathbf{d}}}_{:=\epsilon}. \tag{20}$$

Exploiting the MLWE Structure with the Leakage. We now assume—exactly as we did first in the example—that the vectors $\mathbf{s}'_1, \mathbf{e}'_1$ are of integral coefficients. We now have derived a non-trivial *statstical* information: $\mathbf{e}_1$ and $\mathbf{s}_1$ are random variables centered respectively at $\mathbf{e}'_1$ and $\mathbf{s}'_1$ and with covariances $\sigma_\mathbf{e}^2\Sigma^\dagger\Sigma$ and $\sigma_\mathbf{s}^2\Sigma^\dagger\Sigma$ where $\Sigma = \left(\frac{\overline{\mathbf{D}}\mathbf{d}^\dagger}{\mathbf{d}^\dagger\mathbf{d}}\right)$.

This translates directly into a piece of geometric information as we can construct a new corresponding MLWE instance corresponding to the decoding of ϵ and η:

$$\tau := \mathbf{t}_1 - (\mathbf{A}\mathbf{s}'_1 + \mathbf{e}'_1) = \mathbf{A}\eta + \epsilon.$$

In the general case, we can mimic the technique of multiplying by $Q-1$ done in Remark 7.1, this time by multiplying every quantity by $\mathbf{d}^\dagger\mathbf{d}$ and end up with an instance of the shape $\left(\mathbf{A}, \mathbf{A}\eta + \epsilon \mod (\mathbf{d}^\dagger\mathbf{d})\right)$. The modulus is now elliptic in that $\mathbf{d}^\dagger\mathbf{d}$ is not a rational element. This is not an issue, as it amounts to pursuing the attack in a twisted norm. This approach is standard and is for instance detailed in [ENS+23]. Dividing η and ϵ by $\mathbf{d}^\dagger\mathbf{d}$ leads to an LWE instance for the standard norm with elliptic Gaussians distributions whereas multiplying the initial instance by $\mathbf{d}^\dagger\mathbf{d}$ yields an LWE instance in elliptic norm but with spherical Gaussians. Once again these are the two faces of the same coin and both approaches give the same results.

Solving Approximate Elliptic Secret/Noise MLWE. Let us explain how to solve such an LWE with a distribution of the resulting secret/error being elliptic instead of spherical. As it is now the case in *all* lattice-based schemes based on structured lattices, we will do a *leap of faith* and assume that the security of MLWE is the same as the security of the lattice problem obtained when descending over $\mathbb{Z}$.

To solve this new instance, we will use a reduction to unique-SVP through so-called distorted-BDD, as done for instance in [DDGR20]. The core trick is to first embed the MLWE instance into a module lattice of rank $\ell \cdot n + 1$. Let $\Lambda = \{(\mathbf{x}, \mathbf{y}, w) \,|\, \mathbf{x} + \mathbf{A} \cdot \mathbf{y} - \tau w = 0 \pmod q\}$ where $\mathbf{A}$ is overloaded to also denote the anticirculant matrix corresponding to the multiplication endomorphism in $\mathcal{R}_q^\ell$ by the matrix $\mathbf{A}$. A basis of this lattice is given by

$$\begin{pmatrix} q\mathbf{I}_{\ell n} & 0 & 0 \\ \mathbf{A} & -\mathbf{I}_{kn} & 0 \\ \tau & 0 & \mathbf{I}_n \end{pmatrix}.$$

Now remark that the vector[13] $(\mathbf{e}, \mathbf{s}, 1)$ belong to the lattice Λ, but that any short vector of the shape $(\mathbf{e}', \mathbf{s}', \mathbf{v})$ for small enough $\mathbf{v}$ will also be a solution to the approximate problem, with the relaxation by $\mathbf{v}$. We can be

[13] for the sake of notational simplicity, we denote the vectors of the modules $\mathcal{R}_q^\ell$, $\mathcal{R}_q^k$ and their descent over $\mathbb{Z}$ by the same symbol.

even more precise and remark that with overwhelming probability this vector belongs to the intersection of the ellipsoid defined by the symmetric block-diagonal semi-definite positive matrix:

$$S = \begin{pmatrix} \sigma_e \Sigma & 0 & 0 \\ 0 & \sigma_s \Sigma & 0 \\ 0 & 0 & \mathbf{I}_n \end{pmatrix},$$

that is to say the set $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^{(k+\ell+1)n} \mid \langle \mathbf{x}, S^{-T} S^{-1} \mathbf{x} \rangle \leq \sqrt{d\ell}\}$.

It then suffices to apply the matrix S^{-1} on Λ to re-isotropize the problem and reduce it to a usual spherical DBDD instance.

Solving the Final approx-SVP Instance. From this point, we can apply lattice reduction on this lattice $\Lambda^* = S^{-1}\Lambda$ to retrieve short vectors. To do so, we rely on the DBKZ algorithm, which achieves the best time/quality trade-offs in the literature. Let us do a very brief recall on the output guarantees of this algorithm.

Modelization of the Output of Reduced Bases. For the sake of clarity in the following explanations, we adopt the "Geometric series assumption" (GSA). This assumption states that the norm of the Gram-Schmidt vectors of a reduced basis decrease with a geometric decay. Specifically, in the context of the self-dual Block Korkine-Zolotarev (DBKZ) reduction algorithm proposed by Micciancio and Walter [MW16], the GSA can be instantiated as follows. Suppose we have an output basis $(\mathbf{b}_i)_{i\in[n]}$ obtained from the DBKZ algorithm with a block size denoted as β, applied to a lattice Λ of rank n. Then, the following equation holds for the i-th Gram-Schmidt vector $\mathbf{b}_i^*$ of the basis:

$$\|\mathbf{b}_i^*\| = \gamma^{n-2(i-1)} \mathrm{vol}(\Lambda)^{\frac{1}{n}}, \quad \text{where} \quad \gamma_\beta = \left(\frac{(\pi\beta)^{\frac{1}{\beta}} \cdot \beta}{2\pi e} \right)^{\frac{1}{2(\beta-1)}},$$

for $\mathbf{b}_i^*$ being the i-th Gram Schmidt vector of the basis. In particular, this implies that the first vector of the output basis is satisfying the relation:

$$\|\mathbf{b}_1\| \leq \gamma^n \mathrm{vol}(\Lambda)^{\frac{1}{n}}.$$

To get a finer estimate, when computing the actual figures this analysis can be refined by using the probabilistic simulation of [DDGR20] rather than this coarser GSA-based model to determine the BKZ blocksize β for a successful attack. This helps to take into account the well-known *quadratic tail* phenomenon of reduced bases [YD17].

Solving uSVP. To retrieve vectors of size comparable to $\|(\mathbf{e}^*, \mathbf{s}^*)\|$, we therefore need to select a blocksize β such that:

$$\sqrt{\frac{\beta}{(k+\ell+1)n}} \|(\mathbf{e}_1, \mathbf{s}_1)\| \leq \gamma_\beta^{(k+\ell+1)n} \mathrm{vol}(\Lambda^*)^{\frac{1}{(k+\ell+1)n}}.$$

Conveniently, during the isotropization step, we rescaled the target secret vector to be normally distributed, so that with overwhelming probability we have $\|(\mathbf{e}^*, \mathbf{s}^*)\| \leq \sqrt{\ell d}$. Further, we also know the volume of Λ^* since by multilinearity of the determinant we get $\mathrm{vol}(\Lambda^*) = \mathrm{vol}(\Lambda)\det(S) = q^{\ell d}\det(S)$. All in all, we then seek for the minimal β such that:

$$\sqrt{\beta} \leq \gamma_\beta^{(k+\ell+1)n} \det(S)^{\frac{1}{(k+\ell+1)n}}.$$

7.5 Concrete Model of Lattice Reduction

Before delving into the concrete parameter selection, we need to devise a sound method to convert the complexity of the resulting MLWE instance, constructed through the previous reduction to actual figures representing the bit-security of the problem.

The Core-SVP Hardness. Indeed, to accurately assess the hardness of the underlying problems and ensure a specified level of bit-security, it is necessary to establish a model that simulates the behavior of a practical oracle for Approximate shortest Vector Problem (SVP). This modeling is crucial since our hard problems involve the identification of relatively short vectors in various lattices. To achieve this, we will employ the celebrated (self-dual) Block Korkine-Zolotarev (BKZ) algorithm. Specifically, the BKZ algorithm with a block size denoted by β necessitates a polynomial number of calls to an SVP oracle in dimension β, with a heuristically expected number of calls that are approximately linear—with some implementation tricks.

To account for potential future advancements in this reduction method, we will only consider the cost of a single call to the SVP oracle. This approach, known as *core-SVP hardness*, entails a highly conservative estimation. This cautionary measure is warranted by the possibility of cost amortization for SVP calls within BKZ, particularly when sieving is employed as the SVP oracle. Notably, sieving has become the prevailing standard for larger block sizes, as exemplified in [ADH+19].

From Lattice Reduction Block-Size to Concrete Bitsec. This analysis translates into concrete bit-security estimates following the methodology of NewHope [ADPS16] (so-called "core-SVP methodology"). In this model, the bit complexity of lattice sieving (which is asymptotically the best SVP oracle) is taken as $\lfloor 0.292\beta \rfloor$ in the classical setting [BDGL16] and $\lfloor 0.257\beta \rfloor$ in the quantum setting [CL21] in blocksize β.

8 Parameters Selection

We now turn to parameter selection. Classically for parameter selection of lattice scheme, we rely on the so-called *Core-SVP* methodology to convert lattice reduction blocksize into concrete bitsecurity. We provide a complete overview of this in Section 7.5. The security of our two-round threshold scheme is evaluated against forgery and key recovery/pseudo-randomness of the verification key. We point out that the practical bounds are not directly based on the advantage bounds given in the asymptotic security proof (e.g., we ignore the loss from the forking lemma, and we devise a more direct reduction for $\mathsf{AOM\text{-}MLWE}$). However we still *do enforce* that all the *functional* constraints between parameters are satisfied. This is common practice when dealing with practical security, as epitomized in the parameter selection process of the NIST standards ML-DSA (Dilithium)[LDK+22], FN-DSA(Falcon)[PFH+22], or the recent signatures [EFG+22, dPEK+23].

8.1 Direct Forgery Resilience

8.1.1 On SelfTargetMSIS.

A direct forgery can be done by reverse-engineering a signature from the verification process. More precisely this amount to solve the following problem is usually coined $\mathsf{SelfTargetMSIS}$ assumption. Find a vector $\mathbf{z}_{\mathsf{sol}}$ such that:

$$\left(\mathbf{z}_{\mathsf{sol}} = \begin{bmatrix} c \\ \mathbf{z}' \end{bmatrix}\right) \wedge (\|\mathbf{z}_{\mathsf{sol}}\|_2 \leq B) \wedge \quad \mathsf{H}\left(\left[-\widehat{\mathsf{vk}} \mid \mathbf{A} \mid \mathbf{I}\right] \cdot \mathbf{z}_{\mathsf{sol}},\ \mathsf{M}\right) = c\Big], \tag{21}$$

B being set from the correctness condition of Section 6.1. Following [LDK+22, §C.3], we assume that the best way to solve (21) is either by (i) breaking the second preimage resistance of H or by (ii) generating $\mathbf{w}$ at random, computing $c = \mathsf{H}(\mathbf{w},\ \mathsf{M})$, and finally solving the inhomogenous SIS instance:

$$\left(\left[\mathbf{A} \mid \mathbf{I}\right] \cdot \mathbf{z}' = \mathbf{w} - c \cdot \mathsf{vk}\right) \wedge (\|\mathbf{z}'\| \leq B - W)\,. \tag{22}$$

Solving The Inhomogeneous MSIS instance Eq. (22). We can cast it as finding $\bar{\mathbf{z}}$ at a bounded distance from the point $\mathbf{v} = \mathbf{w} - c \cdot \mathsf{vk}$. This BDD problem can be solved using the so-called *Nearest-Cospace* framework of Espitau and Kirchner [EK20], which states that under the GSA, the decoding can be done in

time $\mathsf{poly}(n)$ calls to a CVP oracle in dimension β, as long as[14]: $\|\bar{\mathbf{z}} - \mathbf{v}\| \leq \min_{x \leq \ell n} \left(\gamma^{(k+\ell)n-x} q^{\frac{k\,n}{(k+\ell)n-x}}\right)$ As such, we need to enforce the following conditions:

$$B \leq \min_{\ell n \leq m \leq (k+\ell)n} \left(\gamma^m q^{\frac{k\,n}{m}}\right).$$

Challenge Space. We need the hash function H to be second preimage resistant. To guarantee this we ensure that $|\mathcal{C}| > 2^\lambda$. Considering how $\mathcal{C}$ is defined in Section 3.2 it is enough to set W such that: $\binom{n}{W} \cdot 2^W \geq 2^\lambda$.

8.2 Breaking Unforgeability

For the second attack, we will follow the security reduction done in Section 6, which quantifies the reduction to the $\mathsf{AOM\text{-}MLWE}$ problem and amounts to perform a key recovery using the maximal number of signature queries to collect as much information as we can. Recall from the security proof that the advantage of breaking the unforgeability game is expressed as:

$$\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\mathsf{2\text{-}round}},\mathcal{A}}(1^\lambda, N, T) \leq \sqrt{Q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{AOM\text{-}MLWE}}_{\mathcal{B}}(1^\lambda)} + N^2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(1^\lambda) + \frac{Q_{\mathsf{S}}}{2^{2\cdot\lambda}} + \mathsf{negl}(\lambda),$$

the $\mathsf{negl}(\lambda)$ term being indeed negligible when the set of relations between coefficients given in Section 6 are satisfied. We normalize the cost by the global number of queries, that is to say Q_{RO} when computing the bit-security:

$$\lambda_{\text{REAL}} = \log_2(\mathsf{Time}(\mathcal{A})/\mathsf{Adv}^{\mathsf{ts\text{-}uf}}_{\mathsf{TS}_{\mathsf{2\text{-}round}},\mathcal{A}}(1^\lambda, N, T)), \tag{23}$$

where the running time of the adversary satisfies $\mathsf{Time}(\mathcal{A}) \geq Q_{\mathsf{RO}}$. Our goal is then to ensure $\lambda_{\text{REAL}} \leq \lambda + O(1)$. The term in $N^2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(1^\lambda)$ is itself exponentially small in λ when normalized by $\mathsf{Time}(\mathcal{A})$ and the term $Q_S 2^{-2\lambda}$ is itself smaller than $2^{-\lambda}$, so that we only care about the first term. As typical with practice-oriented schemes (e.g., [Sch90, FKP16, BKV19]), we treat the advantage by ignoring the square root induced by the forking lemma. To do so we rely on the worst-case analysis of Section 7 to reduce the problem to a single MLWE instance in dimension $d\ell$. In order to evaluate the cost of this remaining MLWE, we need to analyze the shape of the admissible queries in fine-grained way.

8.2.1 Shape Of The Queries

We described the attack for any query matrix $\mathbf{D}$. Let us now restrict ourselves to the matrices of the form prescribed by the definition of the scheme (see Section 4.5 for the detailed definition) and study the best cases scenarii (for the attacker). As the permutation matrices don't affect the geometry of the attack, we can, without loss of generality only study the queries stemming from the set:

$$\mathcal{L}'_{\mathsf{TS}} = \left\{ \begin{bmatrix} \mathbf{c}_1^\top & \mathbf{c}_2^\top & \cdots & \mathbf{c}_{Q_{\mathsf{S}}}^\top \\ \mathbf{B}_1 & & & \\ & \mathbf{B}_2 & & \\ & & \ddots & \\ & & & \mathbf{B}_{Q_{\mathsf{S}}} \end{bmatrix} \subset \mathcal{R}_q^{Q \times (Q-1)} \;\middle|\; \forall i \in [Q_{\mathsf{S}}], (\mathbf{c}_i, \mathbf{B}_i) \in \mathcal{C}_{\mathsf{TS}} \times \mathcal{B}_{\mathsf{TS}}, \right\}.$$

with

$$\mathcal{B}_{\mathsf{TS}} = \left\{ \begin{bmatrix} 1 & 1 & & \\ b_1 & b'_1 & & \\ b_2 & b'_2 & 1 & \\ \vdots & \vdots & & \ddots \\ b_{\tau-1} & b'_{\tau-1} & & 1 \end{bmatrix} \in \mathcal{R}_q^{\tau \times \tau} \;\middle|\; \begin{array}{c} \forall i \in [\tau - 1], (b_i, b'_i) \in \mathbb{T}^2 \\ \wedge\; b_1 \neq b'_1 \end{array} \right\},$$

[14] This equation encompass a common optimization consisting of dropping the final columns of the basis to leverage the slight variation in the volume it can induce

and $\mathcal{C}_{\mathsf{TS}} = \left\{[c, c', 0, \cdots, 0]^\top \in \mathcal{R}_q^\tau \mid c, c' \in \mathcal{C} \cup \{0\}\right\}$.

Size of $\mathbf{d}^\dagger\mathbf{d}$. This latter set will be mainly driving the efficiency of the attack as the vector $\mathbf{d}$ is now a vector of length Q, consisting of $Q - 2Q_{\mathsf{S}}$ 0 and $2Q_{\mathsf{S}}$ coefficients taken in $\mathcal{C}$. This means in particular that $\mathbf{d}^\dagger\mathbf{d}$ is a sum of $2Q_{\mathsf{S}}$ arbitrary terms in the set $\mathcal{C}^2 = \{\mathbf{c}^\dagger\mathbf{c} \,|\, \mathbf{c} \in \mathcal{C}\}$. Remark that Q_{S} is exceedingly large compared to the rest of the parameters (essentially exponential in the security parameter while other quantities are chosen to be polynomial in it in practice) so that the coefficients of $\mathbf{d}^\dagger\mathbf{d}$ will be extremely concentrated. More precisely, its constant coefficient will be exactly WQ_{S} by construction and the other coefficients will behave as discrete Gaussian elements of standard deviation in $O(\sqrt{Q_{\mathsf{S}}})$ by central limit theorem. As such, this element is *almost* a relative integer, in the sense that its complex embeddings will be concentrated around the constant coefficient WQ_{S}. In particular, its inverse will show the same concentration at the value $1/(WQ_{\mathsf{S}})$. Hence, conservatively, we will treat this value as being exactly the rational $1/(WQ_{\mathsf{S}})$

On the shape of the $\overline{\mathbf{D}}$ block. Now for the term corresponding to $\overline{\mathbf{D}}$, remark that identity matrices do not belong to the set $\mathcal{B}_{\mathsf{TS}}$. However, from the shape of the matrix Σ, we can see that the best case (from the attacker perspective), that is to say inducing the least distortion, is realized when the matrix $\overline{\mathbf{D}}$ is as close as possible form the identity. Hence, in a conservative thought, we will assume without loss of security that $\overline{\mathbf{D}}$ is exactly an identity.

8.2.2 Putting everything together

Hence, our analysis reveals that, practically speaking, the secret/vectors of the resulting MLWE instance are $\sqrt{WQ_{\mathsf{S}}}$ times smaller than for the original parameters of the AOM-MLWE challenge itself, meaning that an an attacker would need to solve an $\mathsf{MLWE}_{n,q,k,\ell,\frac{\sigma}{\sqrt{W\cdot Q_{\mathsf{S}}}}}$ type instance. Hence we seek for a set of parameters that are secure even reduced by this large factor. To do so, we can rely on the extensive literature on the cryptanalysis of MLWE. To our knowledge, the state-of-the-art for estimating the concrete hardness of MLWE is the so-called lattice estimator (https://github.com/malb/lattice-estimator). According to this estimator on our tentative parameters, the best-known attacks are the primal uSVP attack by Alkim et al. [ADPS16] and the dual/hybrid attack by Espitau et al. [EJK20] all in all, combined with the so-called dimension for free trick of [Duc18].

8.3 Parameter Sets

Despite the apparently large number of variables, parameters can be set in a systematic way and we can devise an optimization tool to explore the parameter space and find the signatures with the smallest size/communication complexity while still achieving the desired security guarantees. The results of our exploration are collected in Table 2, targeting NIST levels I, III, and V of security, with supporting roughly 2^{60} queries of signatures before being endangered. Moreover, all of them support a threshold of up to 1024 participants, which is the upper limit of the "large" requirements of the NIST preliminary call for threshold. Remarkably, our 2 round signatures are practical with aggregated signature sizes lower than 11KiB. The main overhead in its use is in the offline phase where signers must exchange the tokens, which are of size a couple of hundred kilobytes. It is worth highlighting that if we consider a model where the aggregator stores the preprocessing tokens, the individual signers do not need to include $\mathbf{w}_i$ in the partial signature $\widehat{\mathsf{sig}}_i$. In this case, the online communication per user becomes much smaller: 14KB, 19KB, and 22KB for NIST levels I, III, and V, respectively.

Acknowledgement. This work has been supported in part by JST CREST Grant Number JPMJCR22M1, JST-AIP Acceleration Research JPMJCR22U5, JSPS KAKENHI Grant Numbers JP22KJ1366.

Table 2: All sizes are given in KB for a maximum T of 1024 and Q_S being upper bounded by 2^{59}. on, off refers to the communication cost *per user* in the online/offline phase. The online cost is written $XX(YY)$ for XX being the size of the optimized version where the tokens are already processed by the aggregator and YY being the naive scheme where the tokens are transmitted at an online phase. The corresponding security is given in bits: Sec F for the forgery and Sec K for the key recovery respectively.

Sec (F/K)	$\lfloor \log q \rceil$	$\log \sigma_{\mathbf{t}}$	$\log \sigma_{\mathbf{w}}$	$\nu_{\mathbf{t}}$	$\nu_{\mathbf{w}}$	n	ℓ	k	W	|vk|	|Sig|	on/usr	off/usr
128 / 146	50	5	34.5	38	38	256	9	11	23	5.5	10.8	14.1 (276)	262
192 / 192	50	10	35	34	38	512	6	7	31	7	14.5	19 (461)	442
256 / 282	51	15	37	35	40	512	7	10	44	9.5	18	22 (853)	831

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May / June 2010.

[ABV+12] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 280–297. Springer, Heidelberg, May 2012.

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

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

[ADH+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 717–746. Springer, Heidelberg, May 2019.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

[AKSY22] Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 39–53. ACM Press, November 2022.

[AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 519–548, Virtual Event, August 2021. Springer, Heidelberg.

[ASY22] Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Round-optimal lattice-based threshold signatures, revisited. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *ICALP 2022*, volume 229 of *LIPIcs*, pages 8:1–8:20. Schloss Dagstuhl, July 2022.

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 551–572. Springer, Heidelberg, December 2014.

[BCK+22] Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 517–550. Springer, Heidelberg, August 2022.

[BD22] LTAN Brandão and Michael Davidson. Notes on threshold eddsa/schnorr signatures. National Institute of Standards and Technology, 2022. https://doi.org/10.6028/NIST.IR.8214B.ipd.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th SODA*, pages 10–24. ACM-SIAM, January 2016.

[BFP21] Balthazar Bauer, Georg Fuchsbauer, and Antoine Plouviez. The one-more discrete logarithm assumption in the generic group model. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 587–617. Springer, Heidelberg, December 2021.

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

[BGG+18] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 565–596. Springer, Heidelberg, August 2018.

[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.

[BKP13] Rikke Bendlin, Sara Krehbiel, and Chris Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13*, volume 7954 of *LNCS*, pages 218–236. Springer, Heidelberg, June 2013.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.

[BL22] Renas Bacho and Julian Loss. On the adaptive security of the threshold BLS signature scheme. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 193–207. ACM Press, November 2022.

[BLL+15] Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Heidelberg, November / December 2015.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Heidelberg, August 2013.

[BLR+18] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, April 2018.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Heidelberg, December 2001.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Heidelberg, August 2019.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 71–87. Springer, Heidelberg, April 2008.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.

[BNPS02] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In Paul F. Syverson, editor, *FC 2001*, volume 2339 of *LNCS*, pages 319–338. Springer, Heidelberg, February 2002.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, January 2003.

[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, Heidelberg, August 2022.

[CATZ24] Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Partially non-interactive two-round lattice-based threshold signatures. Cryptology ePrint Archive, Paper 2024/467, 2024. https://eprint.iacr.org/2024/467.

[CCL+20] Guilhem Castagnos, Dario Catalano, Fabien Laguillaumie, Federico Savasta, and Ida Tucker. Bandwidth-efficient threshold EC-DSA. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 266–296. Springer, Heidelberg, May 2020.

[CGG+20] Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1769–1787. ACM Press, November 2020.

[CGRS23] Hien Chu, Paul Gerhart, Tim Ruffing, and Dominique Schröder. Practical Schnorr threshold signatures without the algebraic group model. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 743–773. Springer, Heidelberg, August 2023.

[Che23] Yanbo Chen. DualMS: Efficient lattice-based two-round multi-signature with trapdoor-free simulation. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 716–747. Springer, Heidelberg, August 2023.

[CKM23a] Elizabeth Crites, Chelsea Komlo, and Mary Maller. Fully adaptive schnorr threshold signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 678–709, Cham, 2023. Springer Nature Switzerland.

[CKM23b] Elizabeth C. Crites, Chelsea Komlo, and Mary Maller. Fully adaptive Schnorr threshold signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 678–709. Springer, Heidelberg, August 2023.

[CL21] André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2021.

[CS20] Daniele Cozzo and Nigel P. Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 169–186. Springer, Heidelberg, 2020.

[Csi63] Imre Csiszár. Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten. *Magyar. Tud. Akad. Mat. Kutató Int. Közl*, 8:85–108, 1963.

[CSS+22] Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. Efficient threshold FHE with application to real-time systems. Cryptology ePrint Archive, Report 2022/1625, 2022. https://eprint.iacr.org/2022/1625.

[DDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 329–358. Springer, Heidelberg, August 2020.

[Des90] Yvo Desmedt. Abuses in cryptography and how to fight them. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 375–389. Springer, Heidelberg, August 1990.

[DF90] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, Heidelberg, August 1990.

[DJN+20] Ivan Damgård, Thomas Pelle Jakobsen, Jesper Buus Nielsen, Jakob Illeborg Pagter, and Michael Bæksvang Østergaard. Fast threshold ECDSA with honest majority. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 382–400. Springer, Heidelberg, September 2020.

[DKLs19] Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy*, pages 1051–1066. IEEE Computer Society Press, May 2019.

[DLN+21] Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 659–690. Springer, Heidelberg, May 2021.

[DM20] Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Heidelberg, May 2020.

[DOK+20] Anders P. K. Dalskov, Claudio Orlandi, Marcel Keller, Kris Shrishak, and Haya Shulman. Securing DNSSEC keys via threshold ECDSA from generic MPC. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *ESORICS 2020, Part II*, volume 12309 of *LNCS*, pages 654–673. Springer, Heidelberg, September 2020.

[DOTT21] Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 99–130. Springer, Heidelberg, May 2021.

[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.

[dPEK+23] Rafaël del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Raccoon. Technical report, National Institute of Standards and Technology, 2023. Available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

[Duc18] Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 125–145. Springer, Heidelberg, April / May 2018.

[EFG+22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 222–253. Springer, Heidelberg, May / June 2022.

[EJK20] Thomas Espitau, Antoine Joux, and Natalia Kharchenko. On a dual/hybrid approach to small secret LWE - A dual/enumeration technique for learning with errors and application to security estimates of FHE schemes. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 440–462. Springer, Heidelberg, December 2020.

[EK18] Ali El Kaafarani and Shuichi Katsumata. Attribute-based signatures for unbounded circuits in the ROM and efficient instantiations from lattices. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 89–119. Springer, Heidelberg, March 2018.

[EK20] Thomas Espitau and Paul Kirchner. The nearest-colattice algorithm. Cryptology ePrint Archive, Report 2020/694, 2020. https://eprint.iacr.org/2020/694.

[EKT24] Thomas Espitau, Shuichi Katsumata, and Kaoru Takemure. Two-round threshold signature from algebraic one-more learning with errors, 2024. To Appear in CRYPTO 2024. Available at https://eprint.iacr.org/2024/496.

[ENP24] Thomas Espitau, Guilhem Niot, , and Thomas Prest. Flood and submerse: Verifiable short secret sharing and application to robust threshold signatures on lattices, 2024. To Appear in CRYPTO 2024.

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, December 2020.

[ENS+23] Thomas Espitau, Thi Thu Quyen Nguyen, Chao Sun, Mehdi Tibouchi, and Alexandre Wallet. Antrag: Annular NTRU trapdoor generation - making mitaka as secure as falcon. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VII*, volume 14444 of *LNCS*, pages 3–36. Springer, Heidelberg, December 2023.

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

[FKP16] Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the provable security of (EC)DSA signatures. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1651–1662. ACM Press, October 2016.

[FSZ22] Nils Fleischhacker, Mark Simkin, and Zhenfei Zhang. Squirrel: Efficient synchronized multi-signatures from lattices. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1109–1123. ACM Press, November 2022.

[GG18] Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1179–1194. ACM Press, October 2018.

[GJKR07] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, January 2007.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240. Tsinghua University Press, 2010.

[GKS23] Kamil Doruk Gur, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice signatures from threshold homomorphic encryption, 2023. To Appear in PQCrypto 2024. Available at https://eprint.iacr.org/2023/1318.

[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, Heidelberg, May 2020.

[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.

[GV15] Sergey Gorbunov and Dhinakaran Vinayagamurthy. Riding on asymmetry: Efficient ABE for branching programs. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 550–574. Springer, Heidelberg, November / December 2015.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, August 2015.

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 469–477. ACM Press, June 2015.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[KCLM22] Irakliy Khaburzaniya, Konstantinos Chalkias, Kevin Lewi, and Harjasleen Malvai. Aggregating and thresholdizing hash-based signatures using STARKs. In Yuji Suga, Kouichi Sakurai, Xuhua Ding, and Kazue Sako, editors, *ASIACCS 22*, pages 393–407. ACM Press, May / June 2022.

[KG20] Chelsea Komlo and Ian Goldberg. FROST: Flexible round-optimized Schnorr threshold signatures. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 34–65. Springer, Heidelberg, October 2020.

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

[KRT24] Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from mlwe/msis and dl with rewinding, 2024. To Appear in CRYPTO 2024.

[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.

[Lin22] Yehuda Lindell. Simple three-round multiparty schnorr signing with full simulatability. Cryptology ePrint Archive, Report 2022/374, 2022. https://eprint.iacr.org/2022/374.

[LLL22] Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 680–710. Springer, Heidelberg, November 2022.

[LN18] Yehuda Lindell and Ariel Nof. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1837–1854. ACM Press, October 2018.

[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, Heidelberg, May 2013.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, Heidelberg, May 2014.

[LST18] Benoît Libert, Damien Stehlé, and Radu Titiu. Adaptively secure distributed PRFs from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 391–421. Springer, Heidelberg, November 2018.

[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, Heidelberg, December 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.

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

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 820–849. Springer, Heidelberg, May 2016.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003.

[NIS22] NIST. Call for additional digital signature schemes for the post-quantum cryptography standardization process. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf, 2022.

[NRS21] Jonas Nick, Tim Ruffing, and Yannick Seurin. MuSig2: Simple two-round Schnorr multi-signatures. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 189–221, Virtual Event, August 2021. Springer, Heidelberg.

[PB23] René Peralta and Luís T.A.N. Brandão. Nist first call for multi-party threshold schemes. National Institute of Standards and Technology, 2023. https://doi.org/10.6028/NIST.IR.8214C.ipd.

[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.

[PKM+24] Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248. Springer, Heidelberg, May 2024.

[Pre17] Thomas Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 347–374. Springer, Heidelberg, December 2017.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

[QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th FOCS*, pages 859–870. IEEE Computer Society Press, October 2018.

[Rén61] Alfréd Rényi. On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, Berkeley, Calif., 1961. University of California Press.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

[Sha79a] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sha79b] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

[SS01] Douglas R. Stinson and Reto Strobl. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. In Vijay Varadharajan and Yi Mu, editors, *ACISP 01*, volume 2119 of *LNCS*, pages 417–434. Springer, Heidelberg, July 2001.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Heidelberg, August 2022.

[TZ23] Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Heidelberg, April 2023.

[vEH14] Tim van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. Information Theory*, 60(7):3797–3820, 2014.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Heidelberg, May / June 2022.

[WW23] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 385–416. Springer, Heidelberg, April 2023.

[YD17] Yang Yu and Léo Ducas. Second order statistical behavior of LLL and BKZ. In Carlisle Adams and Jan Camenisch, editors, *SAC 2017*, volume 10719 of *LNCS*, pages 3–22. Springer, Heidelberg, August 2017.

A Visual Aid for Row and Column Masks

Here, we provide slightly more intuition behind the idea of [PKM+24]. Fig. 15 is taken from [PKM+24] for the sake of aiding our explanation. For $i, j \in \mathsf{SS}$, let $\mathbf{m}_{i,j} = \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$ and let HS (resp. CS) be the set of honest (resp. corrupt) signers in the signer set SS. In Fig. 15, $\mathsf{SS} = \{1, 2, 3, 4, 5\}$, $\mathsf{HS} = \{4, 5\}$, and $\mathsf{CS} = \{1, 2, 3\}$.

Let us focus on the column masks $(\mathbf{m}^*_j)_{j\in\mathsf{HS}}$ used to construct the responses $\left(\mathbf{z}_j = c{\cdot}L_{\mathsf{SS},j}{\cdot}\mathbf{s}_j + \mathbf{r}_j + \mathbf{m}^*_j\right)_{j\in\mathsf{HS}}$ to form an intuition of the security proof. The proof hinges on the key observation that while the individual

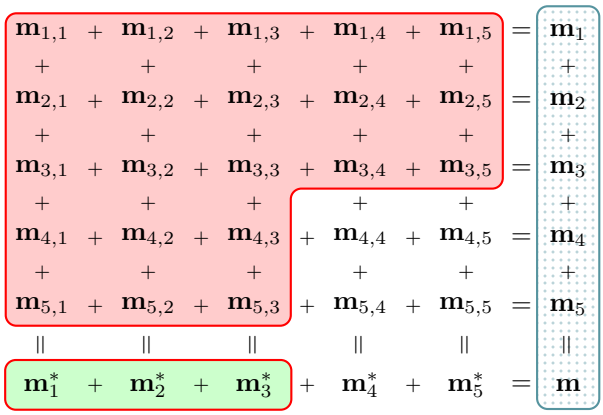

Figure 15: Figure taken from [PKM+24]. Relationships between the individual masks $\mathbf{m}_{i,j}$, the row masks $\mathbf{m}_i$, and column masks $\mathbf{m}^*_j$.

- The row masks $\mathbf{m}_i$ (blue, dotted pattern) are all public.
- An adversary corrupting the user set $\{1, 2, 3\}$ learns the set $(\mathbf{m}_{i,j})_{\min(i,j)\leq 3}$ and can infer the column masks $(\mathbf{m}^*_j)_{j\leq 3}$ (red).

row masks $(\mathbf{m}_j)_{j\in\mathsf{HS}}$ are known to the adversary, the only knowledge the adversary gains on the column masks $(\mathbf{m}^*_j)_{j\in\mathsf{HS}}$ are their sum $\sum_{j\in\mathsf{HS}}\mathbf{m}^*_j$; put differently, $(\mathbf{m}^*_j)_{j\in\mathsf{HS}}$ are distributed randomly, conditioned on their sum being $\sum_{j\in\mathsf{HS}}\mathbf{m}^*_j$. This can be checked from the fact that $\mathbf{m}^*_i$ can be written as $\sum_{j\in\mathsf{HS}}\mathbf{m}_{i,j}+\sum_{j\in\mathsf{CS}}\mathbf{m}_{i,j}$, where each $\mathbf{m}_{i,j}$ for $i,j\in\mathsf{HS}$ remains random to the adversary since $\mathsf{seed}_{i,j}$ is only known to the honest signers i and j. (See Fig. 15 for a pictorial example.) Using this fact, for any $h\in\mathsf{HS}$, the set of honest responses

$$\mathbf{z}_h = c\cdot L_{\mathsf{SS},h}\cdot\mathbf{s}_h+\mathbf{r}_h+\mathbf{m}^*_h \quad\wedge\quad \left(\mathbf{z}_j = c\cdot L_{\mathsf{SS},j}\cdot\mathbf{s}_j+\mathbf{r}_j+\mathbf{m}^*_j\right)_{j\in\mathsf{HS}\setminus\{h\}}$$

is distributed identically to

$$\mathbf{z}_h = c\cdot\sum_{j\in\mathsf{HS}}\left(L_{\mathsf{SS},j}\cdot\mathbf{s}_j+\mathbf{r}_j\right)+\overline{\mathbf{m}}^*_h \quad\wedge\quad \left(\mathbf{z}_j = \overline{\mathbf{m}}^*_j\right)_{j\in\mathsf{HS}\setminus\{h\}}$$

for randomly chosen $(\overline{\mathbf{m}}^*_j)_{j\in\mathsf{HS}\setminus\{h\}}$ and $\overline{\mathbf{m}}^*_h := \sum_{j\in\mathsf{HS}}\mathbf{m}^*_j-\sum_{j\in\mathsf{HS}\setminus\{h\}}\overline{\mathbf{m}}^*_j$. Using the correctness of the Shamir secret sharing protocol, this can be further rewritten as

$$\mathbf{z}_h = c\cdot\mathbf{s}-\sum_{j\in\mathsf{CS}}L_{\mathsf{SS},j}\cdot\mathbf{s}_j+\sum_{j\in\mathsf{HS}}\mathbf{r}_j+\overline{\mathbf{m}}^*_h \quad\wedge\quad \left(\mathbf{z}_j = \overline{\mathbf{m}}^*_j\right)_{j\in\mathsf{HS}\setminus\{h\}}$$

Therefore, the set of honest responses can be simulated only with the signing key $\mathbf{s}$, and importantly, all the large Lagrange coefficients multiplied to the partial signing keys $\mathbf{s}_i$ can be moved around to cancel them out. At this point, we can use similar arguments to the standard Lyubashevsky signature to complete the proof. While the concrete method we generate the masks and how the reduction retains consistency within the security proof is different, the way we use the mask to move around the large Lagrange coefficients is identical to [PKM+24].

B Alternative Reduction to sel-AOM-MLWE

We provide an alternative reduction from the MLWE and MSIS problem to sel-AOM-MLWE. While the parameter sets to establish the hardness of this variant of sel-AOM-MLWE is not used in our work, we

nonetheless included it as we believe to be informative to understand the hardness of AOM-MLWE.

In this reduction, we embed a single MLWE instance $\mathbf{t}^* = \mathbf{A}\mathbf{s}^* + \mathbf{e}^*$ in *all* of the columns of $\mathbf{T} = \mathbf{A}\mathbf{S} + \mathbf{E}$ and define the (alternative) accepted linear combinations $\mathcal{L}_{\mathsf{alt}}$ so that this embedding can be done in an unnoticeable manner. At a high level, since the adversary only obtains $(\mathbf{SD}, \mathbf{ED})$ as the MLWE hints, where $\mathbf{D} \in \mathcal{L}_{\mathsf{alt}} \in \mathcal{R}_q^{Q\times(Q-1)}$, we can embed an MLWE instance in the orthogonal subspace spanned by the columns of $\mathbf{D}$.

B.1 Constraints and Parameter Selection

We provide the set of parameters for which we establish hardness of $\mathsf{sel\text{-}AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L}_{\mathsf{alt}},B_{\mathcal{L}_{\mathsf{alt}}},B_{\mathbf{s}},B_{\mathbf{e}}}$. One concrete example of $\mathcal{L}_{\mathsf{alt}}$ satisfying the below constraints is provide in Section 4.5.

Constraints on Parameters. We first give the intermediate variables that will be used during the proof:

- $\mathcal{D}_i$ for $i \in [Q]$ is a discrete Gaussian distribution $\mathcal{D}_{\sigma_i}$ with width $\sigma_i > 0$, where denote $\sigma_{\mathsf{max}} = \max_{i\in[Q]} \sigma_i$ and $\sigma_{\mathsf{min}} = \min_{i\in[Q]} \sigma_i$. In particular, we denote $\sigma^* = \sigma_{\mathsf{min}}$.
- $\tilde{\mathcal{D}}$ is a discrete Gaussian distribution $\mathcal{D}_\sigma$ with width $\sigma > 0$.
- A bound $\gamma_{\mathcal{L}_{\mathsf{alt}}} > 0$ and an efficient algorithm $\mathsf{AppSolveSVP} : \mathcal{R}_q^{Q\times(Q-1)} \to \mathcal{R}_q^Q$ such that for any input $\mathbf{D}$ in the set of accepted linear combinations $\mathcal{L}_{\mathsf{alt}} \subseteq \mathcal{R}_q^{Q\times(Q-1)}$, $\mathsf{AppSolveSVP}$ outputs $\mathbf{w} \in \{\mathbf{w}' \in \mathcal{R}_q^Q \backslash \{\mathbf{0}\} \mid \mathbf{w}'^\top \mathbf{D} = \mathbf{0}\}$ such that $\|w_i\|_2 \le \gamma_{\mathcal{L}_{\mathsf{alt}}}$ for $i \in [Q]$. That is, we assume we can efficiently solve the (approximate) shortest vector problem of a lattice spanned by the orthogonal subspace of any matrix in $\mathcal{L}_{\mathsf{alt}}$.
- α is the order of the Rényi divergence.
- The accepted "slack" for the AOM-MLWE solution is $B_{\mathcal{L}_{\mathsf{alt}}} = 1$. While we can accommodate for a larger slack, we do not investigate this as this section is mainly to show feasibility.
- $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}$ for Lemma B.3, where n is the dimension of $\mathcal{R}_q$, $\mathcal{B}$ and $\mathcal{B}'$ are constructed from the adversary $\mathcal{A}$ against the sel-AOM-MLWE problem.

We now list the constraints for the proof to hold:

- $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ is hard, implying $\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda) = \mathsf{negl}(\lambda)$. I.e., $\sigma \ge \sqrt{\ell} \cdot \omega(\sqrt{\log n})$ using Lemma 3.17.
- $\mathsf{MSIS}_{q,\ell+1,k,B_{\mathsf{MSIS}}}$ is hard, where $B_{\mathsf{MSIS}} = B_{\mathbf{s}} + B_{\mathbf{e}} + \sqrt{n} \cdot (\gamma_{\mathcal{L}_{\mathsf{alt}}} + e^{1/4} \cdot \sigma_{\mathsf{max}} \cdot \sqrt{n} \cdot (\sqrt{\ell} + \sqrt{k}))$, implying $\mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) = \mathsf{negl}(\lambda)$. I.e., $q > B_{\mathsf{MSIS}} \cdot \sqrt{nk} \cdot \omega(\log(nk))$ using Lemma 3.18.
- $2^{-\frac{nk}{10}} = \mathsf{negl}(\lambda)$ to bound the norm of samples from discrete Gaussians using Lemma 3.2.
- $\alpha = \frac{\sigma^*}{\gamma_{\mathcal{L}_{\mathsf{alt}}} \cdot \sigma \cdot n} \cdot \sqrt{\frac{-\log(\epsilon_{\mathsf{lattice}})}{Q\cdot(\ell+k)}} \ge 2$ and $\sigma^* \ge \gamma_{\mathcal{L}_{\mathsf{alt}}} \cdot \sigma \cdot n \cdot \sqrt{Q \cdot (\ell + k)}$.

B.2 Candidate Asymptotic Parameters

Finally, we give a set of asymptotic parameters which fit the above constraints. Below it is helpful to keep in mind that the number Q of MLWE samples and the "quality" $\gamma_{\mathcal{L}_{\mathsf{alt}}}$ of the accepted linear combinations $\mathcal{L}_{\mathsf{alt}}$ dictate the parameters.

Definition B.1 (Alternative Parameters Establishing Hardness of sel-AOM-MLWE). *We denote the set of following asymptotic parameters along with the restricted accepted linear combinations $\mathcal{L}_{\mathsf{alt}}$ and associated algorithm* AppSolveSVP *explained above as* alt-hard-param*.*

- n, ℓ, k such that $n \ge \lambda$.

- $\widetilde{\mathcal{D}} = \mathcal{D}_\sigma$ with $\sigma = \sqrt{\ell} \cdot \log n$.
- $\mathcal{D}_i = \mathcal{D}_{\sigma_i}$ for $i \in [Q]$ such that $\sigma^* = \min_{i\in[Q]} \sigma_i$.
- $\sigma^* = \gamma_{\mathcal{L}_{\mathsf{alt}}} \cdot \sigma \cdot n \cdot \sqrt{Q \cdot (\ell + k)}$.
- q is the smallest prime larger than $B_{\mathsf{MSIS}} \cdot \sqrt{nk} \cdot \log^2(nk)$, where $B_{\mathsf{MSIS}} = B_{\mathbf{s}} + B_{\mathbf{e}} + \sqrt{n} \cdot (\gamma_{\mathcal{L}_{\mathsf{alt}}} + e^{1/4} \cdot \sigma_{\mathsf{max}} \cdot \sqrt{n} \cdot (\sqrt{\ell} + \sqrt{k}))$.
- Plugging in σ^*, $\alpha = \sqrt{-\log(\epsilon_{\mathsf{lattice}})}$ which is larger than 2 assuming hardness of MLWE and MSIS.

B.3 Reduction

The following establishes the hardness of sel-AOM-MLWE for the above parameter selection.

Theorem B.2 (MLWE and MSIS imply sel-AOM-MLWE). *If there exists an adversary $\mathcal{A}$ against the* $\mathsf{sel\text{-}AOM\text{-}MLWE}_{q,\ell,k,Q,(\mathcal{D}_i)_{i\in[Q]},\mathcal{L}_{\mathsf{alt}},B_{\mathcal{L}_{\mathsf{alt}}},B_{\mathbf{s}},B_{\mathbf{e}}}$ *problem, defined with respect to the parameter selection* alt-hard-param *in Definition B.1, then we can construct an adversary $\mathcal{B}$ and $\mathcal{B}'$ against the* $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ *and* $\mathsf{MSIS}_{q,\ell+1,k,B_{\mathsf{MSIS}}}$ *problems such that*

$$\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda) \le \epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-Q \cdot (\ell + k) \cdot \log(\epsilon_{\mathsf{lattice}})} \cdot \frac{\gamma_{\mathcal{L}_{\mathsf{alt}}} \cdot \sigma \cdot n}{\sigma^*}\right) + 2^{-\frac{nk}{10}}.$$

where $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}$ *and* $\mathsf{Time}(\mathcal{B}), \mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathsf{AppSolveSVP})$.

Concretely, plugging in alt-hard-param *and assuming the hardness of* MLWE *and* MSIS*, we have*

$$\mathsf{Adv}^{\mathsf{sel\text{-}AOM\text{-}MLWE}}_{\mathcal{A}}(1^\lambda) = \mathsf{negl}(\lambda).$$

Proof. Let $\mathcal{A}$ be an adversary against the sel-AOM-MLWE problem. Below, we consider a sequence of games where the first game is the original game and the last is a game that can be reduced from the MSIS problem. The detail of each game is provided in Fig. 16. We denote $\mathsf{Adv}^{\mathsf{Game}_i}_{\mathcal{A}}(1^\lambda)$ as the advantage of $\mathcal{A}$ in Game_i.

Game_1**:** This is the real sel-AOM-MLWE game.

Game_2**:** In this game, the challenger samples short vectors $(\widetilde{\mathbf{s}}, \widetilde{\mathbf{e}}) \xleftarrow{\$} \widetilde{\mathcal{D}}^\ell \times \widetilde{\mathcal{D}}^k$ used nowhere in the game and aborts if it exceed some norm bound. Specifically, given $\mathbf{D} \in \mathcal{L}_{\mathsf{alt}}$ from the adversary $\mathcal{A}$, the challenger first computes $\mathbf{w} \leftarrow \mathsf{AppSolveSVP}(\mathbf{D})$. Due to the assumption on the accepted linear combinations $\mathcal{L}_{\mathsf{alt}}$, AppSolveSVP is efficient and $\mathbf{w} \in \mathcal{R}_q^Q$ is a vector satisfying $\mathbf{w}^\top \mathbf{D} = \mathbf{0}$ and $\|w_i\|_2 \le \gamma_{\mathcal{L}_{\mathsf{alt}}}$ for $i \in [Q]$. It then checks the function $\mathsf{BadNorm}(\widetilde{\mathbf{s}}, \widetilde{\mathbf{e}}, \mathbf{w})$ which equals 1 if and only if $\|w_i \cdot \widetilde{\mathbf{s}}\|_2 \ge e^{1/4} \cdot \|w_i\|_1 \cdot \sigma \cdot \sqrt{n\ell}$ or $\|w_i \cdot \widetilde{\mathbf{e}}\|_2 \ge e^{1/4} \cdot \|w_i\|_1 \cdot \sigma \cdot \sqrt{nk}$. Due to Lemma 3.2 and our parameter selection, we have

$$\left|\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda) - \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda)\right| \le 2^{-\frac{nk}{10}}.$$

Game_3**:** In this game, the challenger modifies how the MLWE secret and noise are set. Specifically, it shifts the MLWE secret and noise by $(\widetilde{\mathbf{s}}\mathbf{w}^\top, \widetilde{\mathbf{e}}\mathbf{w}^\top) \in \mathcal{R}_q^{\ell\times Q} \times \mathcal{R}_q^{k\times Q}$.

The only difference between the previous game is the distribution of the MLWE samples. We use the Rényi divergence to relate the advantage of this game to the previous game. While the advantages differ non-negligibly, the difference is polynomially related, which suffices for our purpose.[15] For a fixed pair of $(\widetilde{\mathbf{s}}\mathbf{w}^\top, \widetilde{\mathbf{e}}\mathbf{w}^\top)$, define two distributions $\mathcal{D}_{\vec{\sigma},(\widetilde{\mathbf{s}}\mathbf{w}^\top, \widetilde{\mathbf{e}}\mathbf{w}^\top)} = \{(\mathbf{S} + \widetilde{\mathbf{s}}\mathbf{w}^\top, \mathbf{E} +, \widetilde{\mathbf{e}}\mathbf{w}^\top) \mid (\mathbf{S}, \mathbf{E}) \xleftarrow{\$}$

[15] From a theoretical perspective, we can rely on the statistical distance by simply assuming $\mathcal{D}_i$ is a discrete Gaussian with width that is super-polynomially larger than the size of $\widetilde{\mathbf{s}}\mathbf{w}^\top$ and $\widetilde{\mathbf{e}}\mathbf{w}^\top$. For concrete efficiency, we rely on a more fine-grained analysis by using the upper bound Q and the Rényi divergence.

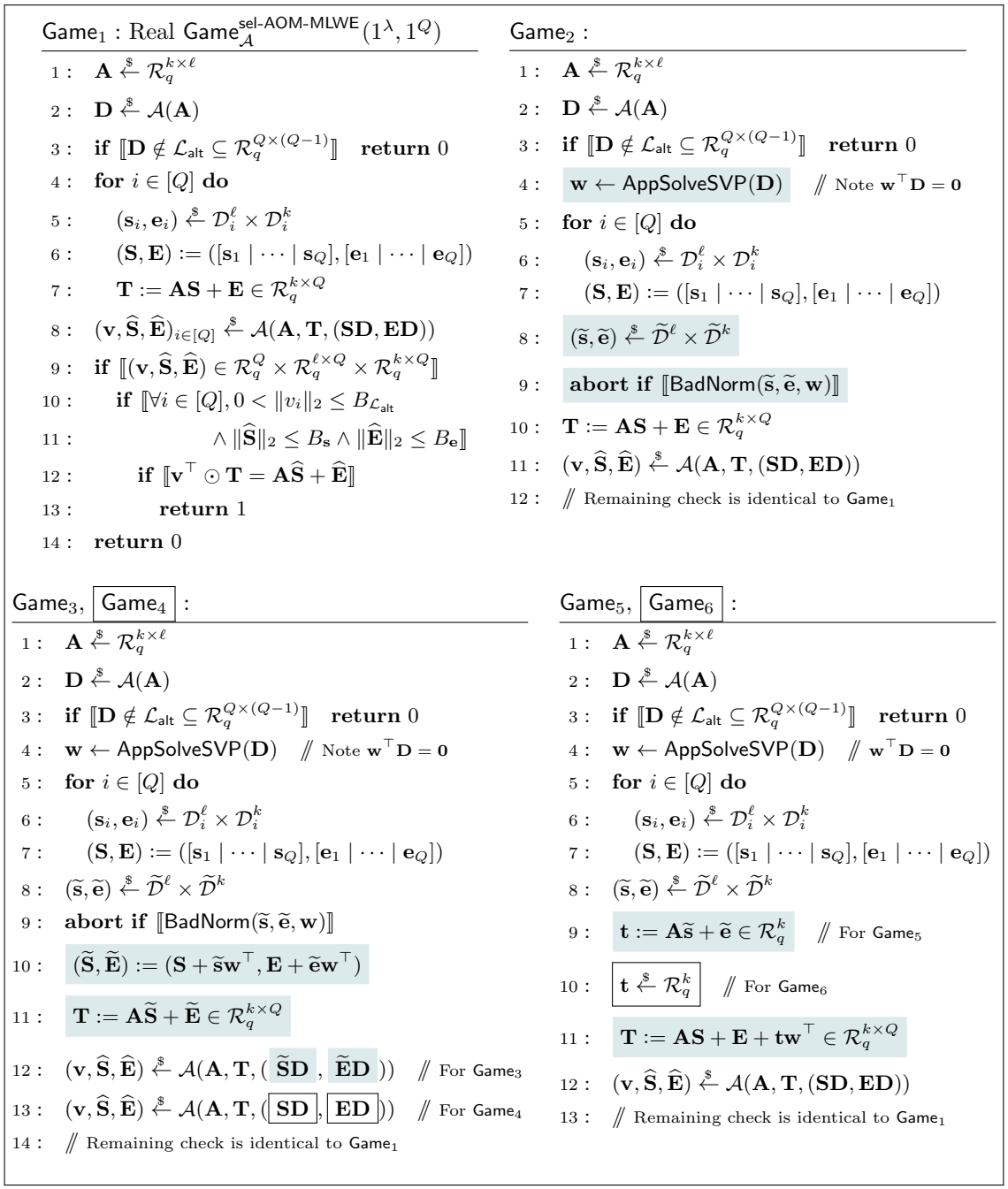

Figure 16: Hybrid games for the proof of Theorem B.2. Recall the game restricts the adversary to output $\mathbf{D}$ that is full-rank and AppSolveSVP solves the shortest vector $\mathbf{w}$ in the orthogonal subspace of $\mathbf{D}$. Game$_4$ is the same as Game$_3$ except the output it provides to the adversary $\mathcal{A}$. Game$_6$ is the same as Game$_5$ except that it samples $\mathbf{t}$ randomly.

$\prod_{i\in[Q]} \mathcal{D}_i^\ell \times \prod_{i\in[Q]} \mathcal{D}_i^k\}$ and $\mathcal{D}_{\vec{\sigma}} = \{(\mathbf{S},\mathbf{E}) \mid (\mathbf{S},\mathbf{E}) \xleftarrow{\$} \prod_{i\in[Q]} \mathcal{D}_i^\ell \times \prod_{i\in[Q]} \mathcal{D}_i^k\}$. We then have the following:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda) &\le \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot R_\alpha(\mathcal{D}_{\vec{\sigma}}; \mathcal{D}_{\vec{\sigma},(\widetilde{\mathbf{s}}\mathbf{w}^\top, \widetilde{\mathbf{e}}\mathbf{w}^\top)}) \\
&\le \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \prod_{i\in[Q]} \exp\left(\frac{\alpha\|w_i\cdot\widetilde{\mathbf{s}}\|_2^2}{2\sigma_i^2}\right) \cdot \prod_{i\in[Q]} \exp\left(\frac{\alpha\|w_i\cdot\widetilde{\mathbf{e}}\|_2^2}{2\sigma_i^2}\right) \\
&\le \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \prod_{i\in[Q]} \exp\left(\frac{\alpha(e^{1/4}\cdot\|w_i\|_1\cdot\sigma\cdot\sqrt{n\ell})^2}{2\sigma_i^2}\right) \\
&\qquad \cdot \prod_{i\in[Q]} \exp\left(\frac{\alpha(e^{1/4}\cdot\|w_i\|_1\cdot\sigma\cdot\sqrt{nk})^2}{2\sigma_i^2}\right) \\
&\le \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda)^{\frac{\alpha-1}{\alpha}} \cdot \exp\left(\frac{Q\cdot\alpha\cdot(\gamma_{\mathcal{L}_{\mathsf{alt}}}\cdot\sigma\cdot n)^2\cdot(\ell+k)}{\sigma^{*2}}\right).
\end{aligned}$$

The first inequality follows from Lemma 3.5, Items 1 and 2, the second follows from Lemma 3.5, Item 3 and Lemma 3.6, the third follows from Lemma 3.2 and the **abort** condition we added in Game_2, and the last follows from the definitions of σ^* and $\gamma_{\mathcal{L}_{\mathsf{alt}}}$ and the facts $\|a\|_1 \le \sqrt{n}\cdot\|a\|_2$ for $a \in \mathbb{R}_q$ and $\sqrt{e}/2 < 1$. We proceed with the hybrid games to prove that $\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda)$ and $\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda)$ are polynomially related for our selection of α.

Game_4**:** In this game, the challenger returns $(\mathbf{SD}, \mathbf{ED})$ to the adversary $\mathcal{A}$ instead of $(\widetilde{\mathbf{SD}}, \widetilde{\mathbf{ED}})$. By definition of $\mathbf{w}$, we have $\mathbf{w}^\top\mathbf{D} = \mathbf{0}$. Hence, $\widetilde{\mathbf{SD}} = (\mathbf{S} + \widetilde{\mathbf{s}}\mathbf{w}^\top)\mathbf{D} = \mathbf{SD}$. The same holds for the MLWE errors. Therefore, the view of $\mathcal{A}$ in both games are identical, and in particular, we have

$$\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(1^\lambda) = \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(1^\lambda).$$

Game_5**:** In this game, the challenger removes the bound check on $(\widetilde{\mathbf{s}}, \widetilde{\mathbf{e}})$ and computes the MLWE instances $\mathbf{T}$ without explicitly computing $(\widetilde{\mathbf{S}}, \widetilde{\mathbf{E}})$. It can be easily checked that the two ways of computing $\mathbf{T}$ are identical as long as **abort** is not triggered. Similarly to Game_2, we have

$$\left|\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(1^\lambda) - \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda)\right| \le 2^{-\frac{nk}{10}}.$$

Game_6**:** Lastly, in this game, the challenger modifies the MLWE instance $\mathbf{t}$ to a random $\mathbf{t} \xleftarrow{\$} \mathcal{R}_q^k$. Notice that in Game_5, the challenger no longer requires the MLWE secret associated to $\mathbf{t}$ to run the game. Thus, it is easy to check that we can construct an MLWE adversary $\mathcal{B}$ that internally runs $\mathcal{A}$ solving the $\mathsf{MLWE}_{q,\ell,k,\mathcal{D}}$ problem such that

$$\left|\mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}}(1^\lambda) - \mathsf{Adv}^{\mathsf{Game}_6}_{\mathcal{A}}(1^\lambda)\right| \le \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda).$$

It is worth noting that this is where we require $\mathsf{AppSolveSVP}$ to be efficient as we have $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathsf{AppSolveSVP})$.

We show in Lemma B.3 that we can construct an MSIS adversary $\mathcal{B}'$ that internally runs $\mathcal{A}$ solving the $\mathsf{MSIS}_{q,\ell+1,k,B_{\mathsf{MSIS}}}$ problem such that

$$\mathsf{Adv}^{\mathsf{Game}_6}_{\mathcal{A}}(1^\lambda) \le \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda).$$

Before providing the proof of Lemma B.3, we finish the proof of Theorem B.2.
Collecting the bounds, we obtain

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda) \le \left(\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}\right)^{\frac{\alpha-1}{\alpha}} \cdot \exp\left(\frac{Q\cdot\alpha\cdot(\gamma_{\mathcal{L}_{\mathsf{alt}}}\cdot\sigma\cdot n)^2\cdot(\ell+k)}{\sigma^{*2}}\right) + 2^{-\frac{nk}{10}}.$$

Plugging our choices of parameters alt-hard-param, we obtain

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda) \le \epsilon_{\mathsf{lattice}} \cdot \exp\left(\sqrt{-Q\cdot(\ell+k)\cdot\log(\epsilon_{\mathsf{lattice}})}\cdot\frac{\gamma_{\mathcal{L}_{\mathsf{alt}}}\cdot\sigma\cdot n}{\sigma^*}\right) + 2^{-\frac{nk}{10}},$$

where $\epsilon_{\mathsf{lattice}} = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}}(1^\lambda) + \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda) + 2^{-\frac{nk}{10}}$ Following an identical argument made in Lemma 4.7, we conclude that the right hand side is negligible, assuming the hardness of the MLWE and MSIS problem. This completes the proof of Theorem B.2. □

It remains to prove the following Lemma B.3.

Lemma B.3. *There exists an adversary $\mathcal{B}'$ that internally runs $\mathcal{A}$ solving the $\mathsf{MSIS}_{q,\ell+1,k,B_{\mathsf{MSIS}}}$ problem such that*

$$\mathsf{Adv}^{\mathsf{Game}_6}_{\mathcal{A}}(1^\lambda) \le \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{B}'}(1^\lambda).$$

Moreover, we have $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathsf{AppSolveSVP})$.

Proof. Let $\mathcal{A}$ be an adversary against the sel-AOM-MLWE problem in Game_6. We construct an adversary $\mathcal{B}'$ solving the MSIS problem having the same advantage as $\mathcal{A}$. Assume $\mathcal{B}'$ is given $\mathbf{M} = [\mathbf{A} \mid \mathbf{t}] \in \mathcal{R}_q^{k\times(\ell+1)}$ as the MSIS problem, where $\mathbf{t} \in \mathcal{R}_q^k$. It then simulates Game_6 to $\mathcal{A}$, where the only difference is that it uses the above computed $(\mathbf{A}, \mathbf{t})$ rather than sampling them. At the end of the game, $\mathcal{A}$ outputs an approximate MLWE solution $(\mathbf{v}, \widehat{\mathbf{S}}, \widehat{\mathbf{E}})$. $\mathcal{B}'$ then computes $\mathbf{S}^* = \begin{bmatrix}\widehat{\mathbf{S}} - \mathbf{v}^\top \odot \mathbf{S} \\ -\mathbf{v}^\top \odot \mathbf{w}^\top \\ \widehat{\mathbf{E}} - \mathbf{v}^\top \odot \mathbf{E}\end{bmatrix} \in \mathcal{R}_q^{(k+\ell+1)\times Q}$ and outputs the first non-zero column in $\mathbf{S}^*$ as the MSIS solution if it exists. If not, it aborts the game.

Let us analyze the success probability of $\mathcal{B}'$. Clearly, we have $\mathsf{Time}(\mathcal{B}') \approx \mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathsf{AppSolveSVP})$. Moreover, if $\mathcal{A}$ breaks sel-AOM-MLWE in Game_6, we have $\mathbf{v}^\top \odot \mathbf{T} = \mathbf{A}\widehat{\mathbf{S}} + \widehat{\mathbf{E}}$. Combining this with $\mathbf{T} = \mathbf{AS} + \mathbf{E} + \mathbf{tw}^\top$, we have

$$\left(\mathbf{A}\widehat{\mathbf{S}} + \widehat{\mathbf{E}}\right) - \mathbf{v}^\top \odot \left(\mathbf{AS} + \mathbf{E} + \mathbf{tw}^\top\right) = \underbrace{[\mathbf{A} \mid \mathbf{t} \mid \mathbf{I}]}_{=[\mathbf{M}|\mathbf{I}]} \underbrace{\begin{bmatrix}\widehat{\mathbf{S}} - \mathbf{v}^\top \odot \mathbf{S} \\ -\mathbf{v}^\top \odot \mathbf{w}^\top \\ \widehat{\mathbf{E}} - \mathbf{v}^\top \odot \mathbf{E}\end{bmatrix}}_{=\mathbf{S}^* \in \mathcal{R}_q^{(k+\ell+1)\times Q}} = \mathbf{0},$$

where we used the fact that for any $\mathbf{u} \in \mathcal{R}_q^c$ and $(\mathbf{B}, \mathbf{C}) \in \mathcal{R}_q^{a\times b} \times \mathcal{R}_q^{b\times c}$, we have $\mathbf{u}^\top \odot (\mathbf{BC}) = \mathbf{B}(\mathbf{u}^\top \odot \mathbf{C})$. We can then bound each column $i \in [Q]$ of $\mathbf{S}^*$ by $\|\mathbf{S}^*_i\|_2 \le \|\widehat{\mathbf{S}}_i\|_2 + \|\widehat{\mathbf{E}}_i\|_2 + \|v_i \cdot w_i\|_2 + \|v_i \cdot \mathbf{S}_i\|_2 + \|v_i \cdot \mathbf{E}_i\|_2 \le B_{\mathbf{s}} + B_{\mathbf{e}} + \sqrt{n}\cdot B_{\mathcal{L}_{\mathsf{alt}}} \cdot (\gamma_{\mathcal{L}_{\mathsf{alt}}} + e^{1/4}\cdot\sigma_{\mathsf{max}}\cdot\sqrt{n}\cdot(\sqrt{\ell}+\sqrt{k})) = B_{\mathsf{MSIS}}$. It remains to check that there is a non-trivial (i.e., non-zero) column in $\mathbf{S}^*$. By the winning condition of $\mathbf{A}$, each entry v_i of $\mathbf{v}$ satisfies $\|v_i\|_2 > 0$. In particular, this implies $v_i \in \mathbb{T}$, where recall $B_{\mathcal{L}_{\mathsf{alt}}} = 1$. Moreover, since $\mathbf{w} \neq \mathbf{0}$, there exists some entry $i \in [Q]$ such that $w_i \neq 0$. Then, we must have $v_i w_i \neq 0$, implying that the i-th column of $\mathbf{S}^*$ is non-zero. Therefore, if $\mathcal{A}$ wins the sel-AOM-UMLWE game, then there always exist a non-zero column in $\mathbf{S}^*$, allowing $\mathcal{B}$ to win the MSIS game as desired. This completes the proof. □

B.3.1 A Concrete Example for $\mathcal{L}_{\mathsf{alt}}$.

Since our result does not use the parameters provided by the alternative reduction, we only briefly discuss a very simple instance of $\mathcal{L}_{\mathsf{alt}}$ for the sake of completeness.

Let us define $\mathcal{L}_{\mathsf{alt}} \in \mathcal{R}_q^{Q\times(Q-1)}$ as a subset of $\{\mathbf{I} \otimes \mathbf{H} \mid \mathbf{H} \in \mathbb{Z}_q^{Q\times(Q-1)}\}$, where $\mathbf{I}$ is the identity matrix of dimension n. Namely, any matrix $\mathbf{D} \in \mathcal{L}_{\mathsf{alt}}$ only contains $\mathbb{Z}_q$ entries. In this particular case, AppSolveSVP can be constructed efficiently. Namely, for any $\mathbf{D} \in \mathcal{L}_{\mathsf{alt}}$, there is an efficiently computable corresponding $\mathbf{H} \in \mathbb{Z}_q^{Q\times(Q-1)}$. For this matrix $\mathbf{H}$, we can first compute some vector $\mathbf{u} \in \mathbb{Z}_q^Q$ such that $\mathbf{u}^\top\mathbf{H} = \mathbf{0}$. Then, assuming the modulus $q = \mathsf{poly}(\lambda)$, we can efficiently brute force search the smallest $a \in \mathbb{Z}_q$ such that $a \cdot \mathbf{u}$ is minimized. Denoting such a vector as $\mathbf{u}^*$. it is clear that $\mathbf{u}^*$ is the shortest vector in the orthogonal space spanned by the columns of $\mathbf{H}$. Then, $\mathsf{AppSolveSVP}(\mathbf{D})$ outputs $\mathbf{w} = \mathbf{u}^* \otimes \mathbf{1}$, where $\mathbf{1}$ is an all one vector of length n. It is clear that $\mathbf{w}^\top\mathbf{D} = \mathbf{0}$ and such $\mathbf{w}$ is short if $\mathbf{u}^*$ is short.