The Algebraic One-More MISIS Problem and Applications to Threshold Signatures

Chenzhi Zhu and Stefano Tessaro

Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
{zhucz20,tessaro}@cs.washington.edu

Abstract. This paper introduces a new one-more computational problem for lattice-based cryptography, which we refer to as the *Algebraic One-More MISIS* problem, or AOM-MISIS for short. It is a modification of the AOM-MLWE problem recently introduced by Espitau et al. (CRYPTO '24) to prove security of new two-round threshold signatures.

Our first main result establishes that the hardness of AOM-MISIS is implied by the hardness of MSIS and MLWE (with suitable parameters), both of which are standard assumptions for efficient lattice-based cryptography. We prove this result via a new generalization of a technique by Tessaro and Zhu (EUROCRYPT '23) used to prove hardness of a one-more problem for linear hash functions assuming their collision resistance, for which no clear lattice analogue was known. Since the hardness of AOM-MISIS implies the hardness of AOM-MLWE, our result resolves the main open question from the work of Espitau et al., who only provided a similar result for AOM-MLWE restricted to *selective* adversaries, a class which does not cover the use for threshold signatures.

Furthermore, we show that our novel formulation of AOM-MISIS offers a better interface to develop tighter security bounds for state-of-the-art two-round threshold signatures. We exemplify this by providing new proofs of security, assuming the hardness of MLWE and MSIS, for two threshold signatures, the one proposed in the same work by Espitau et al., as well as a recent construction by Chairattana-Apirom et al. (ASIACRYPT 2024). For the former scheme, we also show that it satisfies the strongest security notion (TS-UF-4) in the security hierarchy of Bellare et al. (CRYPTO '22), as a result of independent interest.

1 Introduction

One-more assumptions enable proofs of security for several interactive protocols, most notably identification schemes, threshold signatures, multi-signatures, and blind signatures. The perhaps most prominent example is the *one-more discrete logarithm assumption* (OMDL) [BNPS03], which requires the hardness of computing Q discrete logarithm instances given access to an oracle that allows the adversary to compute $Q-1$ discrete logarithms of *arbitrary* group elements. It has been used throughout several security proofs (e.g., cf. [BNPS03, FPS20, NRS21, BCK+22]), where the oracle allows the reduction to simulate secret-key dependent behavior of honest parties without knowing the secret key. This notwithstanding, the main point of controversy is that an assumption such as OMDL is very strong–for instance, Koblitz and Menezes [KM08] point out that in certain groups, it is easier to break OMDL than to solve the standard discrete logarithm problem. The only available route justifying its plausible hardness on standard elliptic curves is a proof [BFP21] in the generic-group model (GGM) [Sho97, Mau05].

Provably-hard one-more problems. A few recent works [TZ23, BLT+24] follow a different path: While they still leverage the power of one-more problems as an intermediate *interface* to design modular security proofs, they also instantiate the underlying mathematical structure to support a proof that the one-more problem is indeed hard based on more standard assumptions, such as the hardness of (standard) discrete log/RSA (in [TZ23]) or of DDH (in [BLT+24]).

This is the angle pursued by this paper—we seek one-more problems which are sufficiently *expressive* to enable useful security proofs, while also enjoying *provable* hardness from standard assumptions. This paper deals specifically with *lattice-based cryptography*. Espitau, Katsumata, and Takemure (EKT) [EKT24] recently introduced a one-more problem they refer to as *algebraic one-more MLWE (AOM-MLWE)* and show that its hardness yields the security of a two-round threshold signature. They also establish the *selective* hardness of AOM-MLWE from the standard MLWE and MSIS assumptions, which is however unfortunately not sufficient to support their application to threshold signatures. Another example—less relevant for this work, however—is the one-more ISIS problem [AKSY22], used in the construction of lattice-based blind signatures, which has also only been validated via cryptanalysis.

Our contributions, in a nutshell. We introduce a new variant of AOM-MLWE which we refer to as *Algebraic One-More MISIS* (AOM-MISIS, for short),[1] and for which we show two different types of results:

- ***Provability.*** We show that the hardness of AOM-MISIS follows from the hardness of MSIS and MLWE, with suitable parameters. In fact, the hardness of AOM-MISIS *implies* the hardness of AOM-MLWE, and thus our result carries over to AOM-MLWE, providing the first reduction of AOM-MLWE to standard assumptions, which was left as a main open question in [EKT24].
- ***Expressivity.*** We show the security of the EKT threshold signature scheme from [EKT24], as well as of the recent scheme by Chairattana-Apirom, Tessaro, and Zhu [CATZ24], assuming the hardness of AOM-MISIS. In turn, this establishes the security of these schemes from MSIS and MLWE. We obtain either better concrete security bounds compared to [CATZ24], or proofs under weaker assumptions compared to [EKT24] for slightly larger concrete parameter sets. We also give a proof that the EKT scheme satisfies the strongest notion of security in the hierarchy of Bellare *et al.* [BCK+22].

Algebraic One-More MISIS. The definition of AOM-MISIS relies on the cyclotomic ring $R = \mathbb{Z}[X]/(X^N+1)$, where N is a power of two, as well as the associated ring $R_q = R/qR \cong \mathbb{Z}_q[X]/(X^N+1)$ for an odd prime q. The problem is defined via the following game:

- **Input.** The adversary is initially given a matrix $A = [\overline{A}|\mathbb{I}_k] \in R_q^{k\times m}$, where $\overline{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$, as well as Q instances $\boldsymbol{t}_i \leftarrow A\boldsymbol{s}_i$, for $i \in [Q]$, where $\boldsymbol{s}_i \leftarrow\!\!\$\ \mathscr{D}_{\sigma_i}^m$ is "small", and sampled from an m-dimensional discrete Gaussian with parameter σ_i. The instance number Q is a parameter of the game.
- **Oracle access.** The adversary can also (adaptively) query an oracle PI which takes as input a Q-dimensional vector $\boldsymbol{b} \in R^Q$, and returns $\sum_{i\in[Q]} b_i\boldsymbol{s}_i$.

To win, the adversary needs to output both $\hat{\boldsymbol{b}} = (\hat{b}_1, \ldots, \hat{b}_Q) \in R^Q$ and a "short" solution $\hat{\boldsymbol{s}} \in R^m$ such that

$$\sum_{i\in[Q]} \hat{b}_i\boldsymbol{t}_i = A\hat{\boldsymbol{s}}\ .$$

To exclude trivial winning strategies, $\hat{\boldsymbol{b}}$ must however not be in the span of the vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots$ queried to PI. We in fact require something stronger, namely that in order to win, the adversary also needs to additionally output $\boldsymbol{u} \in R^Q$, along with $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{s}}$, and the following two properties need to be satisfied:

[1] Our naming is due to the fact that we prefer to think of our problem as a one-more version of Inhomogenous SIS (SIS), rather than of LWE.

- The vector $\boldsymbol{u}$ is orthogonal to all vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots$ queried to PI, but not to $\hat{\boldsymbol{b}}$, i.e., $\boldsymbol{b}_i^\top \boldsymbol{u} = 0$, but $\hat{\boldsymbol{b}}^\top \boldsymbol{u} \neq 0$. Here, orthogonality is defined with respect to the ring R.
- Both vectors $(\hat{b}_1 \cdot \sigma_1, \ldots, \hat{b}_Q \cdot \sigma_Q)$ and $(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)$ are appropriately small, when interpreted as real vectors. Both conditions are necessary, as otherwise a simple attack exists.

Our main result is a concrete reduction showing that the AOM-MISIS problem is indeed hard if the Module LWE (MLWE) and Module SIS (MSIS) assumptions also hold with suitable parameters, related to the parameters of the above game. Our proof relies on a novel *generalization* of the technique by Tessaro and Zhu [TZ23], which was originally used to obtain hard one-more problems based on linear hash functions. The origin of this technique goes back to the analysis of Okamoto signatures [Oka93], and was used also in [HKL19]. Some prior works [TZ23, HKLN20] can be interpreted as attempts to adapt this approach to the lattice setting in limited ways, but in our context, they would lead to worse parameters and restrictions on the generality of our game. We discuss further details in Section 2 below.

We note that the hardness of AOM-MISIS implies the hardness of AOM-MLWE, and thus we resolve the main open problem from [EKT24], which only provided an analysis for *selective* adversaries issuing their oracle queries beforehand. In fact, the main difference between AOM-MISIS and AOM-MLWE is the winning condition. We require outputting a single short solution for a suitable non-trivial linear combination of the challenges, whereas AOM-MLWE requires outputting Q short solutions for all challenges, given $Q-1$ access to a similar oracle, although with different conditions on the queries. These changes are what in part enables simpler reductions for threshold signatures. A more detailed discussion is provided in Section 4.2.

Applications to threshold signatures. We first recall that in a *t-out-of-n threshold signature scheme* [Des88, DF90], n potential signers each hold a secret share of a secret signing key, with an associated public verification key. Any subset of (at least) t of these signers is able to jointly produce a signature, via interaction, whereas an adversary that controls fewer than t signers should not be able, on its own, to come up with a valid signature.

We leverage AOM-MISIS to obtain tighter analyses for state-of-the-art *two-round* lattice-based threshold signatures, based on the MSIS/MLWE assumptions, and without assuming the hardness of any ad-hoc one-more problem. We focus on two threshold signatures, which we refer to as CTZ [CATZ24] and EKT [EKT24]. There are several small differences between the two constructions, despite the fact that they instantiate similar ideas. Both can be seen as a natural threshold version of the Fiat-Shamir-with-abort paradigm [Lyu09] that also underlies DILITHIUM [LDK+22], albeit dispensing with the actual abort, and using a sufficiently large modulus instead. In particular, EKT is a threshold version of Raccoon [dPKPR24], submitted to the additional NIST call for post-quantum signatures [Natnt]. They are natural lattice analogues of FROST [KG20, BCK+22, CKGW22], a very lightweight threshold Schnorr signature.

Taking some liberty from their formal description, in both schemes, as a result of the second round, the i-th signer produces an affine signature share

$$\boldsymbol{z}_i = \boldsymbol{r}_i + c \cdot \lambda_i \mathsf{ss}_i \ ,$$

in a suitable algebraic structure, where ss_i is the i-th signer's key share, c is a hash value associated with the signature, and λ_i is a linear reconstruction coefficient associated with the secret sharing scheme. With S being the set of signers, the final signature has format $(\boldsymbol{R}, \boldsymbol{z})$, where $\boldsymbol{R} = A \sum_{i \in S} \boldsymbol{r}_i$, ($A$ is a public matrix) and $\boldsymbol{z} = \sum_{i \in S} \boldsymbol{z}_i$. However, the two schemes differ in the following aspects:

- In CTZ, to avoid leakage of the secret shares, $\lambda_i \mathsf{ss}_i$ needs to remain sufficiently small and thus a secret sharing scheme with small reconstruction coefficients is needed. This incurs a significant cost due to the larger share size when compared to Shamir secret sharing. The security proof in [CATZ24] is fairly involved, and gives a direct reduction to MSIS.
- To enable the use of Shamir secret sharing, EKT changes the initial setup to ensure that any pair of signers shares a secret key, and these keys are used, in each execution of the signing protocol, to generates (pseudo)random masks $(\mathsf{msk}_i)_{i \in S}$ such that $\sum_{i \in S} \mathsf{msk}_i = 0$. The i-th signer then sends $\boldsymbol{z}_i + \mathsf{msk}_i$ in the second round instead of $\boldsymbol{z}_i$, and this ensures that only the *sum* of the $\boldsymbol{z}_i$'s is leaked. The security proof in [EKT24] relies on the direct use of the AOM-MLWE assumption, and since the constructed adversary is inherently not selective, they need to rely on a conjecture about the hardness of AOM-MLWE.

In Sections 5 and 6, we give new security analyses for both schemes based on direct reductions from AOM-MISIS, which in turn yield concrete security proofs from MLWE and MSIS. Our result for CTZ yields better parameters than the analysis of [CATZ24]. The result about EKT, in addition to now basing security on MLWE/MSIS alone, also shows a strong security property for this scheme, namely TS-UF-4 in the hierarchy of Bellare et al. [BCK+22]. The concrete efficiency numbers derived from our bounds are somewhat worse than those from EKT, but their concrete analysis relies on their own cryptanalysis of AOM-MLWE, whereas we use standard parameters for MLWE and MSIS. Closing the gap between the cryptoanalysis-driven parameter choices and those derived from our security reduction remains an interesting open problem.

The recent work on Ringtail [BKL+24] also proposes a threshold signature scheme similar in spirit to EKT and with a proof of security under MLWE. In Ringtail, unlike EKT, signers need to know already in the first round the set of involved signers S, and thus, unlike EKT and CTZ, Ringtail is not partially non-interactive in the sense of Bellare *et al.* [BCK+22]. Ringtail also needs to authenticate first-round messages. The authors of Ringtail mention removing these restrictions, while preserving comparable efficiency and provability from MLWE/MSIS, is an open problem, which we resolve here based on the EKT scheme.

Ringtail's security bound degrades with the number of random oracle queries, whereas ours degrades with the number of signing sessions. We note that the latter is a system parameter that can be enforced, whereas the former scales with the running time of the adversary.

Remarks on CTZ. While the concrete efficiency of CTZ is significantly worse than that of EKT, we see value in the CTZ construction because it does not rely on pairwise masks. CTZ closely resembles the structure of the original FROST protocol, making it a promising starting point for achieving identifiable aborts (constructing efficient partially non-interactive lattice-based threshold signatures with identifiable aborts is a big open problem in the field). In contrast, achieving identifiable aborts in EKT seems less likely without relying on heavyweight NIZK proofs, due to its reliance on pairwise masks.

Also, the security argument of CTZ works for a class of linear secret sharing schemes rather than a specific scheme. Its main source of inefficiency stems from the underlying secret sharing scheme, and its efficiency could be significantly improved if better instantiations of the underlying secret sharing scheme are proposed.

Other related work. We note here that there are other approaches to threshold signatures. First off, *Fully-Homomorphic Encryption* (FHE) generically yields round-optimal threshold signatures [BGGK17, BGG+18, ASY22]. These require however the homomorphic evaluation of the signing algorithm, and thus come with a substantial computational overhead. Earlier works [DOTT21,

Che23] proposed two-round n-out-of-n threshold signatures derived from constructions for the related notion of multi-signatures. Gur *et al.* [GKS23] proposed two-round construction based on linearly homomorphic encryption (LHE) which supports arbitrary thresholds. Both rounds are message-dependent. More recently, Pino *et al.* [DKM+24] propose a more efficient lattice-based threshold signature scheme that does not rely on FHE or the aforementioned heavy primitives, but the drawback is that the protocol has three message-dependent rounds. Recent work [KRT24] also gives a five-round threshold signature with adaptive security based on MLWE/MSIS, whereas a threshold version of Falcon [PFH+22] was presented in [ENP24]. The latter scheme is designed for robustness but requires four message-dependent rounds and incurs quadratic communication complexity in the threshold.

2 Technical Overview

The main goal of this section is to provide a detailed overview of our main result establishing the hardness of AOM-MISIS based on MSIS and MLWE. We stress that our reduction will involve concrete parameters, but we keep the discussion in this section somewhat qualitative on this front.

At a technical level, it helps to see AOM-MISIS as a lattice-based analogue of the AOMPR framework proposed by Tessaro and Zhu (TZ) [TZ23] to define one-more problems for linear hash functions. Their framework does not cover lattice problems, however—we aim to adapt it to lattices, and this presents a number of challenges, which we explain in this section. We will also discuss how prior works have already tried to overcome such challenges, how those approaches are not sufficient for our purposes, and what we do instead.

We focus first on a variant of AOM-MISIS we denote as wAOM-MISIS, where the adversary does not output the relaxation vector $\hat{\boldsymbol{b}}$ and is instead asked to output Q solutions $\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_Q$ with small norm (instead of a single solution) such that $\boldsymbol{t}_i = A\hat{\boldsymbol{s}}_i$ for all $i \in [Q]$. Also, the constraint $\hat{\boldsymbol{b}}^T \boldsymbol{u} \neq 0$ is now changed to $\boldsymbol{u} \neq 0$ instead, while $\boldsymbol{b}_i^T \boldsymbol{u} = 0$ still must hold for the queries to PI.

This problem is not easier than AOM-MISIS: If the adversary wins the wAOM-MISIS game, there exists an index $i \in [Q]$ such that $u_i \neq 0$, and thus the adversary can win the original game by outputting $(\hat{\boldsymbol{s}}_i, \boldsymbol{e}_i, \boldsymbol{u})$, where $\boldsymbol{e}_i$ is the i-th unit vector. This is also the idea underlying the proof that AOM-MISIS hardness implies the hardness of AOM-MLWE in Section 4.2.

Reduction ideas from [TZ23]. We now provide a self-contained review of TZ's proof framework in the context of wAOM-MISIS, and explain where it fails. In particular, given an adversary $\mathcal{A}$ for the wAOM-MISIS game, we build a simple MSIS adversary $\mathcal{B}$ that takes a random matrix $A \in R_q^{k \times m}$ as input and outputs a vector $\boldsymbol{z}$ with small norm such that $A\boldsymbol{z} = 0$. To start with, $\mathcal{B}$ runs $\mathcal{A}$ by simulating the wAOM-MISIS game faithfully using the matrix A and sampling $\boldsymbol{s}_i$ by itself. After receiving $(\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_Q, \boldsymbol{u})$, if $\mathcal{A}$ wins, $\mathcal{B}$ finds an index $i \in [Q]$ such that $\hat{\boldsymbol{s}}_i \neq \boldsymbol{s}_i$ and outputs $\boldsymbol{z} \leftarrow \hat{\boldsymbol{s}}_i - \boldsymbol{s}_i$. Otherwise, $\mathcal{B}$ aborts.

Analysis of $\mathcal{B}$. It is not hard to see that $\mathcal{B}$ wins the MSIS game if such an index i indeed exists, since $A\boldsymbol{z} = A\hat{\boldsymbol{s}}_i - A\boldsymbol{s}_i = 0$ and $\|\boldsymbol{z}\| \leqslant \|\boldsymbol{s}_i\| + \|\hat{\boldsymbol{s}}_i\|$ is bounded given both $\|\boldsymbol{s}_i\|$ and $\|\hat{\boldsymbol{s}}_i\|$ are bounded. Here $\|\cdot\|$ denotes the ℓ_2-norm. The hard part is to show that such i exists w.h.p. under the assumption that $\mathcal{A}$ wins w.h.p., and this is what we discuss next.

Following the same lines as TZ, we can assume w.l.o.g. that $\mathcal{A}$ is deterministic, and thus the view $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ of $\mathcal{A}$ (assuming the matrix A is fixed) is completely determined by the

challenge secrets $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Also, we denote by $\mathbf{D}$ the set of secrets $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ such that $\mathcal{A}$ wins the wAOM-MISIS game.

While this is not the case for us, suppose for now that $\mathbf{D}$ is finite and each $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ is sampled uniformly from $\mathbf{D}$. The first step is to find a *derangement* Φ of $\mathbf{D}$, i.e., a permutation of $\mathbf{D}$ without fixed points, such that for any $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$, $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) = \mathrm{View}_{\mathcal{A}}(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q)$ with $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) = \Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Therefore, $\mathcal{A}$ also produces the same output $(\hat{\boldsymbol{s}_1}, \ldots, \hat{\boldsymbol{s}_n}, \boldsymbol{u})$ in both cases. Since $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \neq (\boldsymbol{s}'_1, \ldots, \boldsymbol{s}_Q)$, there exists an index $i \in [Q]$ such that $\boldsymbol{s}_i \neq \boldsymbol{s}'_i$ and thus either $\boldsymbol{s}_i \neq \hat{\boldsymbol{s}}_i$ or $\boldsymbol{s}'_i \neq \hat{\boldsymbol{s}}_i$, which means $\mathcal{B}$ wins the MSIS game in at least one of the cases. Since Φ is a permutation of $\mathbf{D}$, for at least half of elements in $\mathbf{D}$ lead to $\mathcal{B}$ winning. Therefore, $\mathcal{B}$ wins with at least half of the probability that $\mathcal{A}$ wins. Crucially, note that Φ does not need to be efficient for this argument to succeed, as the algorithm $\mathcal{B}$ described above is the actual reduction.

Constructing Φ. The first idea is to define Φ such that

$$\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) := (\boldsymbol{s}_1 + u_1 \cdot \Delta, \ldots, \boldsymbol{s}_Q + u_Q \cdot \Delta)\,, \tag{1}$$

where $\Delta \in R^m$ is a non-zero vector such that $A\Delta = 0$ (we will discuss below how to ensure such Δ exists), and $\boldsymbol{u}$ is the vector output by $\mathcal{A}$ in an execution with the secret value $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. It is not hard to check that $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ and $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ produce the same view. First off, since $A(\boldsymbol{s}_i + u_i \cdot \Delta) = A\boldsymbol{s}_i + u_i \cdot (A\Delta) = A\boldsymbol{s}_i$, the input of $\mathcal{A}$ is identical in both cases. Further, for each PI query $\boldsymbol{b}$ made by $\mathcal{A}$, since $\boldsymbol{u}$ is a non-zero vector satisfying $\sum_{i\in[Q]} u_i b_i = 0$, we have $\sum_{i\in[Q]} b_i(\boldsymbol{s}_i + u_i \cdot \Delta) = \sum_{i\in[Q]} b_i \boldsymbol{s}_i + (\sum_{i\in[Q]} b_i u_i) \cdot \Delta = \sum_{i\in[Q]} b_i \boldsymbol{s}_i$, which means the response received by $\mathcal{A}$ is identical in both cases.

Issues and prior approaches. The issue with the above approach is that in our setting the secrets are not sampled uniformly from a set. Two prior works [TZ23, HKLN20] have overcome this issue by sampling $\boldsymbol{s}_i$ uniformly from a bounded box $\mathscr{B}^m_{\beta_i} := \{\boldsymbol{x} \in R^m | \, \|\boldsymbol{x}\| \leqslant \beta_i\}$. Still, the challenge now is that Φ is no longer a permutation over a subset of $\mathscr{B}^m_{\beta_1} \times \cdots \times \mathscr{B}^m_{\beta_Q}$, since $\|\boldsymbol{s}_i + u_i \cdot \Delta\|$ can be larger than β_i even though $\|\boldsymbol{s}_i\| \leqslant \beta_i$.

To overcome this issue, Hauck *et al.* [HKLN20][2] proposed to set β_i to be very large such that the total faction of $\boldsymbol{s}_i \in \mathscr{B}^m_{\sigma_i}$ such that $\boldsymbol{s}_i + u_i \cdot \Delta$ falls outside $\mathscr{B}^m_{\sigma_i}$ is negligible in the security parameter κ. Then, one can still show $\mathcal{B}$ wins with nearly at least half of the winning probability of $\mathcal{A}$. However, this requires $\sigma_i = \Omega(2^\kappa \|u_i \Delta\|)$, which results in very inefficient constructions using this approach.

Chairattana-Apirom et al. [CATZ24] overcame the 2^κ barrier with a different approach. They let $\boldsymbol{s}_1$ be sampled from $\mathscr{B}^m_{\beta_1}$, whereas $\boldsymbol{s}_i$ is sampled from $\mathscr{D}^m_{\sigma_i}$. They use this in the security proof of their threshold signature scheme based on the MSIS assumption, but their technique can be massaged into a reduction from MSIS to a restricted version of the wAOM-MISIS game, where the PI queries are restricted, but still sufficient for their application.

Their proof is in some sense a probabilistic relaxation of TZ's, in that for any $\boldsymbol{s}_1 \in R^m$, they consider a *random variable* $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1)$ which represents the view of $\mathcal{A}$ given the first secret is $\boldsymbol{s}_1$, whereas the remaining secrets are sampled afresh from discrete Gaussian distributions. If for any $\boldsymbol{s}_1 \neq \boldsymbol{s}'_1 \in R^m$, such that $A\boldsymbol{s}_1 = A\boldsymbol{s}'_1$, we can show that $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1)$ and $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}'_1)$ are identically distributed, then we can once again carry out the same argument as above. They in fact generalize this argument by showing that if the Rényi divergence of $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1)$ from $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}'_1)$ is small, one

[2] We note that [HKLN20] precedes [TZ23], but deals with a lattice-analogue of a linear-hash function based framework introduced by [HKL19] similar in spirit to that of [TZ23].

can still show that the winning probabilities of $\mathcal{B}$ and $\mathcal{A}$ are sufficiently related. Finally, they show the Rényi divergence is indeed bounded for the special types of queries they consider. Doing so, they ensure that σ_i depends linearly on $\sqrt{Q}$ instead of 2^κ.

Still, their reduction fails if $\mathcal{A}$ issues general PI queries, and this is because the Rényi divergence of $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1)$ from $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}'_1)$ is no longer bounded. For example, the first secret (either $\boldsymbol{s}_1$ or $\boldsymbol{s}'_1$) can be recovered from an PI query with input $(1, 0, \ldots, 0)$. Moreover, they need to ensure that almost all $\boldsymbol{s}_1$'s have one partner $\boldsymbol{s}'_i \neq \boldsymbol{s}_i \in \mathscr{B}^m_{\beta_1}$ such that $A\boldsymbol{s}_1 = A\boldsymbol{s}'_1$, and this requires $\beta_1 \geqslant 2^{\kappa/N} q^{k/m}$,[3] which leads to inefficient parameters in their threshold signature.

2.1 Step 1: Generalizing TZ's argument

Our goal is to provide a reduction that supports an adversary $\mathcal{A}$ without any extra restrictions on its queries to PI, and furthermore that ensures hardness even for sufficiently small parameter choices in wAOM-MISIS. For this reason, we take a different route and generalize TZ's argument to a setting where each $\boldsymbol{s}_i$ is sampled (independently) from an arbitrary distribution $\mathcal{P}_i$, rather than uniformly from a finite set.

Let $\mathcal{P} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_Q$ be the joint distribution of $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Then, instead of requiring Φ to be a permutation of $\mathbf{D}$ (recalling that $\mathbf{D}$ is the set of secrets $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ that make $\mathcal{A}$ win the wAOM-MISIS game), we require that the distribution of $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ conditioned on $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$ (denoted as $\mathcal{P}_{|\mathbf{D}}$) is identical to that of $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ conditioned on $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$ (denoted as $\Phi(\mathcal{P}_{|\mathbf{D}})$). The original TZ approach corresponds to the case when $\mathcal{P}$ is a uniform distribution over $\mathbf{D}$. Here, since arbitrary distributions over $\mathbf{D}$ are allowed, we no longer require $\mathbf{D}$ to be a finite set. The two other requirements for Φ remain the same, i.e., Φ has no fixed point over $\mathbf{D}$ and $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) = \mathrm{View}_{\mathcal{A}}(\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q))$.

Our key observation here is that the previous argument to lower bound the success probability of $\mathcal{B}$ still applies. To see this, it helps us to extend Φ to the space of all potential secrets (not only those in $\mathbf{D}$) by defining $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) := (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ for all $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in R^m \backslash \mathbf{D}$. Then, consider the following two adversaries $\mathcal{B}'$ and $\mathcal{B}''$.

- $\mathcal{B}'$ is the same as $\mathcal{B}$ except that $\mathcal{B}'$ samples $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ from $\Phi(\mathcal{P})$.
- $\mathcal{B}''$ is the same as $\mathcal{B}$ except that 1. $\mathcal{B}''$ samples $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \leftarrow\!\!\$\, \mathcal{P}$ and computes secrets as $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \leftarrow \Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$; 2. after $\mathcal{A}$ returns, $\mathcal{B}''$ outputs $\hat{\boldsymbol{s}}_i - \boldsymbol{r}_i$ if there exists $\boldsymbol{r}_i \neq \hat{\boldsymbol{s}}_i$.

We note that $\mathcal{B}'$ and $\mathcal{B}''$ do not need to be efficient. They are only used to compute the winning probability of the (efficient) adversary $\mathcal{B}$.

Denote now by $\mathsf{PWin}_{\mathcal{X}}$ the winning probability of $\mathcal{X} \in \{\mathcal{A}, \mathcal{B}, \mathcal{B}', \mathcal{B}''\}$. Then, the conclusion that $\mathsf{PWin}_{\mathcal{B}} \geqslant 1/2\mathsf{PWin}_{\mathcal{A}}$ is a straightforward corollary of the following three facts.

Fact 1. $\mathsf{PWin}_{\mathcal{B}} = \mathsf{PWin}_{\mathcal{B}'}$;
Fact 2. $\mathsf{PWin}_{\mathcal{B}} = \mathsf{PWin}_{\mathcal{B}''}$;
Fact 3. $\mathsf{PWin}_{\mathcal{B}'} + \mathsf{PWin}_{\mathcal{B}''} \geqslant \mathsf{PWin}_{\mathcal{A}}$.

Fact 1 is straightforward since $\mathcal{P}$ is identical to $\Phi(\mathcal{P})$, which means $\mathcal{B}$ and $\mathcal{B}'$ behave the same. Fact 2 is also not hard to see. Since $\mathrm{View}_A(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) = \mathrm{View}_A(\Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)) = \mathrm{View}_A(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$, even if $\mathcal{B}''$ sets the secrets to $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$ instead of $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q$, the output of $\mathcal{B}''$ remain the same. However, then, $\mathcal{B}''$ is identical to $\mathcal{B}$, which implies the second fact.

[3] Since N is usually set to 512 or 1024, the leading term here is $q^{k/m}$.

Finally, to prove the third fact, we can interpret the sampling process of $\mathcal{B}'$ as first sampling $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$ from $\mathcal{P}$ and then set $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \leftarrow \Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$. Then, the only difference between $\mathcal{B}'$ and $\mathcal{B}''$ is that $\mathcal{B}'$ check whether there exists $\boldsymbol{s}_i \neq \hat{\boldsymbol{s}}_i$, while $\mathcal{B}''$ check whether there exists $\boldsymbol{r}_i \neq \hat{\boldsymbol{s}}_i$. Since Φ has no fixed point over $\mathbf{D}$, we know $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \neq (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ if $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \in \mathbf{D}$. Therefore, for each $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \in \mathbf{D}$, at least one of $\mathcal{B}'$ and $\mathcal{B}''$ wins, which implies Fact 3.

However, if we use Φ from Equation 1, this extended argument does not give us any benefit. Indeed, Φ as define is a bijection over $\mathbf{D}$, and therefore, the relaxed condition $\mathcal{P}_{|\mathbf{D}} = \Phi(\mathcal{P}_{|\mathbf{D}})$ still requires $\mathcal{P}$ to be a uniform distribution over $\mathbf{D}$. Also, it is unclear whether there are other ways to construct such Φ.

Further relaxation on the requirements of Φ. The way out from the above situation is that that the two distributions do not need to be exactly identical. In fact, this condition is only needed by Fact 1. Concretely, we relax the condition that $\mathcal{P}_{|\mathbf{D}} = \Phi(\mathcal{P}_{|\mathbf{D}})$ to requiring that the Rényi divergence of $\mathcal{P}_{|\mathbf{D}}$ from $\Phi(\mathcal{P}_{|\mathbf{D}})$ is small, denoted as $R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right)$, where $\alpha > 1$ is a parameter we can choose. Then, due to the property of Rényi divergence (formally see Lemma 4),

$$\mathsf{PWin}_{\mathcal{B}'} \leqslant \left(R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right) \cdot \mathsf{PWin}_{\mathcal{B}}\right)^{(\alpha-1)/\alpha} .$$

Combining with Fact 2 and 3, it gives

$$\mathsf{PWin}_{\mathcal{A}} \leqslant \mathsf{PWin}_{\mathcal{B}} + \left(R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right) \cdot \mathsf{PWin}_{\mathcal{B}}\right)^{(\alpha-1)/\alpha} ,$$

which upper bounds the winning probability of $\mathcal{A}$ by the advantage of solving the MSIS problem. We note that it is possible to get analogous statements using other distance measures, but they fail to give equally good parameters in our application scenarios.

Also, we emphasize that although both our work and [CATZ24] use Rényi divergence, the latter work bounds Rényi divergence between the *views* of $\mathcal{A}$, when run with different secrets, whereas we bound the Rényi divergence between distributions of *secrets* before and after applying the Φ map. The context where we use Rényi divergence is also very different from that of other works in lattice-based cryptography.

2.2 Step 2: Upper bounding $R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right)$

It is left to show that $R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right)$ is bounded, for the specific construction of Φ given in Equation 1, and again assuming a suitable Δ exists. The intuition that the two distributions should be close is that Φ is local in the sense that it maps each element to a nearby one given $\|(u_1\Delta, \ldots, u_Q\Delta)\|$ is small. Since each secret is sampled from a discrete Gaussian distribution, elements that are close in distance have similar probabilities of being sampled. However, turning this idea into a proof is still challenging, as both $\mathcal{P}_{|\mathbf{D}}$ and $\Phi(\mathcal{P}_{|\mathbf{D}})$ are not "well-structured" distributions, since both the set $\mathbf{D}$ and $\boldsymbol{u}$ can be arbitrarily determined by $\mathcal{A}$.

Our key insight here is that $\mathbf{D}$ can be decomposed into several disjoint sets such that the distributions conditioned on each set is a one-dimensional discrete Gaussian. Then, we can upper bound the Rényi divergence of the distributions conditioned on each set, which implies an upper bound on $R_\alpha\left(\Phi(\mathcal{P}_{|\mathbf{D}}) \| \mathcal{P}_{|\mathbf{D}}\right)$.

More precisely, for each $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$, our definition of Φ in Equation 1 also ensures that $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ is identical to $\mathrm{View}_{\mathcal{A}}(\boldsymbol{s}_1 + ku_1 \cdot \Delta, \ldots, \boldsymbol{s}_Q + ku_Q \cdot \Delta)$ for any $k \in \mathbb{Z}$, where $\boldsymbol{u}$ is the vector output by $\mathcal{A}$ given secrets being $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Denote $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q] := \{(\boldsymbol{s}_1 + ku_1 \cdot$

$\Delta, \ldots, \boldsymbol{s}_Q + ku_Q \cdot \Delta) | k \in \mathbb{Z}\}$. We can show that Φ is a bijection over $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$, and all the $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$ sets are either the same or disjoint with each other. Therefore, we just need to bound

$$R_\alpha \left(\Phi(\mathcal{P}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]}) \| \mathcal{P}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]} \right)$$

for each $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$.

Now the problem is much easier since $\mathcal{P}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]}$ is well structured: each $\mathcal{P}$ is a discrete Gaussian distribution over R^{Qm}, and $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$ is a one-dimensional lattice over R^{Qm}. In fact, we can transform $\mathcal{P}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]}$ into a discrete Gaussian over $\mathbb{Z}$ with standard deviation $\|(u_1/\sigma_1 \cdot \Delta, \ldots, u_Q/\sigma_Q \cdot \Delta))\|^{-1}$,[4] while $\Phi(\mathcal{P}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]})$ is exactly the same distribution shifted by 1. Therefore, we can use the upper bound from [TT15] (see also Lemma 6) of the Rényi divergence.

We refer to Section 4.3 for the proof details.

Further optimizations. We explain in this subsection two further optimizations we do to the reduction, which give us better parameters.

Outputting only one solution. We observe that the reduction $\mathcal{B}$ uses only one of the solutions output by $\mathcal{A}$, which means most of the solutions are actually redundant. In fact, the reduction still works if $\mathcal{A}$ outputs a single solution $(i, \hat{\boldsymbol{s}})$ for the i-th challenge satisfying $u_i \neq 0$ and $\mathcal{B}$ just outputs $\hat{\boldsymbol{s}} - \boldsymbol{s}_i$ if $\hat{\boldsymbol{s}} \neq \boldsymbol{s}_i$. This is because for each $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in \mathbf{D}$ and $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) = \Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$, we have $\boldsymbol{s}_i \neq \boldsymbol{s}_i + u_i \cdot \Delta = \boldsymbol{s}'_i$ if $u_i \neq 0$, and thus $\mathcal{B}$ wins in at least one of the cases if $\mathcal{A}$ outputs a solution for the i-th challenge. This leads us to define our AOM-MISIS problem, where $\mathcal{A}$ is only required to output a single solution $\hat{\boldsymbol{s}}$ together with a relaxation vector $\hat{\boldsymbol{b}}$ such that $\sum_{i\in[Q]} \hat{b}_i \boldsymbol{t}_i = A\hat{\boldsymbol{s}}$. Similarly, the requirement on the special solution becomes $\sum_{i\in[Q]} u_i \hat{b}_i \neq 0$. The relaxation vector $\hat{\boldsymbol{b}}$ is needed when reducing the security of the threshold signatures to the hardness of AOM-MISIS.

Reducing the norm of Δ. One question we do not answer in the above discussion is how to guarantee the existence of a small non-zero vector $\Delta \in R^m$ such that $A\Delta = 0$. Such Δ must exist in $\mathscr{B}^m_\beta$ if $|\mathscr{B}^m_{\beta/2}| > q^{kN}$, which implies $\beta = \Omega(q^{k/m})$. However, this results in a large $\|\Delta\|$ negatively impacting both the Rényi divergence bound and σ_i, which depends linearly on $\|\Delta\|$.

We can get a smaller $\|\Delta\|$ by relying on the MLWE assumption to embed a small Δ into A as a (very minimal) trapdoor. We sample A as $[\boldsymbol{d}|D|\mathbb{I}_k]$, where $D \leftarrow\!\!\$\ R_q^{(m-k-1)\times k}$ and $\boldsymbol{d} = D\boldsymbol{a} + \boldsymbol{e}$ with $\boldsymbol{a} \leftarrow\!\!\$\ \mathscr{B}^{m-k-1}_{\beta_{\mathsf{lwe}}}$ and $\boldsymbol{e} \leftarrow\!\!\$\ \mathscr{B}^{k}_{\beta_{\mathsf{lwe}}}$. Then, we can let $\Delta = (1, -\boldsymbol{a}, -\boldsymbol{e})$, which satisfies $A\Delta = 0$, and $\|\Delta\| \leqslant \sqrt{m}\beta_{\mathsf{lwe}}$, which is much smaller than $q^{k/m}$. By the MLWE assumption, $[\boldsymbol{d}|D]$ is computationally indistinguishable from a matrix uniformly sampled from $R_q^{(m-k)\times k}$.

Two remarks are in order. The first is that this trick would not have worked to improve the parameters of [CATZ24] directly, as they need something stronger than the existence of a small Δ. Second, the way we use MLWE here is different than its use to improve parameters in prior works on lattice-based signatures and threshold signatures (e.g. [Lyu12, BTT22, DKM+24]). It is really tailored at supporting our tighter analysis of the reduction via the embedding of a small enough Δ.

[4] This is the reason why we bound $\|u_1/\sigma_1, \ldots, u_Q/\sigma_Q\|$ in the winning condition.

3 Preliminaries

3.1 Notation

For any positive integers $k < n$, $[n]$ denotes $\{1, \ldots, n\}$, and $[k..n]$ denotes $\{k, \ldots, n\}$. We use κ to denote the security parameter. For a sequence of variables $x_1, \ldots, x_\ell$, we use $x_{[i]}$ to denote $(x_1, \ldots, x_j)$ and $x_{[i..j]}$ to denote $(x_i, \ldots, x_j)$.

For a finite set S, $|S|$ denotes the size of S, and $x \leftarrow\!\!\$\ S$ denotes sampling an element uniformly from S and assigning it to x. For a distribution $\mathcal{D}$, denote $\mathrm{Supp}(\mathcal{D})$ as the support of $\mathcal{D}$, and $x \leftarrow\!\!\$\ \mathcal{D}$ denotes sampling x according to $\mathcal{D}$. For any $y \in \mathrm{Supp}(\mathcal{D})$, denote $\mathcal{D}(y) := \mathsf{Pr}_{x \leftarrow\!\!\$\ \mathcal{D}}[x = y]$, and for $S \subseteq \mathrm{Supp}(\mathcal{D})$, denote $\mathcal{D}(S) := \mathsf{Pr}_{x \leftarrow\!\!\$\ \mathcal{D}}[x \in S]$. For any set T and any function $F : \mathrm{Supp}(\mathcal{D}) \to T$, denote $F(\mathcal{D})$ as the distribution of $F(x)$ for x sampled from $\mathcal{D}$.

For any vector space V over a field F and a set $S \in V$, we denote $\mathsf{Span}_F(S)$ as the F-span of S, which is the smallest F-subspace of V that contains S. In particular, we omit F from the subscript if $F = \mathbb{R}$. For a finite set $S = \{v_1, \ldots, v_n\} \subseteq V$, we say S is F-linearly independent if and only if for any non-zero $(a_1, \ldots, a_n) \in F^n$, $\sum_{i \in [n]} a_i v_i \neq 0$. We say S is a F-basis of V if and only if S is F-linearly independent and $\mathsf{Span}_F(S) = V$. When F is not specified, we assume $F = \mathbb{R}$. The dimension of V is equal to the size of S.

For any integer $p > 0$ and any $x \in \mathbb{Z}_p$, denote $\bar{x} \in \mathbb{Z}$ to be the lift of x such that $\bar{x} \in [0..p-1]$ and $\bar{x} = x \mod p$. We use $\lfloor \cdot \rceil : \mathbb{R} \to \mathbb{Z}$ to denote the rounding operator that maps any $x \in \mathbb{R}$ to $\lfloor x + 1/2 \rfloor$. For any integers $v > 0$ and $q > 2^v$, denote $q_v = \lfloor q/2^v \rceil$ and denote $\lfloor \cdot \rceil_v : \mathbb{Z}_q \to \mathbb{Z}_{q_v}$ a function that maps $x \in \mathbb{Z}_q$ to $\lfloor \bar{x}/2^v \rceil \in \mathbb{Z}_{q_v}$. These operations can be extended an element x in R or R_q by applying them to each coefficient of x.

3.2 Polynomial Rings

Let q be an odd prime and N be a power of 2. We denote the ring $R := \mathbb{Z}[X]/(X^N + 1)$, contained in the cyclotomic field $K := \mathbb{Q}[X]/(X^N + 1)$, and let $R_q := R/qR \cong \mathbb{Z}_q[X]/(X^N + 1)$. Denote $K_\mathbb{R} := \mathbb{R} \otimes K \cong \mathbb{R}[X]/(X^N + 1)$. For an element $v \in K_\mathbb{R}$, where $v = \sum_{i=0}^{N-1} v_i X^i$, we denote its conjugate as $v^* = \sum_{i=0}^{N-1} -v_i X^{N-i}$. We use ϕ to denote the coefficient embedding that embeds $K_\mathbb{R}$ in $\mathbb{R}^N$, and ϕ maps v to vector $(v_0, \ldots, v_{N-1}) \in \mathbb{R}^N$. When applying ϕ to a vector $\boldsymbol{v} \in K_\mathbb{R}^m$, ϕ maps $\boldsymbol{v}$ to a vector in $\mathbb{R}^{mN}$ by applying ϕ to each entry of $\boldsymbol{v}$. The map ϕ is a bijection, and we denote its inverse by ϕ^{-1}. An ℓ_p-norm of $\boldsymbol{v} \in K_\mathbb{R}^m$ is given by

$$\|\boldsymbol{v}\|_p := \|\phi(\boldsymbol{v})\|_p = \left(\sum_{i=1}^{m} \sum_{j=0}^{N-1} |v_{i,j}|^p \right)^{\frac{1}{p}},$$

where $v_{i,j}$ denotes the coefficient of X^j of the i-th entry of $\boldsymbol{v}$. Additionally, the ℓ_∞-norm of $\boldsymbol{v}$ is defined as $\|\boldsymbol{v}\|_\infty := \max_{i \in [m], j \in [0..N-1]} |v_{i,j}|$. For the ℓ_2-norm, we omit the subscript and denote $\|\boldsymbol{v}\|$ as the ℓ_2-norm of $\boldsymbol{v}$. Denote the conjugate transpose of $\boldsymbol{v} \in K_\mathbb{R}^m$ as $\boldsymbol{v}^\dagger := (\boldsymbol{v}^*)^T$. We define the inner product of two vectors $\boldsymbol{v}, \boldsymbol{v'} \in K_\mathbb{R}^m$ as $\langle \boldsymbol{v}, \boldsymbol{v'} \rangle := \phi(\boldsymbol{v})^T \phi(\boldsymbol{v'}) = \langle \phi(\boldsymbol{v}), \phi(\boldsymbol{v'}) \rangle$. We have $\|\boldsymbol{v}\| = \langle \boldsymbol{v}, \boldsymbol{v} \rangle$. We say $\boldsymbol{v}$ is a unit vector if $\|\boldsymbol{v}\| = 1$.

Also, we define a map ϕ_M that maps each element in $K_\mathbb{R}$ to a matrix in $\mathbb{R}^{N \times N}$ as follows. Let $M_X := \begin{pmatrix} \mathbf{0} & -1 \\ \mathbb{I}_{N-1} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^N$, where I_{N-1} is the identity matrix in $\mathbb{R}^{N-1}$. For each $v \in K_\mathbb{R}$, $\phi_\mathsf{M}(v) := \sum_{i=0}^{N-1} v_i M_X^i$, which can be viewed as the matrix representation of v. In particular, for ϕ

and ϕ_{M}, the following properties hold: for any $v, v' \in K_{\mathbb{R}}$, $\phi_{\mathsf{M}}(v^*) = \phi_{\mathsf{M}}(v)^T$, $\phi_{\mathsf{M}}(vv') = \phi_{\mathsf{M}}(v)\phi_{\mathsf{M}}(v')$ and $\phi_{\mathsf{M}}(v)\phi(v') = \phi(vv')$. We extend the above definitions to R_q by representing each $v \in R_q$ as $v = \sum_{i=0}^{N-1} v_i X^i$, where $v_i \in \{-(q-1)/2, \ldots, (q-1)/2\}$.

For a matrix $M \in K_{\mathbb{R}}^{m\times m}$, we denote its conjugate transpose as $M^\dagger = (M^*)^T$, and we say M is *hermitian* if $M = M^\dagger$. We say M is *positive definite* if and only if M is hermitian and for all $\boldsymbol{x} \in K_{\mathbb{R}}^m \backslash \{\mathbf{0}\}$, $\langle \boldsymbol{x}, M\boldsymbol{x}\rangle > 0$, or equivalently, $\phi_{\mathsf{M}}(M)$ is positive definite. Also, denote $\sigma_{\min}(M) := \inf_{\boldsymbol{x}\in K_{\mathbb{R}}^m, \|\boldsymbol{x}\|=1} \langle \boldsymbol{x}, M\boldsymbol{x}\rangle$ as the smallest singular value of M and $\sigma_{\max}(M) := \sup_{\boldsymbol{x}\in K_{\mathbb{R}}^m, \|\boldsymbol{x}\|=1} \langle \boldsymbol{x}, M\boldsymbol{x}\rangle$ as the largest singular value of M.

We borrow the following lemma from [BCK+14], which establishes the property of the set of signed monomials $\mathcal{S}_{\mathsf{b}} := \{\pm 1, \ldots, \pm X^{N-1}\} \subseteq R_q$.

Lemma 1 (Lemma 3.1 of [BCK+14]). *Let $\mathcal{S}_{\mathsf{b}} := \{\pm 1, \ldots, \pm X^{N-1}\} \subseteq R_q$. For any $b, \bar{b} \in \mathcal{S}_{\mathsf{b}}$ such that $b \neq \bar{b}$, there exists $\gamma \in R$ such that $(b - \bar{b})\gamma = 2 \mod q$ and γ is a polynomial with coefficients only in $\{-1, 0, 1\}$.*

3.3 Lattices and Discrete Gaussian Distributions

In this subsection, we give definitions for lattices and discrete Gaussian distributions over $\mathbb{R}$ and $K_{\mathbb{R}}$. An m-dimensional lattice Λ over $\mathbb{Z}$ (resp. R) is a discrete additive subgroup of $\mathbb{Z}$ (resp. R). Equivalently, $\Lambda = \mathcal{L}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}) := \{\sum_{i\in[k]} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z}\}$ for a set of linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k \in \mathbb{Z}^m$ (resp. R^m), which is referred to as a basis of Λ. The size k is the *rank* of the lattice Λ. We say Λ is a *full rank* lattice if $k = m$ (resp. $k = mN$ for Λ over R). For any $a \in \mathbb{Z}^m$ (resp. R^m), $\Lambda + a$ is a *coset* of Λ. The *dual lattice* of Λ is denoted as $\Lambda^* = \{\boldsymbol{x} \in \mathsf{Span}(\Lambda) : \forall\ \boldsymbol{y} \in \Lambda, \langle \boldsymbol{x}, \boldsymbol{y}\rangle \in \mathbb{Z}\}$. A Λ-subspace is the linear span of some subset of Λ, i.e., a subspace S such that $S = \mathsf{Span}(S \cap \Lambda)$.

For a matrix $A \in R_q^{k\times m}$, we define the R-lattice $\Lambda_q^\perp(A) \subseteq R^m$ as

$$\Lambda_q^\perp(A) := \{\boldsymbol{x} \in R^m : A\boldsymbol{x} = 0 \bmod q\}\ .$$

We know $\Lambda_q^\perp(A)$ has full-rank since $qR^m \subseteq \Lambda_q^\perp(A)$.

For a positive definite matrix $\Sigma \in \mathbb{R}^{m\times m}$ (resp. an invertible positive definite matrix $\Sigma \in K_{\mathbb{R}}^{m\times m}$) and a vector $\boldsymbol{c} \in \mathbb{R}^n$ (resp. $K_{\mathbb{R}}^m$), we define the function $\rho_{\Sigma,\boldsymbol{c}}$ over $\mathbb{R}^m$ (resp. $K_{\mathbb{R}}^m$) as

$$\rho_{\Sigma,\boldsymbol{c}}(\boldsymbol{x}) := \exp\left(-\pi \left\langle \boldsymbol{x} - \boldsymbol{c}, \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{c})\right\rangle\right)\ .$$

Then, we denote $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}$ as the discrete Gaussian distribution over a lattice coset $\Lambda + \boldsymbol{a} \subseteq \mathbb{Z}^m$ (resp. R^m) with covariance matrix Σ, centered at $\boldsymbol{c} \in \mathbb{R}^m$, where for $\boldsymbol{x} \in \Lambda + \boldsymbol{a}$, we define

$$\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}(\boldsymbol{x}) := \mathsf{Pr}[\boldsymbol{x} \leftarrow\!\!\$\ \mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}] = \frac{\rho_{\Sigma,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{\Sigma,\boldsymbol{c}}(\Lambda + \boldsymbol{a})}$$

where $\rho_{\Sigma,\boldsymbol{c}}(\Lambda + \boldsymbol{a}) = \sum_{\boldsymbol{x}\in\Lambda+\boldsymbol{a}} \rho_{\Sigma,\boldsymbol{c}}(\boldsymbol{x})$. For $\Lambda + \boldsymbol{a} \subseteq R^m$, we denote $\mathscr{D}^{m,\bmod q}_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}(\boldsymbol{x})$ as the distribution of $(\boldsymbol{x} \bmod q) \in R_q^m$ for $\boldsymbol{x}$ sampled from $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}$.

Also, we make some remarks about the notations we will use throughout the paper. When $\Sigma = \sigma^2 \mathbb{I}_m$ for $\sigma \in \mathbb{R}$, we will use $\rho_{\sigma,\boldsymbol{c}}$ and $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\sigma,\boldsymbol{c}}$ as $\rho_{\Sigma,\boldsymbol{c}}$ and $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}$, respectively. If the center $\boldsymbol{c} = 0$, then we omit the subscript $\boldsymbol{c}$ from $\rho_{\Sigma,\boldsymbol{c}}$ and $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}$. Moreover, when $\Lambda + \boldsymbol{a} = \mathbb{Z}^m$ (resp. $\Lambda + \boldsymbol{a} = R^m$), we omit $\Lambda + \boldsymbol{a}$ from the subscript of $\mathscr{D}^m_{\Lambda+\boldsymbol{a},\Sigma,\boldsymbol{c}}$.

The smoothing parameter of a lattice Λ with respect to $\varepsilon > 0$, denoted by $\eta_\varepsilon(\Lambda)$, is the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^* \backslash \{0\}) \leqslant \varepsilon$. Throughout the paper, we set $\varepsilon = 2^{-2\kappa}$.

Game $\mathrm{MSIS}^{\mathcal{A}}_{q,N,k,m,\beta}$:	Game $\mathrm{MLWE}^{\mathcal{A}}_{q,N,k,m,\beta}$:
$\overline{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$	$\overline{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\overline{A}\|\mathbb{I}_k]$
$A \leftarrow [\overline{A}\|\mathbb{I}_k]$	$\boldsymbol{s} \leftarrow\!\!\$\ \mathscr{B}_\beta^m$; $\boldsymbol{t}_0 \leftarrow A\boldsymbol{s}$; $\boldsymbol{t}_1 \leftarrow\!\!\$\ R_q^k$
$\boldsymbol{x} \leftarrow \mathcal{A}(A)$	$b \leftarrow\!\!\$\ \{0,1\}$
Return $(\|\boldsymbol{x}\| \leqslant \beta \ \wedge\ A\boldsymbol{x} = 0)$	$\hat{b} \leftarrow \mathcal{A}(A, t_b)$
	Return $(\hat{b} = b)$

Fig. 1. The module-SIS and module-LWE problems, where $R := \mathbb{Z}[X]/(X^N+1)$, $R_q := R/qR$ and $\mathscr{B}^m_{\beta_{\mathsf{lwe}}} := \{\boldsymbol{x} \in R^m \mid \|\boldsymbol{x}\|_\infty \leqslant \beta_{\mathsf{lwe}}\}$.

We borrow the following lemma from [AGHS13] that bounds the ℓ_2-norm of discrete Gaussian random variables and adapt it to lattices over $K_{\mathbb{R}}$.

Lemma 2 (Lemma 3 of [AGHS13] adapted to $K_{\mathbb{R}}$). *For any $\varepsilon \in (0,1)$, a lattice $\Lambda \subseteq R^m$, $\boldsymbol{c} \in K^m_{\mathbb{R}}$, and $\sigma \geqslant \eta_\varepsilon(\Lambda)$, then*

$$\mathsf{Pr}[\|\boldsymbol{x}-\boldsymbol{c}\| \geqslant \sigma\sqrt{mN} \ : \ \boldsymbol{x} \leftarrow\!\!\$\ \mathscr{D}_{\Lambda,\sigma,\boldsymbol{c}}] \leqslant \frac{1+\varepsilon}{1-\varepsilon}\cdot 2^{-mN} .$$

We also borrow the following lemma from [GPV08] that show upper bounds of smoothing parameters for general lattices.

Lemma 3 (Lemma 2.6 of [GPV08]). *For any full-rank lattice Λ in $\mathbb{R}^m$ and $\varepsilon > 0$, $\eta_\varepsilon(\Lambda) \leqslant \frac{\sqrt{\log(2m(1+1/\varepsilon))/\pi}}{\lambda_1^\infty(\Lambda^*)}$, where $\lambda_1^\infty(\Lambda^*)$ denotes the ℓ_2 norm of the shortest non-zero vector in the ℓ_∞ norm in the dual lattice Λ^*.*

3.4 Assumptions

We recall the module short integer solution (MSIS) problem and the module learning with error (MLWE) problem (defined in Figure 1). The advantage of $\mathcal{A}$ for the MSIS problem is defined as

$$\mathsf{Adv}^{\mathrm{msis}}_{q,N,k,m,\beta}(\mathcal{A}) := \mathsf{Pr}\left[\mathrm{MSIS}^{\mathcal{A}}_{q,N,k,m,\beta} = 1\right] .$$

The advantage of $\mathcal{A}$ for the MLWE problem is defined as

$$\mathsf{Adv}^{\mathrm{mlwe}}_{q,N,k,m,\beta}(\mathcal{A}) := \mathsf{Pr}\left[\mathrm{MLWE}^{\mathcal{A}}_{q,N,k,m,\beta} = 1\right] .$$

3.5 Rényi Divergence

We define the notion of Rényi Divergence between two distributions P, Q which we will use in our analysis of the scheme.

Definition 1 (Rényi Divergence). *Let P, Q be two discrete probability distributions such that $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. We define the Rényi Divergence of order α, for $\alpha \in (1,\infty)$ as $R_\alpha(P\|Q) := \left(\sum_{x\in Supp(P)} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}}\right)^{\frac{1}{\alpha-1}}$.*

The following lemma gives basic properties of the Rényi Divergence.

Lemma 4 (Lemma 4.1 of [LSS14]). *Let $\alpha \in (1, \infty)$ and P, Q be discrete probability distributions with $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. Then, the following properties hold:*

- ***Data Processing Inequality:*** *$R_\alpha\left(P^f \| Q^f\right) \leqslant R_\alpha\left(P \| Q\right)$ for any function f, where P^f (and Q^f) denotes the distribution which samples $x \leftarrow\!\!\$\ P$ ($x \leftarrow\!\!\$\ Q$) and outputs $f(x)$.*
- ***Probability Preservation:*** *Let $E \subseteq \mathrm{Supp}(Q)$ be an arbitrary event. Then, for $\alpha \in (1, \infty)$,*

$$\mathsf{Pr}_{x \leftarrow\!\!\$\ P}[E] \leqslant \left(\mathsf{Pr}_{x \leftarrow\!\!\$\ Q}[E] R_\alpha\left(P \| Q\right)\right)^{(\alpha-1)/\alpha} .$$

Also, we show the following property.

Lemma 5. *Let P and Q denote two distributions over the union of a countable number of disjoint sets $\{S_i\}_{i \in U}$ such that $P(S_i) = Q(S_i)$ for any $i \in U$. Then, for any $\alpha > 1$, if there exists δ such that $R_\alpha\left(P_{|S_i} \| Q_{|S_i}\right) \leqslant \delta$ for any $i \in U$, then $R_\alpha\left(P \| Q\right) \leqslant \delta$.*

Proof.

$$\begin{aligned} R_\alpha\left(P \| Q\right)^{\alpha-1} &= \sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}} = \sum_{i \in U} \sum_{x \in S_i} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}} \\ &= \sum_{i \in U} Q(S_i) \sum_{x \in S_i} \frac{(P(x)/P(S_i))^\alpha}{(Q(x)/Q(S_i))^{\alpha-1}} \\ &= \sum_{i \in U} Q(S_i) R_\alpha\left(P_{|S_i} \| Q_{|S_i}\right)^{\alpha-1} \leqslant \sum_{i \in U} Q(S_i) \delta^{\alpha-1} = \delta^{\alpha-1} , \end{aligned}$$

which concludes the lemma. □

We borrow the following lemma from [TT15], which upperbounds the Rényi Divergence between two discrete Gaussian distributions with different centers.

Lemma 6 (Lemma 5 of [TT15]). *For any m-dimensional lattice $\Lambda \subseteq \mathbb{R}^m$, $\sigma > 0$, and two vectors $\boldsymbol{c}, \boldsymbol{c}' \in \mathbb{R}^m$, let $P = \mathscr{D}^m_{\Lambda,\sigma,\boldsymbol{c}}$ and $Q = \mathscr{D}^m_{\Lambda,\sigma,\boldsymbol{c}'}$. If $\boldsymbol{c}, \boldsymbol{c}' \in \Lambda$, set $\varepsilon = 0$. Otherwise, fix $\varepsilon \in (0, 1)$ and assume $\sigma \geqslant \eta_\varepsilon(\Lambda)$. Then,*

$$R_\alpha\left(P \| Q\right) \leqslant \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \exp\left(\alpha\pi \frac{\|\boldsymbol{c} - \boldsymbol{c}'\|^2}{\sigma^2}\right) .$$

3.6 Threshold signatures

We use the formalization proposed by Bellare et al. [BCK+22], which is also used in [CATZ24, TZ23].

Syntax. A (partially) non-interactive threshold signature schemes for n signers and threshold t is a tuple of efficient (randomized) algorithms $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{SPP}, \mathsf{LPP}, \mathsf{LR}, \mathsf{PS}, \mathsf{Agg}, \mathsf{Vf})$ that behave as follows. An execution of the scheme involves a leader as well as n signers. The setup algorithm $\mathsf{Setup}(1^\kappa)$ initializes the state st_i for each signer $i \in [n]$, as well as the state st_0 for the leader, and additionally a public parameter *par*. (We assume *par* is given as implicit input to all other algorithms.) The key generation algorithm $\mathsf{KeyGen}()$ returns a public verification key pk, and

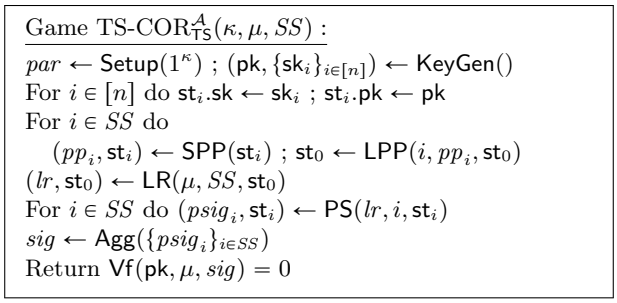

Fig. 2. The TS-COR game for a threshold signature scheme TS for n signers and threshold t.

a secret key sk_i for each signer i. (This means in particular that we assume ideal key generation, and as in prior works, do not model distributed key generation.)

The signing protocol proceeds in two rounds: In a first, message-independent offline round, any signer i can run $\mathsf{SPP}(\mathsf{st}_i)$ to generate a *pre-processing token* pp, which is sent to the leader, and the leader runs $\mathsf{LPP}(i, pp, \mathsf{st}_0)$ to update its state st_0 to incorporate token pp. In the second, online, round, for a signer set $SS \subseteq [n]$ with size t and message $\mu \in \{0,1\}^*$, the leader runs $\mathsf{LR}(\mu, SS, \mathsf{st}_0)$ to generate a leader request lr with $lr.\mathsf{msg} = \mu$ and $lr.\mathsf{SS} = SS$ and sends lr to each signer $i \in SS$. Then, each signer i runs $\mathsf{PS}(lr, i, \mathsf{st}_i)$ to generate its partial signature $psig_i$. Finally, the leader computes a signature sig for μ by running $\mathsf{Agg}(\{psig_i\}_{i\in SS})$. The (deterministic) verification algorithm $\mathsf{Vf}(\mathsf{pk}, \mu, sig)$ outputs a bit that indicates whether sig is valid for (pk, μ).

An honest execution of the signing protocol to sign a message $\mu \in \{0,1\}^*$ is represented in the correctness game TS-COR (defined in Figure 2). We say that TS is *correct* with correctness error $\varepsilon_{\mathsf{cor}}$ if for $\mu \in \{0,1\}^*$ and $SS \subseteq [n]$ with $|SS| \geqslant t$, we have $\mathsf{Pr}[\text{TS-COR}_{\mathsf{TS}}(\kappa, \mu, SS) = 1] \leqslant \varepsilon_{\mathsf{cor}}$.

SECURITY. Roughly speaking, unforgeability ensures that an adversary cannot forge a signature for a message that has not been considered as signed, considering the corruption status and the adversary's interactions with honest parties. A hierarchy for unforgeability of threshold signatures is proposed in [BCK+22]. In this paper, we consider TS-UF-0 and TS-UF-4, which differ only in how they define when a message is considered as signed. In particular, for TS-UF-0, a message is considered as signed only if the adversary received a partial signature from at least one honest signer for μ, and for TS-UF-4, the condition is much stronger - a message is considered as signed only if there exists a leader request lr for μ such that the adversary received partial signatures from all honest parties in $lr.\mathsf{SS}$ for that leader request lr. Formally, the TS-UF-0 and TS-UF-4 games are defined in Figure 3, where $\mathsf{TS.HF}$ denotes the space of the hash functions used in TS from which the random oracle is drawn. The advantage of $\mathcal{A}$ for the TS-UF-X game is defined as $\mathsf{Adv}^{\text{ts-uf-X}}_{\mathsf{TS}}(\mathcal{A}, \kappa) := \mathsf{Pr}\left[\text{TS-UF-X}^{\mathcal{A}}_{\mathsf{TS}}(\kappa) = 1\right]$ for $\mathsf{X} \in \{0, 4\}$.

4 Algebraic One-More MISIS Problem

In this section, we first formally define the algebraic one-more MISIS (AOM-MISIS) problem, then provide a comparison with the AOM-MLWE problem, and finally present a formal reduction from standard lattice problems to AOM-MISIS.

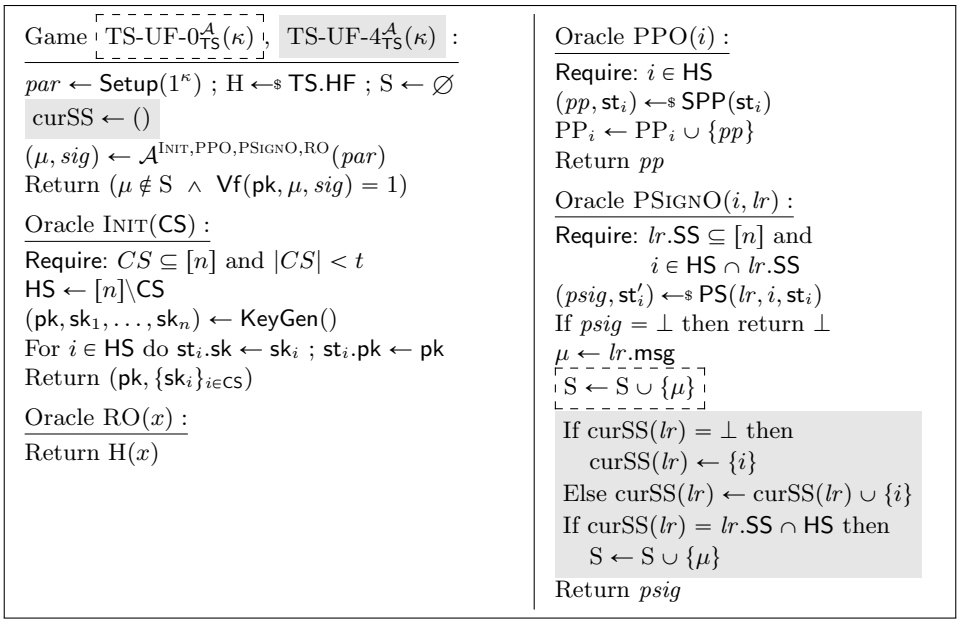

Fig. 3. The game TS-UF-0 and the game TS-UF-4 for a threshold signature scheme TS for n signers and threshold t, where TS-UF-0 contains all but the highlight boxes and TS-UF-4 contains all but the dashed boxes. In particular, in TS-UF-4, a table curSS is maintained to record, for each lr, the set of signers that have received a partial signing query on lr. A message μ is considered signed if and only if $\mathrm{curSS}(lr)$ contains the set of honest parties in $lr.\mathsf{SS}$.

4.1 Definition of AOM-MISIS

The algebraic one-more MISIS game is defined in Figure 4. The game is defined implicitly over the cyclotomic ring $R = \mathbb{Z}[X]/(X^N + 1)$, where N is a power of two, as well as the associated ring $R_q = R/qR \cong \mathbb{Z}_q[X]/(X^N + 1)$ for an odd prime q. The adversary $\mathcal{A}$ is given Q MISIS challenges $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_Q$ and the goal is to output a short solution $\hat{\boldsymbol{s}}$ for a linear combination of the challenges $\sum_{i \in [Q]} \hat{b}_i \boldsymbol{t}_i$, where the norm of $\hat{\boldsymbol{b}}$ is suitably bounded. The adversary can also issue queries to the oracle PI which reveal linear combinations of the secrets. To win, the adversary also needs to output an additional vector $\boldsymbol{u}$ which is orthogonal to all queries to PI, but not to the vector $\hat{\boldsymbol{b}}$. We can also view the existence of such $\boldsymbol{u}$ as a constraint on the PI queries. We refer the reader to the introduction for some additional intuition. Here, for $\mathsf{par} = (q, N, k, m, Q, (\sigma_i)_{i \in [Q]}, \beta_\mathsf{s}, \beta_\mathsf{b}, \beta_\mathsf{u})$, we define the *advantage* of an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{A}, \kappa) = \mathsf{Pr}\left[\text{AOM-MISIS}^{\mathcal{A}}_{\mathsf{par}}(\kappa) = 1\right] . \tag{2}$$

We note that we slightly abuse notation in the asymptotic definition of the game, since it is understood that all parameters grow with κ, including Q, and thus the notation $(\sigma_i)_{i \in Q}$ is not entirely well-defined. This will not be an issue in actual use cases, and we dispense with a more rigorous definition.

Remark 1. We note that for each $i \in [Q]$, it is equivalent to sample $\boldsymbol{s}_i$ from $\mathscr{D}^{m, \bmod q}_{\sigma_i}$ instead of $\mathscr{D}^m_{\sigma_i}$, since the view of $\mathcal{A}$ remains unchanged. With this insight, it is natural to allow σ_i to be ∞, in which case $\boldsymbol{s}_i$ is sampled uniformly from R^m_q, $1/\sigma_i = 0$, and we require $\hat{b}_i = 0$ in the winning

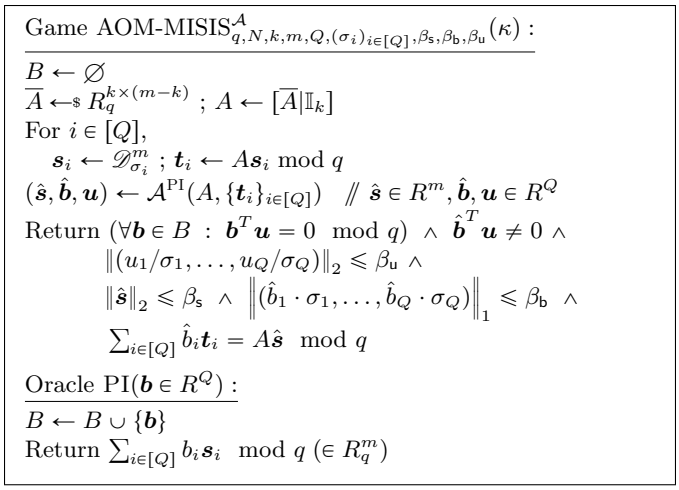

Fig. 4. The AOM-MISIS game, where $R := \mathbb{Z}[X]/(X^N + 1)$ and $R_q := R/qR$.

condition (since, o.w., $\left\|\hat{b}_i \cdot \sigma_i\right\|_1$ is ∞). This extension is used, in particular, in our security reduction of the CTZ construction (see Section 5.1), where the random values for generating the key shares are sampled uniformly from R_q^m.

4.2 Comparison with prior work

We present in Figure 5 the AOM-MLWE problem proposed by Espitau et al. [EKT24]. In our formulation, the PI oracle corresponds to the $\mathcal{O}_{\mathsf{solve}}$ oracle in their notation, and the constraints on PI queries are explicitly stated within the game. More precisely, the constraints are:

- The number of PI queries is exactly $Q - 1$.
- Denote $\begin{pmatrix} \boldsymbol{v}^T \\ \underline{D} \end{pmatrix} = [\boldsymbol{d}_1 | \cdots | \boldsymbol{d}_{Q-1}]$, where $\boldsymbol{d}_i$ denotes the i-th PI query. $\underline{D}$ is an invertible matrix.
- Let $\boldsymbol{w} \leftarrow (\boldsymbol{v}^T\underline{D}^{-1})^T$. The ℓ_2-norm of each entry of $\boldsymbol{w}$ is bounded by β_{d}.

We also define, for $\mathsf{par} = (q, N, k, m, Q, (\sigma_i)_{i\in[Q]}, \beta_{\mathsf{s}}, \beta_{\mathsf{e}}, \beta_{\mathsf{b}}, \beta_{\mathsf{d}})$, the corresponding advantage

$$\mathsf{Adv}^{\text{aom-mlwe}}_{\mathsf{par}}(\mathcal{A}, \kappa) = \mathsf{Pr}\left[\text{AOM-MLWE}^{\mathcal{A}}_{\mathsf{par}}(\kappa) = 1\right] .$$

Compared with the AOM-MLWE problem, our problem differs in the following key aspects:

1. Our problem only asks the adversary to output one special solution $\hat{\boldsymbol{s}}$ for a linear combination of the challenges, while the AOM-MLWE problem demands one solution for each challenge. This makes our problem inherently easier to solve, as noted at the beginning of Section 2. Moreover, this relaxation simplifies the security reduction, as it only needs to extract one solution rather than multiple, making it easier to reduce from our assumption. Additionally, it enables better parameter selections, since the norm bound applies to a single solution instead of n solutions.
2. The constraints on the PI queries differ between AOM-MLWE and our problem. In particular, their constraints are a special case of ours: given the PI queries satisfying their constraints, we can verify that these queries also satisfy our constraints by setting $\boldsymbol{u} = (1, -\boldsymbol{w})$, since $\boldsymbol{u}\begin{pmatrix} \boldsymbol{v}^T \\ \underline{D} \end{pmatrix} = 0$. We will show this formally later in this section.

Game $\text{AOM-MLWE}^{\mathcal{A}}_{q,k,m,Q,(\sigma_i)_{i\in[Q]},\beta_{\mathsf{s}},\beta_{\mathsf{e}},\beta_{\mathsf{b}},\beta_{\mathsf{d}}}(\kappa)$:

$\mathsf{cnt} \leftarrow 0$
$\overline{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\overline{A}|\mathbb{I}_k]$
For $i \in [Q]$,
  $\boldsymbol{s}_i \leftarrow \mathscr{D}^m_{\sigma_i}$; $\boldsymbol{t}_i \leftarrow A\boldsymbol{s}_i \bmod q$
$(\hat{\boldsymbol{b}}, \hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_Q) \leftarrow \mathcal{A}^{\text{PI}}(A, \{\boldsymbol{t}_i\}_{i\in[Q]})$ // $\hat{\boldsymbol{s}} \in R^m, \hat{\boldsymbol{b}}$
If $\mathsf{cnt} \neq Q-1$ then return 0
Parse $\begin{pmatrix} \boldsymbol{v}^T \\ \underline{D} \end{pmatrix} \leftarrow [\boldsymbol{d}_1|\cdots|\boldsymbol{d}_{Q-1}]$
If $\underline{D}^{-1}$ does not exist then return 0
$\boldsymbol{w} \leftarrow (\boldsymbol{v}^T\underline{D}^{-1})^T$
Return $(\forall i \in [Q-1] \ : \ \|w_i\| \leqslant \beta_{\mathsf{d}}) \ \wedge$
  $\left\|(\hat{s}_{1,[m-k]}, \ldots, \hat{s}_{Q,[m-k]})\right\|_2 \leqslant \beta_{\mathsf{s}} \ \wedge$
  $\left\|(\hat{s}_{1,[(m-k+1)..m]}, \ldots, \hat{s}_{Q,[(m-k+1)..m]})\right\|_2 \leqslant \beta_{\mathsf{e}} \ \wedge$
  $(\forall i \in [Q] \ : \ b_i \neq 0 \ \wedge \ \|b_i\| \leqslant \beta_{\mathsf{b}} \ \wedge \ \hat{b}_i\boldsymbol{t}_i = A\hat{\boldsymbol{s}}_i \bmod q)$

Oracle $\text{PI}(\boldsymbol{b} \in R^Q)$:

$\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$
$\boldsymbol{d}_{\mathsf{cnt}} \leftarrow \boldsymbol{b}$
Return $\sum_{i\in[Q]} b_i\boldsymbol{s}_i$

Fig. 5. The AOM-MLWE game.

3. The norm bounding approach is different. In AOM-MLWE, the norm of each entry of $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{w}}$ is bounded individually. In contrast, we bound the norm of the entire vectors $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{u}}$, with each entry weighted by the standard deviation of the corresponding challenge. This adjustment leads to better parameter selections.

In short, our problem is easier to reduce from and enables better parameter selections. In Lemma 7, we formally show that the hardness of our problem implies the hardness of theirs, and we discuss the improvement in the parameter selections in Section 6.1, Remark 3.

Remark 2. We note that Espitau et al. [EKT24] propose an alternative way of defining constraints, which, like ours, requires the existence of a nonzero vector $\boldsymbol{u}$ that is orthogonal to all PI queries. The key difference is that they impose a bound on the norm of each entry of $\boldsymbol{u}$. We can show that the hardness of our problem also implies the hardness of theirs under these constraints. However, we omit a formal analysis here, as the proof idea is very similar, and this alternative version is not used to establish the security of their threshold signature scheme.

Lemma 7. *For any* $\mathsf{par} = (q, N, k, m, Q, (\sigma_i)_{i\in[Q]}, \beta_{\mathsf{s}}, \beta_{\mathsf{e}}, \beta_{\mathsf{b}}, \beta_{\mathsf{d}})$ *and any adversary* $\mathcal{A}$ *playing the game* $\text{AOM-MLWE}^{\mathcal{A}}_{\mathsf{par}}$*, there exists an* AOM-MISIS *adversary* $\mathcal{B}$ *running in time roughly the same as* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\text{aom-mlwe}}_{\mathsf{par}}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}'}(\mathcal{B}, \kappa) \ ,$$

where $\mathsf{par}' = (q, N, k, m, Q, (\sigma_i)_{i\in[Q]}, \beta'_{\mathsf{s}} = \beta_{\mathsf{s}} + \beta_{\mathsf{e}}, \beta'_{\mathsf{b}} = \sqrt{N}\beta_{\mathsf{b}}\sigma_1, \beta_{\mathsf{u}} = 1/\sigma_1 + \beta_{\mathsf{d}}\sqrt{Q}/(\min_{i\in[2..Q]}\sigma_i))$.

Roughly, the proof idea is that for an adversary $\mathcal{A}$ that wins the game AOM-MLWE, $\mathcal{A}$ can win the game AOM-MISIS by outputting only the solution to the first challenge (i.e., outputting $\hat{s}_1$ and $(\hat{b}_1, 0, \ldots, 0)$) and outputting $\boldsymbol{u} \leftarrow (1, -\boldsymbol{v}^T\underline{D}^{-1})$. It is not hard to check $\boldsymbol{u}$ satisfies the constraints of the game AOM-MISIS. We do this explicitly in the following proof.

Proof. For any adversary $\mathcal{A}$ described in the lemma, we construct $\mathcal{B}$ as follows. To start with, $\mathcal{B}$ runs $\mathcal{A}$ on its input $A, \{\boldsymbol{t}_i\}_{i\in[Q]}$ by forwarding all PI queries from $\mathcal{A}$ to its own PI oracle. After receiving $\mathcal{A}$'s output $(\hat{\boldsymbol{b}}, \hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_Q)$, if $\mathcal{A}$ wins the AOM-MLWE game simulated by $\mathcal{B}$, $\mathcal{B}$ returns $((\hat{b}_1, 0, \ldots, 0), \hat{\boldsymbol{s}}_1, (1, -\boldsymbol{w}^T))$, where $\boldsymbol{w}$ is defined in the AOM-MLWE game. Otherwise $\mathcal{B}$ aborts.

We now show $\mathcal{B}$ wins the AOM-MISIS game given $\mathcal{A}$ wins by checking all the winning conditions of $\mathcal{B}$.

- For the first condition, since $(1, -\boldsymbol{w}^T) \cdot \begin{pmatrix} \boldsymbol{v}^T \\ \underline{D} \end{pmatrix} = \boldsymbol{v}^T - \boldsymbol{v}^T \underline{D}^{-1} \underline{D} = 0$, for any $i \in [Q-1]$, $(1, -\boldsymbol{w}^T) \cdot \boldsymbol{d}_i = 0$, so the first condition is satisfied.
- For the second condition, $(\hat{b}_1, 0, \ldots, 0) \begin{pmatrix} 1 \\ -\boldsymbol{w} \end{pmatrix} = \hat{b}_1 \neq 0$.
- For the third condition, $\|(1/\sigma_1, w_1/\sigma_2, \ldots, w_{Q-1}/\sigma_Q)\| \leqslant \frac{1}{\sigma_1} + \sum_{i\in[Q-1]} \frac{\|w_i\|}{\sigma_i} \leqslant \frac{1}{\sigma_1} + \frac{\beta_{\mathsf{d}}}{\min_{i\in[2..Q]}\sigma_i} = \beta_{\mathsf{u}}$.
- For the fourth condition, $\|\hat{\boldsymbol{s}}_1\| \leqslant \left\|\hat{s}_{1,[m-k]}\right\| + \left\|\hat{s}_{1,[(m-k+1)..k]}\right\| \leqslant \beta_{\mathsf{s}} + \beta_{\mathsf{e}} = \beta'_{\mathsf{s}}$.
- For the fifth condition, $\left\|(\hat{b}_1 \cdot \sigma_1, 0, \ldots, 0)\right\|_1 \leqslant \left\|\hat{b}_1 \cdot \sigma_1\right\|_1 \leqslant \sqrt{N}\sigma_1 \left\|\hat{b}_1\right\| = \beta'_{\mathsf{b}}$.
- For the final condition, $\hat{b}_1 \boldsymbol{t}_1 = A\hat{\boldsymbol{s}} \mod q$.

□

4.3 Reduction from MSIS and MLWE

This section shows our main result establishing hardness of AOM-MISIS from the hardness of MLWE and MSIS. (We remind the reader that we also provided a detailed overview of this proof in Section 2 above.)

Theorem 1. *For any $\varepsilon \in (0,1)$, $\alpha > 1$, any $\mathsf{par} = (q, N, k, m, Q, (\sigma_i)_{i\in[Q]}, \beta_{\mathsf{s}}, \beta_{\mathsf{b}}, \beta_{\mathsf{u}})$ such that $mN \geqslant 2\kappa$ and $\sigma_i \geqslant \sqrt{\log(6mN)/\pi}$, and any* AOM-MISIS *adversary $\mathcal{A}$, there exist a* MSIS *adversary $\mathcal{B}$ and two* MLWE *adversaries $\mathcal{C}$ and $\mathcal{D}$, such that*

$$\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{A}, \kappa) \leqslant 2\delta_\alpha \left(\mathsf{Adv}^{\text{msis}}_{q,k,m,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa) + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{D}, \kappa)\right)^{\frac{\alpha}{\alpha-1}} + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{C}, \kappa) + Q \cdot 2^{-2\kappa+2} ,$$

where $\delta_\alpha = \frac{1+\varepsilon}{1-\varepsilon} \cdot \exp\left((\alpha-1)\pi^2/\log(2(1+1/\varepsilon))\right)$, $\beta_{\mathsf{sis}} = \beta_{\mathsf{s}} + \sqrt{mN}\beta_{\mathsf{b}}$, and $\beta_{\mathsf{lwe}} = 1/(\beta_{\mathsf{u}}\sqrt{mN}\sqrt{\log(2(1+1/\varepsilon))/\pi})$. The three adversaries have, roughly the same running time as that of $\mathcal{A}$.

Proof. We prove the theorem via the following series of games.

$\mathbf{G}_0$: Same as the AOM-MISIS game.

$\mathbf{G}_1$: Same as $\mathbf{G}_0$ except that A is sampled with a short solution embedded, i.e., $A \leftarrow [D\boldsymbol{a} + \boldsymbol{e} | D | \mathbb{I}_k]$ for $(\boldsymbol{a}, \boldsymbol{e}) \leftarrow\!\!\$\ \mathscr{B}^{m-1}_{\beta_{\mathsf{lwe}}}$, where $\mathscr{B}^{m-1}_{\beta_{\mathsf{lwe}}} := \{\boldsymbol{x} \in R^{m-1} | \|\boldsymbol{x}\|_\infty \leqslant \beta_{\mathsf{lwe}}\}$, and $D \leftarrow\!\!\$\ R_q^{k\times(m-k-1)}$. Since the only difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ is that $\overline{A}$ is replaced with a MLWE challenge, there exists a MLWE adversary $\mathcal{C}$ such that

$$\mathsf{Adv}^{\mathbf{G}_0}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\mathbf{G}_1}(\mathcal{A}, \kappa) + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{C}, \kappa) . \tag{3}$$

$\mathbf{G}_2$: Same as $\mathbf{G}_1$ except that the game aborts if there exists $i \in [Q]$ such that $\|\boldsymbol{s}_i\| > \sigma_i\sqrt{mN}$. To bound $\mathsf{Pr}[\mathbf{G}_2 \text{ aborts}]$, since $\boldsymbol{s}_i$ is sampled from $\mathscr{D}^m_{\sigma_i}$ and $\sigma_i \geqslant \sqrt{\log(6mN)/\pi} = \eta_{\varepsilon'}(R^m)$ with $\varepsilon' = 1/2$, by Lemma 2, $\mathsf{Pr}[\|\boldsymbol{s}_i\| > \sigma_i\sqrt{mN}] \leqslant 3 \cdot 2^{-mN} \leqslant 2^{-2\kappa+2}$. By the Union Bound, $\mathsf{Pr}[\mathbf{G}_2 \text{ aborts}] \leqslant \sum_{i\in[Q]} \mathsf{Pr}[\|\boldsymbol{s}_i\| > \sigma_i\sqrt{mN}] \leqslant Q \cdot 2^{-2\kappa+2}$. Therefore,

$$\mathsf{Adv}^{\mathbf{G}_1}(\mathcal{A}, \kappa) \leqslant \mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) + Q \cdot 2^{-2\kappa+2} \,. \tag{4}$$

Consider a variant of the MSIS game, where A is sampled with a short solution embedded (same as the above $\mathbf{G}_1$), i.e., $A \leftarrow [D\boldsymbol{a} + \boldsymbol{e}|D|\mathbb{I}_k]$ for $(\boldsymbol{a}, \boldsymbol{e}) \leftarrow\!\!\$\ \mathscr{B}^{m-1}_{\beta_{\mathsf{lwe}}}$, where $\mathscr{B}^{m-1}_{\beta_{\mathsf{lwe}}} := \{\boldsymbol{x} \in R^{m-1} | \, \|\boldsymbol{x}\|_\infty \leqslant \beta_{\mathsf{lwe}}\}$, and $D \leftarrow\!\!\$\ R_q^{k\times(m-k-1)}$. We refer to the game as td-MSIS. We construct $\mathcal{B}$ playing the td-MSIS game as follows. Given the td-MSIS challenge $A' \in R_q^{k\times(m-k)}$, $\mathcal{B}$ runs $\mathcal{A}$ by simulating the game $\mathbf{G}_2$ with $\mathcal{A}$ faithfully except that $\mathcal{B}$ sets $A \leftarrow A'$ instead of sampling A by itself. After receiving the output $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{b}}, \boldsymbol{u})$ from $\mathcal{A}$, if $\mathcal{A}$ wins the game $\mathbf{G}_2$ simulated by $\mathcal{B}$, $\mathcal{B}$ outputs $\boldsymbol{x} \leftarrow \hat{\boldsymbol{s}} - \sum_{i\in[Q]} \hat{b}_i \boldsymbol{s}_i$. Otherwise, $\mathcal{B}$ aborts.

Analysis of $\mathcal{B}$. Similar to the hybrid between $\mathbf{G}_0$ and $\mathbf{G}_1$, the td-MSIS game is computationally indistinguishable from the MSIS game under the MLWE assumption. Therefore, there exists a MLWE adversary $\mathcal{D}$ such that

$$\mathsf{Adv}^{\text{td-msis}}_{q,k,m-1,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa) \leqslant \mathsf{Adv}^{\text{msis}}_{q,k,m-1,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa) + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{D}, \kappa) \,. \tag{5}$$

Claim. $\mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) \leqslant 2\delta_\alpha \mathsf{Adv}^{\text{td-msis}}_{q,k,m-1,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa)^{(\alpha-1)/\alpha}$.

We can conclude the proof since

$$\begin{aligned}
\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{A}, \kappa) &\leqslant \mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{C}, \kappa) + Q \cdot 2^{-2\kappa+2} \\
&\leqslant 2\delta_\alpha \mathsf{Adv}^{\text{td-msis}}_{q,k,m-1,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa)^{(\alpha-1)/\alpha} + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{C}, \kappa) + Q \cdot 2^{-2\kappa+2} \\
&\leqslant 2\delta_\alpha \left(\mathsf{Adv}^{\text{msis}}_{q,k,m,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa) + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{D}, \kappa)\right)^{\frac{\alpha}{\alpha-1}} \\
&\quad + \mathsf{Adv}^{\text{mlwe}}_{q,k,m-1,\beta_{\mathsf{lwe}}}(\mathcal{C}, \kappa) + Q \cdot 2^{-2\kappa+2} \,,
\end{aligned}$$

where the first inequality follows from Equations (3) and (4), the second inequality follows from the claim, and the third inequality follows from Equation (5). □

We now prove the above claim.

Proof. Consider a fixed randomness $(\boldsymbol{a}, \boldsymbol{e}, D)$ of td-MSIS. Also, w.l.o.g. assume $\mathcal{A}$ is deterministic. Then, the execution of $\mathcal{B}$ is determined by $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q$. We define a map $\Phi_{\boldsymbol{a},\boldsymbol{e},D,\mathcal{A}} : R^{Qm} \to R^{Qm}$ as follows such that the view of $\mathcal{A}$ given $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ is exactly the same as that given $\Phi_{\boldsymbol{a},\boldsymbol{e},D,\mathcal{A}}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Consider the execution given $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. To simplify notation, we omit the subscript of Φ for the rest of the proof. If $\mathcal{A}$ does not win the $\mathbf{G}_2$ simulated by $\mathcal{B}$, we set $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Otherwise, we set $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) = (\boldsymbol{s}_1 + u_1\Delta, \ldots, \boldsymbol{s}_Q + u_Q\Delta)$, where $\boldsymbol{u} \in R^Q$ is output by $\mathcal{A}$ and $\Delta = (1, -\boldsymbol{a}, -\boldsymbol{e}) \in R^m$.

It is not hard to see that the view of $\mathcal{A}$ given $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ is identical to that given $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. In particular, in the case that AWin occurs, $\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ leads to the same view since $\sum_{i\in[Q]} b_i(\boldsymbol{s}_i + u_i\Delta) = \sum_{i\in[Q]} b_i\boldsymbol{s}_i + \Delta\sum_{i\in[Q]} b_iu_i = \sum_{i\in[Q]} b_i\boldsymbol{s}_i$ for any $(b_1, \ldots, b_Q) \in B$ and $A(\boldsymbol{s}_i + u_i\Delta) = A\boldsymbol{s}_i$ for any $i \in [Q]$ due to the fact that $A \cdot \Delta = [D\boldsymbol{a} + \boldsymbol{e}|D|\mathbb{I}_k] \cdot (1, -\boldsymbol{a}, -\boldsymbol{e})^T = \boldsymbol{0}$.

Also, it is not hard to see that Φ is a bijection. For $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \in R^{mQ}$, suppose $\Phi(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Since $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q)$ leads to the same view of $\mathcal{A}$ as $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$, we have either $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ in case that AWin does not occur, or $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) = (\boldsymbol{s}_1 - u_1\Delta, \ldots, \boldsymbol{s}_Q - u_Q\Delta)$ in case that AWin occurs and $\mathcal{A}$ outputs $\boldsymbol{u}$ given $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Also, it also shows that such $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q)$ always exists, which means Φ is a bijection.

In the game $\mathbf{G}_2$, the distribution of $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ is $\mathscr{D}_\Sigma^{mQ}$, where $\Sigma = \mathbb{I}_m \otimes \mathrm{diag}(\sigma_1^2, \ldots, \sigma_Q^2)$. Denote $P = \Phi(\mathscr{D}_\Sigma^{mQ})$. Following the idea described in Section 2.1, consider the following two adversaries $\mathcal{B}'$ and $\mathcal{B}''$.

- $\mathcal{B}'$ is the same as $\mathcal{B}$ except that $\mathcal{B}'$ samples $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ from P.
- $\mathcal{B}''$ is the same as $\mathcal{B}$ except that 1. $\mathcal{B}''$ samples $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \leftarrow_{\$} \mathscr{D}_\Sigma^{mQ}$ and computes secrets as $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \leftarrow \Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$; 2. after $\mathcal{A}$ returns, $\mathcal{B}''$ outputs $\hat{\boldsymbol{s}}_i - \boldsymbol{r}_i$ if there exists $\boldsymbol{r}_i \neq \hat{\boldsymbol{s}}_i$.

We note that $\mathcal{B}'$ and $\mathcal{B}''$ do not need to be efficient. They are only used to compute the winning probability of the (efficient) adversary $\mathcal{B}$.

Denote now by $\mathsf{PWin}_{\mathcal{X}}$ the winning probability of $\mathcal{X} \in \{\mathcal{A}, \mathcal{B}, \mathcal{B}', \mathcal{B}''\}$. Then, we will show the following three facts.

Fact 1. $\mathsf{PWin}_{\mathcal{B}'} \leqslant \delta_\alpha \mathsf{PWin}_{\mathcal{B}}^{(\alpha-1)/\alpha}$;
Fact 2. $\mathsf{PWin}_{\mathcal{B}} = \mathsf{PWin}_{\mathcal{B}''}$;
Fact 3. $\mathsf{PWin}_{\mathcal{B}'} + \mathsf{PWin}_{\mathcal{B}''} \geqslant \mathsf{PWin}_{\mathcal{A}}$.

We can conclude the proof from the facts since

$$\begin{aligned}\mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) = \mathsf{PWin}_{\mathcal{A}} &\leqslant \mathsf{PWin}_{\mathcal{B}'} + \mathsf{PWin}_{\mathcal{B}''} \leqslant \delta_\alpha \mathsf{PWin}_{\mathcal{B}}^{(\alpha-1)/\alpha} + \mathsf{PWin}_{\mathcal{B}} \\ &\leqslant 2\delta_\alpha \mathsf{PWin}_{\mathcal{B}}^{(\alpha-1)/\alpha} = 2\delta_\alpha \mathsf{Adv}^{\text{td-msis}}_{q,k,m-1,\beta_{\mathsf{sis}}}(\mathcal{B}, \kappa)^{(\alpha-1)/\alpha} ,\end{aligned}$$

where the last inequality is due to the fact that $\delta_\alpha \geqslant 1$ and $(\alpha - 1)/\alpha < 1$.

We now show the above three facts. For Fact 1, since the only difference between $\mathcal{B}$ and $\mathcal{B}'$ is the distribution of $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$, by Lemma 4,

$$\mathsf{PWin}_{\mathcal{B}'} \leqslant \left(\mathsf{PWin}_{\mathcal{B}} \cdot R_\alpha\left(P \| \mathscr{D}_\Sigma^{mQ}\right)\right)^{(\alpha-1)/\alpha} .$$

Therefore, Fact 1 follows from the following lemma, which we will prove below.

Lemma 8. *For the discrete Gaussian distribution $\mathscr{D}_\Sigma^{mQ}$ (defined in Section 3.3), where $\Sigma = \mathbb{I}_m \otimes \mathrm{diag}(\sigma_1^2, \ldots, \sigma_Q^2)$, and the distribution $P = \Phi(\mathscr{D}_\Sigma^{mQ})$,*

$$R_\alpha\left(P \| \mathscr{D}_\Sigma^{mQ}\right) \leqslant \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \cdot \exp\left(\alpha\pi^2 / \log(2(1+1/\varepsilon))\right) .$$

The arguments for Facts 2 and 3 follow exactly as in Section 2.1. We repeat here for completeness. For Fact 2, since $\mathrm{View}_A(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) = \mathrm{View}_A(\Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)) = \mathrm{View}_A(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$, even if $\mathcal{B}''$ sets the secrets to $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$ instead of $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q$, the output of $\mathcal{B}''$ remain the same. However, then, $\mathcal{B}''$ is identical to $\mathcal{B}$, which implies the second fact.

Finally, to prove the third fact, we can interpret the sampling process of $\mathcal{B}'$ as first sampling $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$ from $\mathcal{P}$ and then setting $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \leftarrow \Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q)$. Then, the only difference between $\mathcal{B}'$ and $\mathcal{B}''$ is that $\mathcal{B}'$ check whether there exists $\boldsymbol{s}_i \neq \hat{\boldsymbol{s}}_i$, while $\mathcal{B}''$ check whether there exists

$\boldsymbol{r}_i \neq \hat{\boldsymbol{s}}_i$. If $\mathcal{A}$ wins the game $\mathbf{G}_2$ simulated by $\mathcal{B}'$ or $\mathcal{B}''$, we know $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) \neq \Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_Q) = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ by the definition of Φ, and thus, at least one of $\mathcal{B}'$ and $\mathcal{B}''$ wins, which implies Fact 3. □

Proof (of Lemma 8). We first partition the support R^{mQ} into disjoint sets, then show that the Rényi divergence conditioning on each set is small, and finally use Lemma 5 to conclude the lemma. Denote $\mathcal{S}_{\mathsf{L}}$ as the set of $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ such that the event AWin does not occur. Since Φ is the identity function over $\mathcal{S}_{\mathsf{L}}$, $\mathscr{D}_{\Sigma}^{mQ}(\mathcal{S}_{\mathsf{L}}) = P(\mathcal{S}_{\mathsf{L}})$. For each $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q) \notin \mathcal{S}_{\mathsf{L}}$, denote $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q] := \{(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta)\}_{k\in\mathbb{Z}}$, where $\boldsymbol{u}$ is output by $\mathcal{A}$ given $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. By a similar argument as above, we know any $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) \in \mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$ leads to the same view of $\mathcal{A}$. Therefore, $\Phi(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta) = (\boldsymbol{s}_1 + (k+1)u_1\Delta, \ldots, \boldsymbol{s}_Q + (k+1)u_Q\Delta)$, and Φ is a bijection over $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$, which implies $\mathscr{D}_{\Sigma}^{mQ}(\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]) = P(\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q])$. Also, for any $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q), (\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q) \notin \mathcal{S}_{\mathsf{L}}$, $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$ and $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_Q]$ are either equal to disjoint. Therefore, R^{mQ} can be partitioned into disjoint sets $\{\mathcal{S}_{\mathsf{L}}\} \cup \{\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]\}_{(\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q)\in R^{mQ}\setminus\mathcal{S}_{\mathsf{L}}}$, and for each set, the probability that $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$ falls in the set is equal under both distributions $\mathscr{D}_{\Sigma}^{mQ}$ and P.

Therefore, by Lemma 5, we just need to show the Rényi divergence conditioning on each set is small. For $\mathcal{S}_{\mathsf{L}}$, since Φ is the identity function over $\mathcal{S}_{\mathsf{L}}$, $R_\alpha\left(P_{|\mathcal{S}_{\mathsf{L}}} \| \mathscr{D}_{\Sigma}^{mQ}{}_{|\mathcal{S}_{\mathsf{L}}}\right) = 1$.

For any $\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q]$, we show in the following that the conditioned distribution is identical to a one-dimensional discrete Gaussian distribution under a linear transformation. Let $\sqrt{\Sigma}^{-1} = \mathbb{I}_m \otimes \mathrm{diag}(1/\sigma_1, \ldots, 1/\sigma_Q)$. Denote $\boldsymbol{X} := \sqrt{\Sigma}^{-1}(u_1\Delta, \ldots, u_Q\Delta)$ and $\boldsymbol{S} := \sqrt{\Sigma}^{-1}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q)$. Denote $\boldsymbol{S}_\perp := \boldsymbol{S} - s_0\boldsymbol{X}$, where $s_0 := \langle \boldsymbol{S}, \boldsymbol{X}\rangle / \langle \boldsymbol{X}, \boldsymbol{X}\rangle \in \mathbb{R}$, and we have $\langle \boldsymbol{S}_\perp, \boldsymbol{X}\rangle = \langle \boldsymbol{S}, \boldsymbol{X}\rangle - s_0 \langle \boldsymbol{X}, \boldsymbol{X}\rangle = 0$. Then, for any $k \in \mathbb{Z}$,

$$
\begin{aligned}
\rho_\sigma(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta) &= \exp\left(-\pi \left\|\sqrt{\Sigma}^{-1}(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta)\right\|^2\right) \\
&= \exp\left(-\pi \|\boldsymbol{S} + k\boldsymbol{X}\|^2\right) \\
&= \exp\left(-\pi \left(\|\boldsymbol{S}_\perp\|^2 + (s_0 + k)^2 \|\boldsymbol{X}\|^2\right)\right) \\
&\propto \exp\left(-\pi(-s_0 - k)^2 \|\boldsymbol{X}\|^2\right) .
\end{aligned}
$$

Therefore, $\mathscr{D}_{\Sigma}^{mQ}{}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]} = T\left(\mathscr{D}_{\|\boldsymbol{X}\|^{-1},-s_0}\right)$, where $T : \mathbb{Z} \to R^{mQ}$ maps k to $(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta)$. Since $\Phi(\boldsymbol{s}_1 + (k-1)u_1\Delta, \ldots, \boldsymbol{s}_Q + (k-1)u_Q\Delta) = (\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta)$, we know $P(\boldsymbol{s}_1 + ku_1\Delta, \ldots, \boldsymbol{s}_Q + ku_Q\Delta) = \mathscr{D}_{\Sigma}^{mQ}(\boldsymbol{s}_1 + (k-1)u_1\Delta, \ldots, \boldsymbol{s}_Q + (k-1)u_Q\Delta)$. By a similar argument as above, $P_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]} = T\left(\mathscr{D}_{\|\boldsymbol{X}\|^{-1},-s_0+1}\right)$. Since $\|\boldsymbol{X}\|^2 = \sum_{i\in[Q]}(u_i/\sigma_i)^2 \|\Delta\|^2 = \|(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)\|^2 \|\Delta\|^2 \leqslant \sqrt{mN}\beta_{\mathsf{u}}\beta_{\mathsf{lwe}}$, we have

$$
\|\boldsymbol{X}\|^{-1} \geqslant 1/(\sqrt{mN}\beta_{\mathsf{u}}\beta_{\mathsf{lwe}}) = \sqrt{\log(2(1 + 1/\varepsilon))/\pi} \geqslant \eta_\varepsilon(\mathbb{Z}) .
$$

Therefore, by Lemma 6,

$$\begin{aligned} R\left(\mathscr{D}_{\Sigma}^{mQ}{}_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]} \| P_{|\mathcal{S}_{\mathsf{W}}[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_Q]}\right) &\leqslant \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \cdot \exp\left(\alpha\pi \left\|\boldsymbol{X}\right\|^2\right) \\ &\leqslant \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \cdot \exp\left(\alpha\pi m N \beta_{\mathsf{u}}^2 \beta_{\mathsf{lwe}}^2\right) \\ &= \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}} \cdot \exp\left(\alpha\pi^2/\log(2(1+1/\varepsilon))\right) \ . \end{aligned}$$

Therefore, we can conclude the lemma by Lemma 5. □

5 Analysis of the CTZ Construction

As our first application to threshold signatures, this section applies AOM-MISIS to the analysis of the CTZ construction [CATZ24].

5.1 Construction and main security theorem

We present the scheme $\mathsf{CTZ}[\mathsf{SecSha}]$ in Figure 6, where $\mathsf{SecSha} = \mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ is a linear threshold secret sharing scheme with small coefficients, which is defined in Appendix A. In Figure 7, we give the description of the parameters used in the protocol. One small change here is that the signing key is sampled from a discrete Gaussian distribution with standard deviation σ_{sk} instead of a uniform distribution over the set of vectors with ℓ_∞-norm bounded by σ_{sk}. We note that this does not affect the correctness of the scheme, as the ℓ_2-norm of $\widehat{\mathsf{sk}}$ remains bounded by $\sqrt{mN}\sigma_{\mathsf{sk}}$ except for a negligible probability, which is the exact property needed in the correctness proof.

For unforgeability, we show that TS-UF-0 security of CTZ is implied by the hardness of the AOM-MISIS problem in the random oracle model, which is formally stated in the following theorem. In particular, we consider an extension of AOM-MISIS (See Remark 1 for details) in which some σ_i can be ∞. The full proof of Theorem 2 is given in Section 5.3.

Theorem 2 (TS-UF-0 **of** CTZ)**.** *For any integers $q = q(\kappa), k = k(\kappa), m = m(\kappa)$, any linear threshold secret sharing scheme $\mathsf{SecSha} = \mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ with small coefficients, (see Definition 2) and any* TS-UF-0 *adversary $\mathcal{A}$ making at most $\mathsf{q}_s = \mathsf{q}_s(\kappa)$ queries to* PPO *and $\mathsf{q}_h = \mathsf{q}_h(\kappa)$ queries to* RO*, there exists an* AOM-MISIS *adversary $\mathcal{B}$ running in time roughly two times that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\text{ts-uf-0}}_{\mathsf{CTZ}}(\mathcal{A},\kappa) \leqslant \sqrt{\mathsf{q}\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{B},\kappa) + 8\mathsf{q}^3 2^{-2\kappa}} \ .$$

where $\mathsf{q} = \mathsf{q}_h + \mathsf{q}_s + 1$ and $\mathsf{par} = (q, k, m, Q = 1 + K + \mathsf{q}_s(1+\ell), (\sigma_i)_{i\in[Q]}, \beta_{\mathsf{s}} = 2\beta_{\mathsf{z}}, \beta_{\mathsf{b}} = 4\sigma_{\mathsf{sk}}\beta_{\mathsf{c}}, \beta_{\mathsf{u}} = 1/\sigma_{\mathsf{sk}} + \beta_{\mathsf{c}} B_{\mathsf{ss}}\sqrt{8N\mathsf{q}_s}/\sigma_{\mathsf{r}})$ with $\sigma_1 = \sigma_{\mathsf{sk}}$, $\sigma_{1+i} = \infty$ for $i \in [K]$, $\sigma_{1+K+i} = \sigma_{\mathsf{r}}$ for $i \in [\mathsf{q}_s(\ell+1)]$.

For completeness, we recall the correctness theorem from [CATZ24], which is needed for parameter selections later.

Theorem 3 (Correctness of CTZ **[CATZ24]).** *For any integers $1 < t \leqslant n$, any linear threshold secret sharing scheme $\mathsf{SecSha} = \mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ with small coefficients, (see Definition 2) given $\sigma_{\mathsf{r}} \geqslant 2\sqrt{6mN\log(2mN)\kappa/\pi}$ and $\beta_{\mathsf{z}} \geqslant \sqrt{mN}(2\beta_{\mathsf{c}}\sigma_{\mathsf{sk}} + \sigma_{\mathsf{r}}\sqrt{n(1+\ell)})$, the threshold signature scheme $\mathsf{CTZ}[\mathsf{SecSha}]$ is correct with correctness error $\varepsilon_{\mathsf{cor}} \leqslant (2 + 4n(\ell+1)) \cdot 2^{-2\kappa}$.*

$\underline{\mathsf{Setup}(1^\kappa):}$

$\overline{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\overline{A}|\mathbb{I}_k]$
$par \leftarrow A$
For $i \in [n]$ do
  $\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \varnothing$
  $\mathsf{st}_i.\mathrm{mapPP} \leftarrow ()$
Return par

$\underline{\mathsf{KeyGen}():}$

$\widehat{\mathsf{sk}} \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_{\mathsf{sk}}}$; $\mathsf{pk} \leftarrow A\widehat{\mathsf{sk}} \bmod q$
$\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_K \leftarrow\!\!\$\ R_q^m$
$(\mathsf{ss}_1, ..., \mathsf{ss}_L)^T \leftarrow$
  $M \cdot (\widehat{\mathsf{sk}}, \boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_K)^T$
For $i \in [n]$ do $\mathsf{sk}_i \leftarrow (\mathsf{ss}_j)_{j\in T_i}$
Return $(\mathsf{pk}, (\mathsf{sk}_i)_{i\in[n]})$

$\underline{\mathsf{SPP}(\mathsf{st}_i):}$

For $j \in [0..\ell]$ do $\boldsymbol{r}_j \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_r}$
For $j \in [0..\ell]$ do $\boldsymbol{R}_j \leftarrow A\boldsymbol{r}_j \bmod q$
$pp \leftarrow (\boldsymbol{R}_j)_{j\in[0..\ell]}$
$\mathsf{st}_i.\mathrm{mapPP}(pp) \leftarrow (\boldsymbol{r}_j)_{j\in[0..\ell]}$
Return (pp, st_i)

$\underline{\mathsf{LPP}(i, pp, \mathsf{st}_0):}$

$\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \mathsf{st}_0.\mathrm{curPP}_i \cup \{pp\}$
Return st_0

$\underline{\mathsf{LR}(\mu, SS, \mathsf{st}_0):}$

If $\exists\, i \in SS : \mathsf{st}_0.\mathrm{curPP}_i = \varnothing$ then
  Return $\perp$
$lr.\mathsf{msg} \leftarrow \mu$; $lr.\mathsf{SS} \leftarrow SS$
For $i \in SS$ do
  Pick pp_i from $\mathsf{st}_0.\mathrm{curPP}_i$
  $lr.\mathsf{PP}(i) \leftarrow pp_i$
  $\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \mathsf{st}_0.\mathrm{curPP}_i \backslash \{pp_i\}$
Return (lr, st_0)

$\underline{\mathsf{CompPar}(\mathsf{pk}, lr):}$

$\mu \leftarrow lr.\mathsf{msg}$
For $i \in lr.\mathsf{SS}$ do
  $(b_j)_{j\in[\ell]} \leftarrow \mathrm{H}_1(\mathsf{pk}, lr)$
  $(\boldsymbol{R}_{i,j})_{j\in[0..\ell]} \leftarrow lr.\mathsf{PP}(i)$
$\boldsymbol{R} \leftarrow \sum_{i\in lr.\mathsf{SS}} \left(\boldsymbol{R}_{i,0} + \sum_{j\in[\ell]} b_j \boldsymbol{R}_{i,j}\right)$
$c \leftarrow \mathrm{H}_2(\mathsf{pk}, \mu, \boldsymbol{R})$
Return $(\boldsymbol{R}, c, (b_j)_{j\in[\ell]})$

$\underline{\mathsf{PS}(lr, i, \mathsf{st}_i):}$

$pp_i \leftarrow lr.\mathsf{PP}(i)$
If $\mathsf{st}_i.\mathrm{mapPP}(pp_i) = \perp$ then return $(\perp, \mathsf{st}_i)$
$(\boldsymbol{r}_j)_{j\in[0..\ell]} \leftarrow \mathsf{st}_i.\mathrm{mapPP}(pp_i)$
$\mathsf{st}_i.\mathrm{mapPP}(pp_i) \leftarrow \perp$
$(\boldsymbol{R}, c, (b_j)_{j\in[\ell]}) \leftarrow \mathsf{CompPar}(\mathsf{st}_i.\mathsf{pk}, lr)$
$(\mathsf{ss}_j)_{j\in T_i} \leftarrow \mathsf{st}_i.\mathsf{sk}$
$\boldsymbol{z} \leftarrow \boldsymbol{r}_0 + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}_j$
  $+2c \cdot \sum_{j\in T_i} \lambda_j^{lr.\mathsf{SS}} \mathsf{ss}_j \bmod q$
Return $((\boldsymbol{R}, \boldsymbol{z}), \mathsf{st}_i)$

$\underline{\mathsf{Agg}(\mathrm{PS}, \mathsf{st}_0):}$

$\boldsymbol{R} \leftarrow \perp$; $\boldsymbol{z} \leftarrow 0$
For $(\boldsymbol{R}', \boldsymbol{z}') \in \mathrm{PS}$ do
  If $\boldsymbol{R} = \perp$ then $\boldsymbol{R} \leftarrow \boldsymbol{R}'$
  If $\boldsymbol{R} \neq \boldsymbol{R}'$ then return $(\perp, \mathsf{st}_0)$
  $\boldsymbol{z} \leftarrow \boldsymbol{z} + \boldsymbol{z}'$
Return $((\boldsymbol{R}, \boldsymbol{z}), \mathsf{st}_0)$

$\underline{\mathsf{Vf}(\mathsf{pk}, \mu, sig):}$

$(\boldsymbol{R}, \boldsymbol{z}) \leftarrow sig$
If $\|\boldsymbol{z}\| > \beta_{\mathsf{z}}$ then return 0
$c \leftarrow \mathrm{H}_2(\mathsf{pk}, \mu, \boldsymbol{R})$
Return $(A\boldsymbol{z} = \boldsymbol{R} + 2c \cdot \mathsf{pk} \bmod q)$

Fig. 6. Lattice-based t-out-of-n threshold signatures $\mathsf{CTZ}[\mathsf{SecSha}]$, where $\mathsf{SecSha} = \mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ is t-out-of-n a linear secret sharing scheme with small coefficients (see Definition 2). In particular, K denotes the randomness size of SecSha, L denotes the total share size, M denotes the sharing matrix, T_i denotes the set of shares of party i, and $\lambda_j^{lr.\mathsf{SS}}$ denotes the reconstruction coefficient. Also, $\mathrm{H}_1 : \{0,1\}^* \to \mathcal{S}_{\mathsf{b}}^{\ell}$ and $\mathrm{H}_2 : \{0,1\}^* \to \mathcal{S}_{\mathsf{c}}$. We remark that, as stated earlier, the public parameter par is implicitly given to all algorithms except Setup.

5.2 Parameter selection

In this section, we first discuss the asymptotic parameters selection derived from the security theorems and the hardness of AOM-MISIS, then compare these parameters with those proposed in [CATZ24], and finally estimate the concrete efficiency based on the parameter selections.

Asymptotic parameter selections. Denote β_{lwe} as the norm of the underlying MLWE assumption. Initially, we select $N, m, k, \beta_{\mathsf{lwe}}$ such that N is a power of $N \geqslant 2\kappa$, $m, k = \mathsf{poly}(\kappa)$, and $\beta_{\mathsf{lwe}} \geqslant m\log(N)$.[5] (We note that when estimating the concrete efficiency, we will enumerate

[5] This is for guaranteeing the underlying MLWE is hard.

Parameters	Description
κ	Security parameter
n	Number of signers
t	Threshold for signing
N	$N \geqslant 2\kappa$, power of two defining the ring R
q	Prime modulus
k	Number of rows of A
m	$m > k$, number of columns of A,
$\ell + 1$	$\ell = 2\kappa/\log(2N)$, number of nonces for each signer
$\mathcal{S}_{\mathsf{b}}$	$\mathcal{S}_{\mathsf{b}} = \{\pm 1, \pm X, \ldots, \pm X^{N-1}\}$, set for the aggregating coefficients b_j
β_{c}	Satisfying $2^{\beta_{\mathsf{c}}}\binom{N}{\beta_{\mathsf{c}}} \geqslant 2^{2\kappa}$, the ℓ_1-norm of the challenge c,
$\mathcal{S}_{\mathsf{c}}$	$\mathcal{S}_{\mathsf{c}} = \{c \in R : \|c\|_\infty = 1, \|c\|_1 = \beta_{\mathsf{c}}\}$, set of the challenges c
σ_{sk}	Standard deviation of the signing key sk
σ_{r}	Standard deviation of the nonces $\boldsymbol{r}_i$
ν_{pk}	Only for EKT: satisfying $\lfloor q/2^{\nu_{\mathsf{pk}}} \rceil = \lfloor q/2^{\nu_{\mathsf{pk}}} \rceil$, number of bits saved on pk
ν_{r}	Only for EKT: satisfying $\lfloor q/2^{\nu_{\mathsf{r}}} \rceil = \lfloor q/2^{\nu_{\mathsf{r}}} \rceil$, number of bits saved on $\boldsymbol{h}$
$q_{\nu_{\mathsf{pk}}}, q_{\nu_{\mathsf{r}}}$	Only for EKT: $(q_{\nu_{\mathsf{pk}}}, q_{\nu_{\mathsf{r}}}) = (\lfloor q/2^{\nu_{\mathsf{pk}}} \rceil, \lfloor q/2^{\nu_{\mathsf{r}}} \rceil)$ the rounded moduli
β_{z}	ℓ_2-norm bound of a valid signature vector $\boldsymbol{z}$
	(or $(\boldsymbol{z}, 2^{\nu_{\mathsf{pk}}}\boldsymbol{h})$ for EKT)

Fig. 7. Parameters for CTZ and EKT. Some parameters only apply to EKT.

through plausible $(N, m, k, \beta_{\mathsf{lwe}})$ tuples and pick the one that yields the best efficiency.) Then, we set other parameters as follows.

- Set β_{c} as the smallest integer such that $2^{\beta_{\mathsf{c}}}\binom{N}{\beta_{\mathsf{c}}} \geqslant 2^{2\kappa}$.
- $\sigma_{\mathsf{sk}} = \max\{2\beta_{\mathsf{lwe}}\sqrt{mN}, \sqrt{\log(6mN)/\pi}\}$. The first term is usually the leading term.
- $\sigma_{\mathsf{r}} = \max\{\sigma_{\mathsf{sk}}\beta_{\mathsf{c}}B_{\mathsf{ss}}\sqrt{8N\mathsf{q}_s}, 2\sqrt{6mN\log(2mN)\kappa/\pi}\}$. The first term is usually the leading term.
- $\beta_{\mathsf{z}} = \sqrt{mN}(2\beta_{\mathsf{c}}\sigma_{\mathsf{sk}} + \sigma_{\mathsf{r}}\sqrt{n(1+\ell)})$
- Denote $\beta_{\mathsf{sis}} = 2\beta_{\mathsf{z}} + 4\sigma_{\mathsf{sk}}\beta_{\mathsf{c}}\sqrt{mN}$.
- Select q such that the problem $\mathrm{MSIS}_{q,k,m,\beta_{\mathsf{sis}}}$ and the problem $\mathrm{MLWE}_{q,N,k,m,\beta_{\mathsf{lwe}}}$ are assumed to be exponentially hard in κ.

By Theorem 2 and Theorem 1 with $\varepsilon = 1/2$ and $\alpha = 2$ (we can further optimize the concrete bound by adjusting α), TS-UF-0 of CTZ is implied by the hardness of $\mathrm{MSIS}_{q,k,m,\beta_{\mathsf{sis}}}$ and $\mathrm{MLWE}_{q,N,k,m,\beta_{\mathsf{lwe}}}$.

Compared with [CATZ24] The differences of the parameter selections between [CATZ24] and ours shows in the selections of σ_{sk} and σ_{r}. In particular, the prior work requires

- $\sigma_{\mathsf{sk}} = q^{k/m}2^{2\kappa/(Nm)}$.
- $\sigma_{\mathsf{r}} = \max\{N\beta_{\mathsf{c}}B_{\mathsf{ss}}\sigma_{\mathsf{sk}}\sqrt{32\pi\mathsf{q}_s mN}, \frac{16N\sqrt{3m}}{\sqrt{\pi}}q^{\frac{k}{m}}\sqrt{N(\log(2mN) + 2\kappa)}\}$. The first term is usually the leading term.

For σ_{sk}, $q^{k/m}$ is significantly larger than $2\beta_{\mathsf{lwe}}\sqrt{mN}$, which influences the choice of k and m. In particular, one needs to set m to be several times larger than k to ensure σ_{sk} does not cause the parameters to grow excessively. For σ_{r}, we can see a factor of $N\sqrt{m}$ improvement by comparing the first term inside the maximization.

Concrete efficiency. We show a set of concrete parameters and estimated efficiency for $\kappa = 128$ and $n = 32$, and compare them with those from [CATZ24] in Figure 8, where we can see improvements in both signature sizes and communication complexity. We derive the parameters following

	$\log_2(q)$	k	m	σ_{sk}	σ_{r}	β_{z}	$\lvert\mathsf{pk}\rvert$	$\lvert sig\rvert$	Comm.
[CATZ24]	119	8	50	2^{18}	$2^{105.06}$	$2^{117.25}$	60.73KB	440.32KB	2.02MB
This work	94.1	7	17	$2^{9.1}$	$2^{80.5}$	$2^{91.96}$	42.14KB	144.47KB	1.24MB

Fig. 8. The concrete parameters and estimated efficiency for $\kappa = 128$ and $n = 32$ in [CATZ24] and this work. We set $(N, \ell, \beta_{\mathsf{c}}) = (512, 26, 64)$. The last second column denotes the communication complexity per signer.

our parameter selections mentioned above, where SecSha are instantiated using Lemma 14. To estimate the concrete hardness of MSIS and MLWE, we use the state-of-art tool, lattice estimator,[6] which is also used in other prior works [DKM+24, EKT24, BKL+24].

5.3 Security reduction of CTZ

The proof structure generally follows from the security proof of FROST [BTZ22], which reduces the security of FROST to the algebraic one-more discrete logarithm assumption, as the CTZ construction can be seen as a lattice analog of FROST. The key differences lie in how the reduction computes $\boldsymbol{u}$ and how we bound the norm of the output solution $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{b}})$ and $\boldsymbol{u}$.

To prove Theorem 2, we borrow the following variant of Forking Lemma from [CATZ24]. The differences from the generalized Forking Lemma [BN06] are that here each h_i might be sampled independently from a different distribution and that if the index output by $\mathcal{A}$ is guaranteed to lie within a subset $S \subseteq [q]$, then the final bound on the success probability of $\mathsf{Fork}^{\mathcal{A}}$ depends on $|S|$ instead of q (as observed in [BTZ22]). The former is needed in our proof since the ranges of H_1 and H_2 differ, whereas the latter provides tighter bounds in our security analysis.

Lemma 9. *Let $q \geqslant 1$ be an integer, $S \subseteq [q]$ be a set, and HG be an algorithm that outputs $h_1, \ldots, h_q$ where each h_i is independently sampled. Let $\mathcal{A}$ be a randomized algorithm that on input $x, h_1, \ldots, h_q$ outputs a pair (I, Out), where $I \in \{\perp\} \cup S$ and Out is a side output. Let IG be a randomized algorithm that generates x. The accepting probability of $\mathcal{A}$ is defined as*

$$\mathrm{acc}(\mathcal{A}) = \mathsf{Pr}_{x \leftarrow\$ \mathsf{IG}, h_1, \ldots, h_q \leftarrow\$ \mathsf{HG}}[(I, \mathrm{Out}) \leftarrow\$ \mathcal{A}(x, h_1, \ldots, h_q) \ : \ I \neq \perp] \ .$$

Consider algorithm $\mathsf{Fork}^{\mathcal{A}}$ described in Figure 9. The accepting probability of $\mathsf{Fork}^{\mathcal{A}}$ is defined as

$$\mathrm{acc}(\mathsf{Fork}^{\mathcal{A}}) = \mathsf{Pr}_{x \leftarrow\$ \mathsf{IG}}[\alpha \leftarrow\$ \mathsf{Fork}^{\mathcal{A}}(x) \ : \ \alpha \neq \perp] \ .$$

Then, $\mathrm{acc}(\mathsf{Fork}^{\mathcal{A}}) \geqslant \mathrm{acc}(\mathcal{A})^2 / |S|$.

Proof (of Theorem 2). Let $\mathcal{A}$ be a TS-UF-0 adversary as described in the theorem. W.l.o.g. we assume that $\mathcal{A}$ is deterministic and corrupts exactly $t - 1$ signers. Also, we assume if $\mathcal{A}$ returns $(\mu^*, (\boldsymbol{R}^*, \boldsymbol{z}^*))$, the RO query $\mathrm{H}_2(\mathsf{pk}, \mu^*, \boldsymbol{R}^*)$ was made by $\mathcal{A}$, which adds at most one RO query. Also, since the game makes at most one RO query to H_1 and H_2 respectively for each signing query, the total number of RO queries to each of H_1 and H_2 is bounded $\mathsf{q} = \mathsf{q}_h + \mathsf{q}_s + 1$. We first construct an algorithm $\mathcal{C}$ compatible with the syntax in Lemma 9 and construct $\mathcal{B}$ from $\mathsf{Fork}^{\mathcal{C}}$.

[6] https://github.com/malb/lattice-estimator

$\underline{\mathsf{Fork}^{\mathcal{A}}(x)}$:
Pick the random coin ρ of $\mathcal{A}$ at random
$(h_1, \ldots, h_q), (\bar{h}_1, \ldots \bar{h}_q) \leftarrow\!\!\$\ \mathsf{HG}$
$(I, \mathrm{Out}) \leftarrow \mathcal{A}(x, h_1, \ldots, h_q; \rho)$
If $I = \perp$ then return $\perp$
$(\bar{I}, \overline{\mathrm{Out}}) \leftarrow \mathcal{A}(x, h_1, \ldots, h_{I-1}, \bar{h}_I, \ldots, \bar{h}_q; \rho)$
If $I \neq \bar{I}$ then return $\perp$
Return $(I, \mathrm{Out}, \overline{\mathrm{Out}})$

Fig. 9. The forking algorithm build from $\mathcal{A}$.

CONSTRUCTION OF $\mathcal{C}$. The input of $\mathcal{C}$ consists of $A, (\boldsymbol{t}_i)_{i\in[K+1+\mathsf{q}_s(\ell+1)]}$, and a list of hash values $h_1, \ldots, h_{2\mathsf{q}}$, where $(A, (\boldsymbol{t}_i)_{i\in[K+1+\mathsf{q}_s(\ell+1)]})$ are sampled following the AOM-MISIS game, and for each $i \in [\mathsf{q}]$, h_{2i-1} is sampled uniformly from $\mathcal{S}_\mathsf{b}$ and h_{2i} is sampled uniformly from $\mathcal{S}_\mathsf{c}$. To start with, $\mathcal{C}$ sets $par \leftarrow A$, initializes $\mathsf{st}_i.\mathrm{mapPP} \leftarrow ()$ for $i \in [n]$, and in addition, initializes a counter $\mathrm{ctr}_h \leftarrow 0$ for counting the number of random oracle queries. Then, $\mathcal{C}$ runs $\mathcal{A}$ on input par with access to oracles $\widetilde{\mathrm{INIT}}$, $\widetilde{\mathrm{PPO}}$, $\widetilde{\mathrm{PSIGNO}}$ and $\widetilde{\mathrm{RO}}$, which are simulated as follows.

$\widetilde{\mathbf{Init}}(\mathsf{CS})$: $\mathcal{C}$ sets $\mathsf{pk} \leftarrow \boldsymbol{t}_1$ and views $\boldsymbol{t}_{1+\hat{j}}$ as $A\boldsymbol{\rho}_{\hat{j}}$ for $\hat{j} \in [K]$. Then, for each $i \in \mathsf{CS}$ and $j \in T_i$, $\mathcal{C}$ computes $\mathsf{ss}_j \leftarrow \mathrm{PI}(\boldsymbol{d})$ with

$$d_{\hat{j}} = \begin{cases} M_{j,\hat{j}}\ , & \hat{j} \in [K+1]\ , \\ 0\ , & o.w. \end{cases} \tag{6}$$

Finally, $\mathcal{C}$ returns $(\mathsf{pk}, (\mathsf{sk}_i = (\mathsf{ss}_j)_{j\in T_i})_{i\in\mathsf{CS}})$.

$\widetilde{\mathbf{PPO}}(i)$: For the j-th query, $\mathcal{C}$ sets $\boldsymbol{R}_{\hat{j}} \leftarrow \boldsymbol{t}_{K+1+(j-1)(\ell+1)+\hat{j}+1}$ for $\hat{j} \in [0..\ell]$. Since $\mathcal{C}$ does not sample $\{\boldsymbol{r}_{\hat{j}}\}_{\hat{j}\in[0..\ell]}$, $\mathcal{C}$ uses $\mathsf{st}_i.\mathrm{mapPP}$ to store the index j instead, i.e., $\mathcal{C}$ sets $\mathsf{st}_i.\mathrm{mapPP}(\{\boldsymbol{R}_{\hat{j}}\}_{\hat{j}\in[0..\ell]}) \leftarrow j$.

$\widetilde{\mathbf{PSignO}}(i, lr)$: The same as $\mathrm{PSIGNO}(i, lr)$ except that $\mathcal{C}$ computes $\boldsymbol{z}$ using the PI oracle as follows. Let $j \leftarrow \mathsf{st}_i.\mathrm{mapPP}(lr.\mathsf{PP}(i))$ and $\boldsymbol{d}$ be a vector in R^Q such that

$$d_{\hat{j}} = \begin{cases} 2c \cdot \sum_{\hat{i}\in T_i} \lambda_{\hat{i}}^{lr.\mathsf{SS}} M_{\hat{i},\hat{j}}\ , & \hat{j} \in [K+1]\ , \\ 1\ , & \hat{j} = K+1+(j-1)(\ell+1)+1\ , \\ b_{j'}\ , & \hat{j} = K+1+(j-1)(\ell+1)+1+j', j' \in [\ell]\ , \\ 0\ , & o.w. \end{cases} \tag{7}$$

$\mathcal{C}$ computes $\boldsymbol{z} \leftarrow \mathrm{PI}(\boldsymbol{d})$.

$\widetilde{\mathbf{RO}}$ query $\mathrm{H}_1(x)$: If $\mathrm{H}_1(x) \neq \perp$, $\mathcal{C}$ returns $\mathrm{H}_1(x)$. Otherwise, parse x as $(\widetilde{\mathsf{pk}}, lr)$. If the parsing fails or $\widetilde{\mathsf{pk}} \neq \mathsf{pk}$, $\mathcal{C}$ sets $\mathrm{H}_1(x) \leftarrow\!\!\$\ \mathcal{S}_\mathsf{b}^\ell$ and returns $\mathrm{H}_1(x)$. Otherwise, $\mathcal{C}$ increases ctr_h by 1, sets $\mathrm{H}_1(x) \leftarrow h_{2\mathrm{ctr}_h-1}$. Also, $\mathcal{C}$ computes $\boldsymbol{R} \leftarrow \sum_{i\in lr.\mathsf{SS}}(\boldsymbol{R}_{i,0} + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{R}_{i,j})$, where $(\boldsymbol{R}_{i,j})_{j\in[0..\ell]} \leftarrow lr.\mathsf{PP}(i)$ and $\{b_j\}_{j\in[\ell]} \leftarrow h_{2\mathrm{ctr}_h-1}$. If $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R}) = \perp$, $\mathcal{C}$ sets $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R}) \leftarrow h_{2\mathrm{ctr}_h}$. Finally, $\mathcal{C}$ returns $\mathrm{H}_1(x)$.

$\widetilde{\mathbf{RO}}$ query $\mathrm{H}_2(x)$: If $\mathrm{H}_2(x) \neq \perp$, $\mathcal{C}$ returns $\mathrm{H}_2(x)$. Otherwise, parse x as $(\widetilde{\mathsf{pk}}, \mu, \boldsymbol{R})$. If the parsing fails or $\widetilde{\mathsf{pk}} \neq \mathsf{pk}$, $\mathcal{C}$ sets $\mathrm{H}_2(x) \leftarrow\!\!\$\ \mathcal{S}_\mathsf{c}$. Otherwise, $\mathcal{C}$ increases ctr_h by 1 and sets $\mathrm{H}_2(x) \leftarrow h_{2\mathrm{ctr}_h}$. Finally, $\mathcal{C}$ returns $\mathrm{H}_2(x)$.

After receiving the output $(\mu^*, (\boldsymbol{R}^*, \boldsymbol{z}^*))$ from $\mathcal{A}$, $\mathcal{C}$ aborts if $\mathcal{A}$ does not win the TS-UF-0 game. Otherwise, $\mathcal{C}$ finds the index I such that $\mathrm{H}_2(\mathsf{pk}, \mu^*, \boldsymbol{R}^*)$ is set to h_I during the simulation. By our assumption of $\mathcal{A}$, we know such I must exist. Then, $\mathcal{C}$ returns $(I, \mathrm{Out} = (\mu^*, \boldsymbol{R}^*, \boldsymbol{z}^*))$.

Analysis of $\mathcal{C}$. To use Lemma 9, we define $S := \{2i\}_{i\in[\mathsf{q}]}$, IG as the algorithm that samples $(A, (\boldsymbol{t}_i)_{i\in[K+1+\mathsf{q}_s(\ell+1)]})$ following the AOM-MISIS game, and HG as the algorithm that samples h_{2i-1} uniformly from $\mathcal{S}_\mathsf{b}$ and samples h_{2i} uniformly from $\mathcal{S}_\mathsf{c}$ for each $i \in [\mathsf{q}]$. From the simulation, we know that the output index I of $\mathcal{C}$ is always in S. Also, it is not hard to check that $\mathcal{C}$ simulates the game TS-SUF-0 perfectly, which implies $\mathrm{acc}(\mathcal{C}) \geqslant \mathsf{Adv}^{\text{ts-uf-0}}_{\mathsf{CTZ}}(\mathcal{A}, \kappa)$. By Lemma 9, we have that

$$\mathrm{acc}(\mathsf{Fork}^{\mathcal{C}}) \geqslant \mathsf{Adv}^{\text{ts-uf-0}}_{\mathsf{CTZ}}(\mathcal{A}, \kappa)^2/\mathsf{q} \,.$$

Construct $\mathcal{B}$ from $\mathsf{Fork}^{\mathcal{C}}$. We now construct the AOM-MISIS adversary $\mathcal{B}$ using $\mathsf{Fork}^{\mathcal{C}}$. To start with, $\mathcal{B}$ receives $(A, \{\boldsymbol{t}_i\}_{i\in[Q]})$ from the AOM-MISIS game with $Q = K + 1 + \mathsf{q}_s(\ell + 1)$ and runs $\mathsf{Fork}^{\mathcal{C}}(A, \{\boldsymbol{t}_i\}_{i\in[Q]})$ with access to the PI oracle from the AOM-MISIS game. If $\mathsf{Fork}^{\mathcal{C}}$ outputs $(I, \mathrm{Out} = (\mu^*, \boldsymbol{R}^*, \boldsymbol{z}^*), \overline{\mathrm{Out}} = (\bar{\mu}^*, \bar{\boldsymbol{R}}^*, \bar{\boldsymbol{z}}^*))$, we know $A\boldsymbol{z}^* = \boldsymbol{R}^* + 2h_I\mathsf{pk} \bmod q$ and $A\bar{\boldsymbol{z}}^* = \bar{\boldsymbol{R}}^* + 2\bar{h}_I\mathsf{pk} \bmod q$, which implies $A(\boldsymbol{z}^* - \bar{\boldsymbol{z}}^*) = 2(h_I - \bar{h}_I)\mathsf{pk} \bmod q$. Therefore, $\mathcal{B}$ sets $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{z}^* - \bar{\boldsymbol{z}}^*$ and $\hat{\boldsymbol{b}} \leftarrow (2(h_I - \bar{h}_I), 0, \ldots, 0)$, and it holds that $\sum_{i\in[Q]} \hat{b}_i\boldsymbol{t}_i = 2(h_I - \bar{h}_I)\boldsymbol{t}_0 = 2(h_I - \bar{h}_I)\mathsf{pk} = A(\boldsymbol{z}^* - \bar{\boldsymbol{z}}^*) = \hat{\boldsymbol{s}} \bmod q$.

We now show how $\mathcal{B}$ sets $\boldsymbol{u}$ such that $\hat{\boldsymbol{b}}^T\boldsymbol{u} \neq 0$ and $\boldsymbol{d}^T\boldsymbol{u} = 0$ for any oracle query $\mathrm{PI}(\boldsymbol{d})$.

By the linearity of SecSha, there exists a sweeping vector $\boldsymbol{w} \in \mathbb{Z}^{K+1}$ such that $M_{\mathsf{CS}}\boldsymbol{w} = \boldsymbol{0}$ and $w_1 = 1$, and we set $u_{[K+1]} = \boldsymbol{w}$. Therefore, $\hat{\boldsymbol{b}}^T\boldsymbol{u} = \hat{b}_1u_1 = \hat{b}_1 \neq 0$. Also, for each PI query made during the execution of $\widetilde{\textsc{Init}}$, the query is of the form $\boldsymbol{d} = (M_{j,1}, \ldots, M_{j,K+1}, 0, \ldots, 0)$, where $j \in \bigcup_{i\in\mathsf{CS}} T_i$, and thus $\boldsymbol{d}^T\boldsymbol{u} = (M_{j,1}, \ldots, M_{j,K+1}) \cdot \boldsymbol{w} = 0$.

For $j \in [\mathsf{q}_s]$, $\mathcal{B}$ sets $u_{K+1+(j-1)(\ell+1)+[\ell+1]}$ as follows. To simplify notation in the following analysis, we use $\boldsymbol{v}$ to denote the vector $u_{K+1+(j-1)(\ell+1)+[\ell+1]} \in R^{\ell+1}$. We say a $\widetilde{\mathrm{PSign}}\mathrm{O}$ query (i, lr) corresponds to the j-th token if and only if it is the valid query with $\mathsf{st}_i.\mathrm{mapPP}(lr.\mathsf{PP}(i)) = j$, where a valid query means the $\widetilde{\mathrm{PSign}}\mathrm{O}$ oracle does not return $\perp$. From the simulation, there is at most one $\widetilde{\mathrm{PSign}}\mathrm{O}$ query corresponding to the j-th token during each execution of $\mathcal{A}$. Therefore, there are the following cases:

Case 1: No query corresponds to the j-th token during both executions. In this case, $\mathcal{B}$ set $\boldsymbol{v} \leftarrow \boldsymbol{0}$.

Case 2: Only one query corresponds to the j-th token during the two executions. Denote $\boldsymbol{d}$ as the PI query made during the execution of the $\widetilde{\mathrm{PSign}}\mathrm{O}$ query corresponding to the j-th token, where $\boldsymbol{d}$ follows the form given in Equation (7). $\mathcal{B}$ sets $\boldsymbol{v} \leftarrow (-\sum_{\hat{j}\in[K+1]} d_{\hat{j}}u_{\hat{j}}, 0, \ldots, 0)$, which implies $\boldsymbol{d}^T\boldsymbol{u} = \sum_{\hat{j}\in[K+1]} d_{\hat{j}}u_{\hat{j}} + v_1 = 0$.

Case 3: There is one query corresponding to the j-th token during each of the two executions. Denote $\boldsymbol{d}$ (resp. $\bar{\boldsymbol{d}}$) as the PI query made during the execution of the $\widetilde{\mathrm{PSign}}\mathrm{O}$ query corresponding to the j-th token before (resp. after) rewinding. If $\boldsymbol{d} = \bar{\boldsymbol{d}}$, then $\mathcal{C}$ sets $\boldsymbol{v}$ in the same way as Case 2. Otherwise, let $\hat{k} \in [\ell]$ be the index such that $d_{K+1+(j-1)(\ell+1)+1+\hat{k}} \neq \bar{d}_{K+1+(j-1)(\ell+1)+1+\hat{k}}$. (If such $\hat{k}$ does not exist, $\mathcal{B}$ aborts.) Denote $b := d_{K+1+(j-1)(\ell+1)+1+\hat{k}}$ and $\bar{b} := d_{K+1+(j-1)(\ell+1)+1+\hat{k}}$. By Equation (7), we know that 2 divides $d_{\hat{j}}$ and $\bar{d}_{\hat{j}}$ for $\hat{j} \in [K+1]$. Therefore, denote $\Delta := \sum_{\hat{j}\in[K+1]}(d_{\hat{j}}/2)u_{\hat{j}}$ and $\bar{\Delta} := \sum_{\hat{j}\in[K+1]}(\bar{d}_{\hat{j}}/2)u_{\hat{j}}$. Since $b, \bar{b} \in \mathcal{S}_\mathsf{b}$, by Lemma 1, there exists $\gamma \in R$

such that $\gamma(b-\bar{b}) = 2 \mod q$. $\mathcal{C}$ sets

$$v_{\hat{j}} \leftarrow \begin{cases} -2\Delta + b\gamma(\Delta - \bar{\Delta}) \,, & \hat{j} = 1 \,, \\ -\gamma(\Delta - \bar{\Delta}) \,, & \hat{j} = \hat{k} \,, \\ 0 \,, & o.w. \end{cases}$$

Then, it holds that $\boldsymbol{d}^T\boldsymbol{u} = \sum_{\hat{j}\in[K+1]} d_{\hat{j}}u_{\hat{j}} + v_1 + bv_{\hat{k}} = 2\Delta - 2\Delta + b\gamma(\Delta-\bar{\Delta}) - b\gamma(\Delta-\bar{\Delta}) = 0$ and $\bar{\boldsymbol{d}}^T\boldsymbol{u} = \sum_{\hat{j}\in[K+1]} \bar{d}_{\hat{j}}u_{\hat{j}} + v_1 + \bar{b}v_{\hat{k}} = 2\bar{\Delta} - 2\Delta + b\gamma(\Delta-\bar{\Delta}) - \bar{b}\gamma(\Delta-\bar{\Delta}) = 2\bar{\Delta} - 2\Delta + (b-\bar{b})\gamma(\Delta-\bar{\Delta}) = 2\bar{\Delta} - 2\Delta + 2(\Delta-\bar{\Delta}) = 0$.

Analysis of $\mathcal{B}$. Denote $\mathsf{BadHash}$ as the event that $h_1, \bar{h}_1, \ldots, h_{2\mathsf{q}}, \bar{h}_{2\mathsf{q}}$ are *not* all distinct. We now show that $\mathcal{B}$ wins the AOM-MISIS game if $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur.

We first show that if $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur, $\mathcal{B}$ does not abort. Suppose $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur. Then, the only step where $\mathcal{B}$ might abort is in Case 3 above. In Case 3, suppose $\boldsymbol{d} \neq \bar{\boldsymbol{d}}$ and $d_{K+1+(j-1)(\ell+1)+1+[\ell]} = \bar{d}_{K+1+(j-1)(\ell+1)+1+[\ell]}$ (in which case $\mathcal{B}$ aborts). Let (i, lr) be the $\widetilde{\text{PSIGNO}}$ query that corresponds to the j-th token *before* rewinding. Then, there exists $J \in [\mathsf{q}]$ such that $h_{2J-1} = \mathrm{H}_1(\mathsf{pk}, lr) = d_{K+1+(j-1)(\ell+1)+1+[\ell]}$. Since $\mathsf{BadHash}$ does not occur, the only possibility is that the $2J-1 < I$ and (i, lr) is also the $\widetilde{\text{PSIGNO}}$ query that corresponds to the j-th token *after* rewinding. Let $\boldsymbol{R}$ be the aggregated nonce computed from $\mathsf{CompPar}(\mathsf{pk}, lr)$. Denote $\hat{J} \in [\mathsf{q}]$ be the index such that $h_{2\hat{J}} = \mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R})$ before rewinding. From the simulation of the random oracles, it holds that $\hat{J} \leqslant J$ and thus $2\hat{J} \leqslant 2J \leqslant I$. Also, since $lr.\mathsf{msg} \neq \mu^*$ (o.w., $\mathcal{C}$ would not win the game), we have $2\hat{J} \neq I$ and thus $2\hat{J} < I$. Therefore, $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R})$ in the second execution of $\mathcal{C}$ is also $h_{2\hat{J}}$. This implies $\boldsymbol{d} = \bar{\boldsymbol{d}}$, which contradicts our assumption.

If $\mathcal{B}$ does not abort, we have $\|\hat{\boldsymbol{s}}\| \leqslant \|\boldsymbol{z}^*\| + \|\bar{\boldsymbol{z}}^*\| \leqslant 2\beta_{\mathsf{z}} = \beta_{\mathsf{s}}$ and

$$\left\|(\hat{b}_1 \cdot \sigma_1, \ldots, \hat{b}_Q \cdot \sigma_Q)\right\|_1 = \left\|2(h_I - \bar{h}_I)\sigma_{\mathsf{sk}}\right\|_1 \leqslant 4\sigma_{\mathsf{sk}}\beta_{\mathsf{c}} = \beta_{\mathsf{b}} \,.$$

It is left to bound $\|(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)\|$. For $j \in [\mathsf{q}_s]$, denote $\boldsymbol{v} := u_{K+1+(j-1)(\ell+1)+[\ell+1]}$. There are three cases as mentioned in the construction of $\mathcal{B}$. For the first case, $\|\boldsymbol{v}\| = 0$. For the second case, $\|\boldsymbol{v}\| = \left\|2c \cdot \sum_{(\hat{i},\hat{j})\in T_i\times[K+1]} \lambda_{\hat{i}}^{lr.\mathsf{SS}} M_{\hat{i},\hat{j}} w_{\hat{j}}\right\| \leqslant 2B_{\mathsf{ss}} \|c\| \leqslant 2B_{\mathsf{ss}}\sqrt{\beta_{\mathsf{c}}}$. For the third case, $\|\Delta\|_1 = \left\|c \cdot \sum_{(\hat{i},\hat{j})\in T_i\times[K+1]} \lambda_{\hat{i}}^{lr.\mathsf{SS}} M_{\hat{i},\hat{j}} w_{\hat{j}}\right\|_1 \leqslant B_{\mathsf{ss}}\beta_{\mathsf{c}}$ and similarly $\left\|\bar{\Delta}\right\|_1 \leqslant B_{\mathsf{ss}}\beta_{\mathsf{c}}$. Since $\|\gamma\| \leqslant \sqrt{N}$ (by lemma 1),

$$\begin{aligned} \|\boldsymbol{v}\|^2 &= \left\|\gamma(\bar{b}\Delta - b\bar{\Delta}), 0, \ldots, 0, -\gamma(\Delta - \bar{\Delta}), 0, \ldots, 0)\right\|^2 \\ &\leqslant 2(\|\Delta\|_1 + \left\|\bar{\Delta}\right\|_1)^2 \|\gamma\|^2 \leqslant 8NB_{\mathsf{ss}}^2\beta_{\mathsf{c}}^2 \,. \end{aligned}$$

Therefore, $\|(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)\| \leqslant 1/\sigma_{\mathsf{sk}} + \sqrt{\sum_{j\in[\mathsf{q}_s]} \|\boldsymbol{v}/\sigma_{\mathsf{r}}\|^2} \leqslant 1/\beta_{\mathsf{s}} + \beta_{\mathsf{c}}B_{\mathsf{ss}}\sqrt{8N\mathsf{q}_s}/\sigma_{\mathsf{r}} \leqslant \beta_{\mathsf{u}}$.

Since each of $h_1, h_3, \ldots, h_{2\mathsf{q}-1}$ and $h_2, h_4, \ldots, h_{2\mathsf{q}}$ are sampled uniformly from $\mathcal{S}_{\mathsf{b}}^{\ell}$ and $\mathcal{S}_{\mathsf{c}}$ respectively, $\mathsf{Pr}[\mathsf{BadHash}] \leqslant (2\mathsf{q})^2/|\mathcal{S}_{\mathsf{b}}|^{\ell} + (2\mathsf{q})^2/|\mathcal{S}_{\mathsf{c}}| \leqslant 8\mathsf{q}^2 2^{-2\kappa}$. Therefore,

$$\begin{aligned} \mathsf{Adv}_{\mathsf{par}}^{\text{aom-misis}}(\mathcal{B}, \kappa) &\geqslant \mathrm{acc}(\mathsf{Fork}^{\mathcal{C}}) - \mathsf{Pr}[\mathsf{BadHash}] \\ &\geqslant \mathsf{Adv}_{\mathsf{CTZ}}^{\text{ts-uf-0}}(\mathcal{A}, \kappa)^2/\mathsf{q} - 8\mathsf{q}^2 2^{-2\kappa} \,, \end{aligned}$$

which concludes the theorem.

□

6 Analysis of the EKT Construction

We apply AOM-MISIS to the analysis of the EKT construction and discuss the parameter selections in this section.

6.1 Construction and main security theorem

We recall the EKT construction from [EKT24] in Figure 10. A summary of the parameters used by the scheme is provided in Figure 7. The scheme also depends on a pseudorandom function PRF with suitable domain and range. The Lagrange coefficient $L_{U,j} \in \mathbb{Z}_q$ for any subset $U \subseteq [n]$ and $j \in U$ is defined as $L_{U,j} := \prod_{i \in U \setminus \{j\}} \frac{-i}{j-i}$.

The following theorem establishes TS-UF-4 security for EKT from the hardness of AOM-MISIS. In the theorem statement, $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}}(\mathcal{C}, \kappa)$ refers to the standard PRF advantage of an adversary $\mathcal{C}$. The full proof of Theorem 4 is given in Section 6.3.

Theorem 4 (TS-UF-4 of EKT). *For any integers $q = q(\kappa), k = k(\kappa), m = m(\kappa)$, any pseudorandom function PRF, and any TS-UF-4 adversary $\mathcal{A}$ making at most $\mathsf{q}_s = \mathsf{q}_s(\kappa)$ queries to PPO and $\mathsf{q}_h = \mathsf{q}_h(\kappa)$ queries to RO, given $\sigma_\mathsf{r} > 2N \cdot q^{\frac{1}{m} + \frac{1}{N(m-k)}}$, there exist an AOM-MISIS adversary $\mathcal{B}$ and a PRF adversary $\mathcal{C}$ running in time roughly two times that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\text{ts-uf-4}}_{\mathsf{EKT}[\mathsf{PRF}]}(\mathcal{A}, \kappa) \leqslant \sqrt{\mathsf{q}\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{B}, \kappa) + 8\mathsf{q}^3 2^{-2\kappa}} + n^2 \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}}(\mathcal{C}, \kappa) + \mathsf{q}_s^2 \cdot 2^{-2\kappa+1} .$$

where $\mathsf{q} = \mathsf{q}_h + \mathsf{q}_s + 1$ and $\mathsf{par} = (q, k, m, Q = 1 + \mathsf{q}_s(1+\ell), (\sigma_i)_{i \in [Q]}, \beta_\mathsf{s} = \left(2^{\nu_\mathsf{r}+2} + \beta_\mathsf{c} \cdot 2^{\nu_{\mathsf{pk}}+1}\right) \cdot \sqrt{Nk} + 4\beta_\mathsf{z}, \beta_\mathsf{b} = 4\sigma_\mathsf{sk}\beta_\mathsf{c}, \beta_\mathsf{u} = 1/\sigma_\mathsf{sk} + \beta_\mathsf{c}\sqrt{8N\mathsf{q}_s}/\sigma_\mathsf{r})$ with $\sigma_1 = \sigma_\mathsf{sk}$, $\sigma_{1+i} = \sigma_\mathsf{r}$ for $i \in [\mathsf{q}_s(\ell+1)]$.

Remark 3. We note that we can directly establish TS-UF-0 security of EKT from AOM-MISIS by applying the unforgeability theorem from [EKT24] and Lemma 7. However, the resulting parameters are worse than those in Theorem 4. Specifically, β_u becomes $1/\sigma_\mathsf{sk} + 2\beta_\mathsf{c}\sqrt{\ell N \mathsf{q}_s}/\sigma_\mathsf{r}$, and β_b becomes $4\sigma_\mathsf{sk}\sqrt{\beta_\mathsf{c} N}$.

Also, we recall the correctness theorem from [EKT24], which is needed for parameter selection later.

Theorem 5 (Correctness of EKT [EKT24]). *For any integers $1 < t \leqslant n$, any pseudorandom function PRF, given $\sigma_\mathsf{r} \geqslant \sqrt{(\log(2Nm) + \kappa)/\pi}$ and $\beta_\mathsf{z} \geqslant (\beta_\mathsf{c} 2^{\nu_{\mathsf{pk}}} + 2^{\nu_\mathsf{r}})\sqrt{mN} + e^{1/4}(2\beta_\mathsf{c}\sigma_\mathsf{sk} + \sigma_\mathsf{r}\sqrt{n(1+\ell)})\sqrt{N}(\sqrt{k} + \sqrt{m-k})$, the threshold signature scheme $\mathsf{EKT}[\mathsf{PRF}]$ is correct with correctness error negligible in κ.*

6.2 Parameter selection

In this section, we first discuss the asymptotic parameters selection derived from the security theorems and the hardness of AOM-MISIS, then compare these parameters with those proposed in [EKT24], and finally estimate the concrete efficiency based on the parameter selections. We also discuss how our parameters are compared to the parameters of Ringtail [BKL+24].

$\underline{\mathsf{Setup}(1^\kappa):}$
$\widehat{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\widehat{A}|\mathbb{I}_k]$
$par \leftarrow A$
For $i \in [n]$ do
  $\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \varnothing$
  $\mathsf{st}_i.\mathrm{mapPP} \leftarrow ()$
Return par

$\underline{\mathsf{KeyGen}():}$
$\widehat{\mathsf{sk}} \leftarrow\!\!\$\ \mathscr{D}_{\sigma_{\mathsf{sk}}}^m$
$\mathsf{pk} \leftarrow \left\lfloor 2A \cdot \widehat{\mathsf{sk}} \right\rceil_{\nu_{\mathsf{pk}}}$ // $\mathsf{pk} \in R_{q_{\nu_{\mathsf{pk}}}}^k$
For $(i,j) \in [n]\times[n]$ do
  $\mathsf{seed}_{i,j} \leftarrow \{0,1\}^\kappa$
$\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{t-1} \leftarrow\!\!\$\ R_q^{m-k}$
For $i \in [n]$ do
  $\boldsymbol{s}_i \leftarrow \widehat{\mathsf{sk}}_{[m-k]} + \sum_{j=1}^{t-1} \boldsymbol{a}_j i^j$
  $\mathsf{sk}_i \leftarrow (\boldsymbol{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[n]})$
Return $(\mathsf{pk}, (\mathsf{sk}_i)_{i\in[n]})$

$\underline{\mathsf{SPP}(\mathsf{st}_i):}$
For $j \in [0..\ell]$ do $\boldsymbol{r}_j \leftarrow\!\!\$\ \mathscr{D}_{\sigma_{\mathsf{r}}}^m$
For $j \in [0..\ell]$ do $\boldsymbol{R}_j \leftarrow A \cdot \boldsymbol{r}_j$
$pp \leftarrow (\boldsymbol{R}_j)_{j\in[0..\ell]}$
$\mathsf{st}_i.\mathrm{mapPP}(pp) \leftarrow (\boldsymbol{r}_j)_{j\in[0..\ell]}$
Return (pp, st_i)

$\underline{\mathsf{LPP}(i, pp, \mathsf{st}_0):}$
$\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \mathsf{st}_0.\mathrm{curPP}_i \cup \{pp\}$
Return st_0

$\underline{\mathsf{LR}(\mu, SS, \mathsf{st}_0):}$
If $\exists\ i \in SS : \mathsf{st}_0.\mathrm{curPP}_i = \varnothing$ then
  Return $\perp$
$lr.\mathsf{msg} \leftarrow \mu$; $lr.\mathsf{SS} \leftarrow SS$
For $i \in SS$ do
  Pick pp_i from $\mathsf{st}_0.\mathrm{curPP}_i$
  $lr.\mathsf{PP}(i) \leftarrow pp_i$
  $\mathsf{st}_0.\mathrm{curPP}_i \leftarrow \mathsf{st}_0.\mathrm{curPP}_i \backslash \{pp_i\}$
Return (lr, st_0)

$\underline{\mathsf{CompPar}(\mathsf{pk}, lr):}$
$\mu \leftarrow lr.\mathsf{msg}$
For $i \in lr.\mathsf{SS}$ do
  $(b_j)_{j\in[\ell]} \leftarrow \mathrm{H}_1(\mathsf{pk}, lr)$
  $(\boldsymbol{R}_{i,j})_{j\in[0..\ell]} \leftarrow lr.\mathsf{PP}(i)$
$\boldsymbol{R} \leftarrow \left\lfloor \sum_{i\in lr.\mathsf{SS}} \left(\boldsymbol{R}_{i,0} + \sum_{j\in[\ell]} b_j \boldsymbol{R}_{i,j} \right) \right\rceil_{\nu_{\mathsf{r}}}$
$c \leftarrow \mathrm{H}_2(\mathsf{pk}, \mu, \boldsymbol{R})$
Return $(\boldsymbol{R}, c, (b_j)_{j\in[\ell]})$

$\underline{\mathsf{PS}(lr, i, \mathsf{st}_i):}$
$pp_i \leftarrow lr.\mathsf{PP}(i)$
If $\mathsf{st}_i.\mathrm{mapPP}(pp_i) = \perp$ then return $(\perp, \mathsf{st}_i)$
$(\boldsymbol{r}_j)_{j\in[0..\ell]} \leftarrow \mathsf{st}_i.\mathrm{mapPP}(pp_i)$
$\mathsf{st}_i.\mathrm{mapPP}(pp_i) \leftarrow \perp$
$(\boldsymbol{R}, c, (b_j)_{j\in[\ell]}) \leftarrow \mathsf{CompPar}(\mathsf{st}_i.\mathsf{pk}, lr)$
$(\boldsymbol{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[n]}) \leftarrow \mathsf{st}_i.\mathsf{sk}$
$\mathsf{mask} \leftarrow \sum_{j\in lr.SS} \mathsf{PRF}(\mathsf{seed}_{i,j}, (\mathsf{pk}, lr))$
$\mathsf{mask}' \leftarrow \sum_{j\in lr.SS} \mathsf{PRF}(\mathsf{seed}_{j,i}, (\mathsf{pk}, lr))$
$\boldsymbol{z} \leftarrow \boldsymbol{r}_{0,[m-k]} + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}_{j,[m-k]}$
  $+2c \cdot L_{lr.\mathsf{SS},j} \cdot \boldsymbol{s}_j + \mathsf{mask} - \mathsf{mask}' \bmod q$
Return $((\boldsymbol{R}, \boldsymbol{z}), \mathsf{st}_i)$

$\underline{\mathsf{Agg}(\mathrm{PS}, \mathsf{st}_0):}$
$\boldsymbol{R} \leftarrow \perp$; $\boldsymbol{z} \leftarrow 0$
For $(\boldsymbol{R}', \boldsymbol{z}') \in \mathrm{PS}$ do
  If $\boldsymbol{R} = \perp$ then $\boldsymbol{R} \leftarrow \boldsymbol{R}'$
  If $\boldsymbol{R} \neq \boldsymbol{R}'$ then return $(\perp, \mathsf{st}_0)$
  $\boldsymbol{z} \leftarrow \boldsymbol{z} + \boldsymbol{z}'$
$c \leftarrow \mathrm{H}_2(\mathsf{pk}, \mu, \boldsymbol{R})$
$\boldsymbol{h} \leftarrow \boldsymbol{R} - \left\lfloor \widehat{A}\boldsymbol{z} - 2^{\nu_{\mathsf{pk}}} \cdot c \cdot \overline{\mathsf{pk}} \right\rceil_{\nu_{\mathsf{r}}}$ // $\boldsymbol{h} \in R_{q_{\nu_{\mathsf{r}}}}^k$
Return $((c, \boldsymbol{z}, \boldsymbol{h}), \mathsf{st}_0)$

$\underline{\mathsf{Vf}(\mathsf{pk}, \mu, sig):}$
$(c, \boldsymbol{z}, \boldsymbol{h}) \leftarrow sig$
If $\left\|(\boldsymbol{z}, 2^{\nu_{\mathsf{r}}}\overline{\boldsymbol{h}} \bmod q)\right\|_2 > \beta_{\mathsf{z}}$ then return 0
$c' \leftarrow \mathrm{H}_2(\mathsf{pk}, \mu, \left\lfloor \widehat{A}\boldsymbol{z} - 2^{\nu_{\mathsf{pk}}} \cdot c \cdot \overline{\mathsf{pk}} \right\rceil_{\nu_{\mathsf{r}}} + \boldsymbol{h})$
Return $(c' = c)$

Fig. 10. Lattice-based t-out-of-n threshold signatures $\mathsf{EKT}[\mathsf{PRF}]$, where PRF is a pseudorandom function. Here, $\mathrm{H}_1 : \{0,1\}^* \to \mathcal{S}_{\mathsf{b}}^\ell$ and $\mathrm{H}_2 : \{0,1\}^* \to \mathcal{S}_{\mathsf{c}}$. Also, $L_{lr.\mathsf{SS},j}$ denotes the Lagrange coefficient, and $\overline{\mathsf{pk}}, \overline{\boldsymbol{h}} \in R^k$ denote the lift (see Section 3.1 for more details) of pk and $\boldsymbol{h}$ respectively. Also, we remark that, as stated earlier, the public parameter par is implicitly given to all algorithms except Setup.

ASYMPTOTIC PARAMETER SELECTIONS. Denote β_{lwe} as the norm of the underlying MLWE assumption. Initially, we select $N, m, k, \beta_{\mathsf{lwe}}$ such that N is a power of $N \geqslant 2\kappa$, $m, k = \mathsf{poly}(\kappa)$, and $\beta_{\mathsf{lwe}} \geqslant m\log(N)$.[7] (We note that when estimating the concrete efficiency, we will enumerate through plausible $(N, m, k, \beta_{\mathsf{lwe}})$ tuples and pick the one that yields the best efficiency.) Then, we set other parameters as follows.

[7] This is for guaranteeing the underlying MLWE is hard.

- Set β_{c} as the smallest integer such that $2^{\beta_{\mathsf{c}}}\binom{N}{\beta_{\mathsf{c}}} \geqslant 2^{2\kappa}$.
- $\sigma_{\mathsf{sk}} = \max\{2\beta_{\mathsf{lwe}}\sqrt{mN}, \sqrt{\log(6mN)/\pi}\}$. The first term is usually the leading term.
- $\sigma_{\mathsf{r}} = \max\{\sigma_{\mathsf{sk}}\beta_{\mathsf{c}}\sqrt{8N\mathsf{q}_s}, \sqrt{(\log(2Nm)+\kappa)/\pi}, 2N \cdot q^{\frac{1}{m}+\frac{1}{N(m-k)}}\}$. The first term is usually the leading term.
- $\nu_{\mathsf{pk}} = \log_2(\sigma_{\mathsf{r}}/\beta_{\mathsf{c}})$ and $\nu_{\mathsf{r}} = \log_2(\sigma_{\mathsf{r}})$.
- β_{z} is set as shown in Theorem 5
- Denote $\beta_{\mathsf{sis}} = 2\beta_{\mathsf{z}} + 4\sigma_{\mathsf{sk}}\beta_{\mathsf{c}}\sqrt{mN}$.
- Select q such that the problem $\mathrm{MSIS}_{q,k,m,\beta_{\mathsf{sis}}}$ and the problem $\mathrm{MLWE}_{q,N,k,m,\beta_{\mathsf{lwe}}}$ are assumed to be exponentially hard in κ.

By Theorem 4 and Theorem 1 with $\varepsilon = 1/2$ and $\alpha = 2$ (we can further optimize the concrete bound by adjusting α), TS-UF-4 of EKT is implied by the hardness of $\mathrm{MSIS}_{q,k,m,\beta_{\mathsf{sis}}}$ and $\mathrm{MLWE}_{q,N,k,m,\beta_{\mathsf{lwe}}}$.

Comparison with [EKT24]. Although the prior work does not give a security reduction to standard lattice assumptions, they provide candidate asymptotic parameters based on the heuristic assumption that the selective version of AOM-MLWE is as hard as the adaptive version. Still, our asymptotic parameters are slightly better than their candidate asymptotic parameters provided in [EKT24]. The key difference lies in the choice of σ_{r}, which significantly impacts efficiency. In particular, the prior work requires $\sigma_{\mathsf{r}} = \Omega(\beta_{\mathsf{c}}\beta_{\mathsf{lwe}}N\sqrt{\mathsf{q}_s Nk})$, while we require $\sigma_{\mathsf{r}} = \Omega(\beta_{\mathsf{c}}\beta_{\mathsf{lwe}}N\sqrt{\mathsf{q}_s m})$. Therefore, there is roughly a factor of $\sqrt{N}$ improvement.

Comparison with Ringtail [BKL+24]. Ringtail is very close to EKT. The main differences are that the output space of hash function H_1 changes and the nonces $\boldsymbol{r}_j$ are sampled from unbalanced discrete Gaussian distributions. In particular, in Ringtail, the first $m-k$ entries of each nonce $\boldsymbol{r}_{j,[m-k]}$ is sampled from $\mathscr{D}_{\sigma_{\mathsf{r}}}^{m-k}$, while the rest $\boldsymbol{r}_{j,[(m-k+1)..m]}$ is sampled from $\mathscr{D}_{\sigma'_{\mathsf{r}}}^{k}$ with $\sigma'_{\mathsf{r}} \neq \sigma_{\mathsf{r}}$. Therefore, it is possible to compare the parameter selections directly.

The key difference still lies in the choice of σ_{r}. In particular, Ringtail requires $\sigma_{\mathsf{r}} = \Omega(\beta_{\mathsf{c}}\sqrt{\mathsf{q}_h})$, where q_h denotes the number of random oracle queries. Here we offer different trade-offs. A key drawback of their parameters is that σ_{r} depends on q_h, which is typically assumed to be much larger than q_s. This is because q_s refers to the number of online signing queries and is a system parameter that can be enforced, while q_h scales with the offline computational power of the adversary. However, if we set $\mathsf{q}_h = \mathsf{q}_s$, their σ_{r} is smaller than ours roughly by a factor of $\beta_{\mathsf{lwe}}N\sqrt{m}$.

Concrete efficiency. We show a set of concrete parameters and estimated efficiency for $\kappa \in \{128, 192, 256\}$ and $n = 1024$ in Figure 11. We derive the parameters following our parameter selections mentioned above, and similar to the CTZ scheme, we estimate the concrete hardness of MSIS and MLWE using the lattice estimator. We note that our concrete parameters are worse than those given in [EKT24], although in a similar ballpark. However, worse parameters are to be expected. This is because their parameter selection is based on their direct cryptanalysis of AOM-MLWE, whereas we rely on a reduction from two standard lattice assumptions. In a similar spirit, our parameters are worse than those claimed for Ringtail in [BKL+24], however this is to be expected, too, as they heuristically assume $\mathsf{q}_h = \mathsf{q}_s$ (also see the above discussion) to set parameters. In practice, however, we expect q_h to be best approximated conservatively by the running time of the adversary, and this can be as high as 2^{256}, whereas q_s could typically be 2^{60}. The authors of Ringtail were aware of this fact, and their choice was motivated by their conjecture that a better dependency would be possible. We confirm their conjecture for the case of EKT.

κ	$\log_2(q)$	k	m	σ_{sk}	σ_{r}	β_{z}	$\lvert\mathsf{pk}\rvert$	$\lvert sig\rvert$	Comm.
128	69.3	6	11	$2^{10.6}$	$2^{52.6}$	$2^{66.2}$	8.72KB	30.88KB	766.83KB
192	69.6	8	15	$2^{10.6}$	$2^{52.6}$	$2^{66.5}$	11.74KB	42.90KB	1.01MB
256	70.2	10	18	$2^{11.17}$	$2^{53.2}$	$2^{67.1}$	14.76KB	50.72KB	1.29MB

Fig. 11. The concrete parameters and estimated efficiency of the EKT scheme for $\kappa = 128, 192, 256$ and $n = 1024$. We set $(N, \ell, \beta_{\mathsf{c}}) = (512, 26, 64)$. The last second column denotes the communication complexity per signer.

Game Ideal-TUF$^{\mathcal{A}}(\kappa)$:

$\widehat{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\widehat{A}|\mathbb{I}_k]$
$\mathrm{H} \leftarrow\!\!\$\ \mathsf{TS.HF}$; $\mathrm{S} \leftarrow \varnothing$
For $i \in [n]$ do
  $\mathsf{st}_i.\mathrm{mapPP} \leftarrow ()$
$(\mu, sig) \leftarrow \mathcal{A}^{\textsc{Init},\mathrm{PPO},\textsc{PSignO},\mathrm{RO}}(A)$
Return $(\mu \notin \mathrm{S} \ \wedge\ \mathsf{Vf}(\mathsf{pk}, \mu, sig) = 1)$

Oracle $\textsc{Init}(\mathsf{CS})$:

Require: $\mathsf{CS} \subseteq [n]$ and $|\mathsf{CS}| < t$
$\mathsf{HS} \leftarrow [n]\backslash\mathsf{CS}$
$\mathsf{sk} \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_{\mathsf{sk}}}$
$\mathsf{pk} \leftarrow [2A \cdot \mathsf{sk}]_{\nu_{\mathsf{pk}}}$
Return pk

Oracle $\mathrm{RO}(x)$:

Return $\mathrm{H}(x)$

Oracle $\mathrm{PPO}(i)$:

Require: $i \in \mathsf{HS}$
$\boldsymbol{r}_j \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_{\mathsf{r}}}$
For $j \in [0..\ell]$ do $\boldsymbol{R}_j \leftarrow A \cdot \boldsymbol{r}_j$
$pp \leftarrow (\boldsymbol{R}_j)_{j\in[0..\ell]}$
$\mathsf{st}_i.\mathrm{mapPP}(pp) \leftarrow (\boldsymbol{r}_j)_{j\in[0..\ell]}$
Return pp

Oracle $\textsc{SignO}(lr)$:

Require: $lr.\mathsf{SS} \subseteq [n]$ and $|lr.\mathsf{SS}| \geqslant t$
$\mathsf{hon} \leftarrow lr.\mathsf{SS} \cap \mathsf{HS}$
If $\exists\ i \in \mathsf{hon} : \mathsf{st}_i.\mathrm{mapPP}(lr.\mathsf{PP}(i)) = \perp$ then
  Return $\perp$
$\mathrm{S} \leftarrow \mathrm{S} \cup \{lr.\mathsf{msg}\}$
$(c, b_1, \ldots, b_\ell) \leftarrow \mathsf{CompPar}(\mathsf{pk}, lr)$
$\boldsymbol{z} \leftarrow 2c \cdot \mathsf{sk}_{[m-k]}$
For $i \in \mathsf{hon}$ do
  $(\boldsymbol{r}_j)_{j\in[0..\ell]} \leftarrow lr.\mathsf{PP}(i)$
  $\mathsf{st}_i.\mathrm{mapPP}(lr.\mathsf{PP}(i)) \leftarrow \perp$
  $\boldsymbol{z} \leftarrow \boldsymbol{z} + \boldsymbol{r}_{0,[m-k]} + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}_{j,[m-k]}$
Return $\boldsymbol{z}$

Fig. 12. The Ideal-TUF game, where the algorithms $\mathsf{CompPar}$ and Vf are defined in Figure 10.

6.3 Security reduction of EKT

Unlike other security analyses [DKM+24, EKT24] of lattice-based threshold signatures that use masking techniques, our reduction follows a two-step approach, which we believe has independent interest.

In the first step, we reduce the TS-UF-4 game of $\mathsf{EKT}[\mathsf{PRF}]$ to an *ideal* unforgeability game Ideal-TUF for threshold signatures (defined in Figure 12). In this game, no secret sharing of signing key or masking is involved. Moreover, the adversary directly obtains an *aggregation* of all partial signatures from honest parties in $lr.\mathsf{SS}$ via a second-round signing oracle $\textsc{SignO}$, provided that the tokens in lr for honest signers are all valid. Intuitively, the ideal game captures the information hidden by the masks. In particular, all the secret key shares and the partial signatures (except their aggregation) are entirely hidden by the masks.

In the second step, we establish the hardness of the ideal unforgeability game based on AOM-MISIS. The approach is cleaner than prior proofs, as it clearly separates the effects of masks from the main security reduction to the AOM-MISIS problem. In particular, the first step relies only on the security of PRF, while the second step does not involve masking at all.

Concretely, Theorem 4 is a corollary from the following two lemmas.

Lemma 10. *For any integers $q = q(\kappa), k = k(\kappa), m = m(\kappa)$, any PRF scheme* PRF*, and any* TS-UF-4 *adversary $\mathcal{A}$ making at most $\mathsf{q}_s = \mathsf{q}_s(\kappa)$ queries to* PPO *and $\mathsf{q}_h = \mathsf{q}_h(\kappa)$ queries to* RO*, given $\sigma_\mathsf{r} > 2N \cdot q^{\frac{1}{m}+\frac{1}{N(m-k)}}$, there exists an* Ideal-TUF *adversary $\mathcal{B}$ making at most q_s queries to* PPO *and q_h queries to* RO *and an* PRF *adversary $\mathcal{C}$ running in time roughly the same as $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\text{ts-uf-4}}_{\mathsf{EKT}[\mathsf{PRF}]}(\mathcal{A},\kappa) \leqslant \mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{B},\kappa) + n^2 \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}}(\mathcal{C},\kappa) + \mathsf{q}_s^2 \cdot 2^{-2\kappa+1} \ .$$

where n denotes the number of signers.

Lemma 11. *For any integers $q = q(\kappa), k = k(\kappa), m = m(\kappa)$ and any* Ideal-TUF *adversary $\mathcal{A}$ making at most q_s queries to* PPO *and q_h queries, there exists an* AOM-MISIS *adversary $\mathcal{B}$ running in time roughly two times that of $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{A},\kappa) \leqslant \sqrt{\mathsf{q}\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{B},\kappa) + 8\mathsf{q}^3 2^{-2\kappa}} \ .$$

where $\mathsf{q} = \mathsf{q}_h + \mathsf{q}_s + 1$ and $\mathsf{par} = (q, k, m, Q = 1 + \mathsf{q}_s(1+\ell), (\sigma_i)_{i\in[Q]}, \beta_\mathsf{s} = \left(2^{\nu_\mathsf{r}+2} + \beta_\mathsf{c} \cdot 2^{\nu_{\mathsf{pk}}+1}\right) \cdot \sqrt{Nk} + 4\beta_\mathsf{z}, \beta_\mathsf{b} = 4\sigma_{\mathsf{sk}}\beta_\mathsf{c}, \beta_\mathsf{u} = 1/\sigma_{\mathsf{sk}} + \beta_\mathsf{c}\sqrt{8N\mathsf{q}_s}/\sigma_\mathsf{r})$ with $\sigma_1 = \sigma_{\mathsf{sk}}$, $\sigma_{1+i} = \sigma_\mathsf{r}$ for $i \in [\mathsf{q}_s(\ell+1)]$.

The rough idea behind the first reduction (Lemma 10) is as follows. Due to masking, the reduction can simulate the PSignO oracle by responding with a uniformly random vector, unless the adversary has made PSignO queries to all honest party in $lr.\mathsf{SS}$. In this case, the reduction queries its own SignO oracle to obtain an aggregated signature and derives the requested partial signature from it. Thus, the reduction only queries SignO for message $lr.\mathsf{msg}$ when all honest signers in $lr.\mathsf{SS}$ were queried. This implies that the set S of messages considered signed in TS-UF-4 is exactly the same as the set S defined in Ideal-TUF. The second reduction (Lemma 11) is the similar to the proof of Theorem 2, and its proof is provided in Section 6.4.

Proof (of Lemma 10).

Let $\mathcal{A}$ be a TS-UF-4 adversary described in the theorem. We show the lemma via the following series of games.

$\mathbf{G}_0$: This is the same as TS-UF-4. The game is formally defined in Figure 13

$\mathbf{G}_1$: The same as $\mathbf{G}_0$ except that in the oracle PSignO, the response $\boldsymbol{z}$ is computed in a different way, and the game aborts if there are two valid PSignO queries for the same input (i, lr) (denoted as BNonce). The game is formally defined in Figure 14. We first show that if BNonce does not occur, the game is identical to $\mathbf{G}_0$. From the description of $\mathbf{G}_1$, if $\mathrm{curSS}(lr) \neq \mathsf{hon}$, we have $\boldsymbol{z} = \boldsymbol{v} + \mathsf{mask_c} = \boldsymbol{r}_{0,[m-k]} + (\sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}_{j,[m-k]}) + 2c \cdot L_{lr.\mathsf{SS},i} \cdot \boldsymbol{s}_i + \mathsf{mask_h} + \mathsf{mask_c}$, which is exactly the same as $\mathbf{G}_0$. Otherwise, $\boldsymbol{z} = \mathrm{curSumR}(lr) + 2c \cdot \widehat{\mathsf{sk}}_{[m-k]} - \mathrm{curSum}(lr) - \sum_{j\in\mathsf{cor}} 2c \cdot L_{lr.\mathsf{SS},j} \cdot \boldsymbol{s}_j + \mathsf{mask_c}$. Here, $\mathrm{curSumR}(lr) = \sum_{i'\in\mathsf{hon}} \boldsymbol{r}^{(i')}_{0,[m-k]} + (\sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}^{(i')}_{j,[m-k]})$, where $\{\boldsymbol{r}^{(i')}_j\}_{j\in[0..\ell]}$ denotes the nonces for signer $i' \in \mathsf{hon}$, and $\mathrm{curSum}(lr) = \sum_{i'\in\mathsf{hon}\setminus\{i\}} \boldsymbol{v}^{(i')} = \sum_{i'\in\mathsf{hon}\setminus\{i\}} (\boldsymbol{r}^{(i')}_{0,[m-k]} + (\sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}^{(i')}_{j,[m-k]}) + 2c \cdot L_{lr.\mathsf{SS},i'} \cdot \boldsymbol{s}_{i'} + \mathsf{mask_h}^{(i')})$, where $(\cdot)^{(i')}$ denotes the value computed during the query (i', lr). Since $\widehat{\mathsf{sk}} = \sum_{i'\in lr.\mathsf{SS}} L_{lr.\mathsf{SS},i'} \cdot \boldsymbol{s}_{i'}$ and $\sum_{i'\in\mathsf{hon}} \mathsf{mask_h}^{(i')} = \sum_{i'\in\mathsf{hon}} \sum_{j\in\mathsf{hon}} (\mathsf{PRF}(\mathsf{seed}_{i',j}, (\mathsf{pk}, lr)) - \mathsf{PRF}(\mathsf{seed}_{j,i'}, (\mathsf{pk}, lr))) = 0$, we have $\boldsymbol{z} = \boldsymbol{r}^{(i)}_{0,[m-k]} + (\sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}^{(i)}_{j,[m-k]}) + 2c \cdot L_{lr.\mathsf{SS},i} \cdot \boldsymbol{s}_i + \mathsf{mask_h}^{(i)} + \mathsf{mask_c}$, which is identical to $\mathbf{G}_0$.

Game $\mathbf{G}_0^{\mathcal{A}}(\kappa)$:

$\widehat{A} \leftarrow\!\!\$\ R_q^{k\times(m-k)}$; $A \leftarrow [\widehat{A}|\mathbb{I}_k]$
$\mathrm{H} \leftarrow\!\!\$\ \mathsf{TS.HF}$; $\mathrm{S} \leftarrow \varnothing$; $\mathrm{curSS} \leftarrow ()$
$(\mu^*, sig^*) \leftarrow \mathcal{A}^{\textsc{Init},\textsc{PPO},\textsc{PSigno},\textsc{Open},\mathrm{RO}}(A)$
Return $(\mu^* \notin \mathrm{S} \ \wedge\ \mathsf{Vf}(\mathsf{pk}, \mu^*, sig^*) = 1)$

Oracle $\textsc{Init}(\mathsf{CS})$:

$\mathsf{HS} \leftarrow [n]\backslash\mathsf{CS}$
$\widehat{\mathsf{sk}} \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_{\mathsf{sk}}}$; $\mathsf{pk} \leftarrow \left\lfloor 2A \cdot \widehat{\mathsf{sk}} \right\rceil_{\nu_{\mathsf{pk}}}$
For $(i,j) \in [n] \times [n]$ do
  $\mathsf{seed}_{i,j} \leftarrow \{0,1\}^\kappa$
$\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{t-1} \leftarrow\!\!\$\ R_q^{m-k}$
For $i \in [n]$ do
  $\boldsymbol{s}_i \leftarrow \widehat{\mathsf{sk}}_{[m-k]} + \sum_{j=1}^{t-1} \boldsymbol{a}_j i^j$
  $\mathsf{sk}_i \leftarrow (\boldsymbol{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[n]})$
Return $(\mathsf{sk}_i)_{i\in\mathsf{CS}}$

Oracle $\mathrm{PPO}(i)$:

Require: $i \in \mathsf{HS}$
For $j \in [0..\ell]$ do $\boldsymbol{r}_j \leftarrow\!\!\$\ \mathscr{D}^m_{\sigma_r}$
For $j \in [0..\ell]$ do $\boldsymbol{R}_j \leftarrow A \cdot \boldsymbol{r}_j$
$pp \leftarrow (\boldsymbol{R}_j)_{j\in[0..\ell]}$
$\mathsf{st}_i.\mathrm{mapPP}(pp) \leftarrow (\boldsymbol{r}_j)_{j\in[0..\ell]}$
Return pp

Oracle $\textsc{PSigno}(i, lr)$:

Require: $lr.\mathsf{SS} \subseteq [n]$ and $i \in \mathsf{HS} \cap lr.\mathsf{SS}$
$pp_i \leftarrow lr.\mathsf{PP}(i)$
If $\mathsf{st}_i.\mathrm{mapPP}(pp_i) = \perp$ then return $\perp$
If $\mathrm{curSS}(lr) = \perp$ then
  $\mathrm{curSS}(lr) \leftarrow \{i\}$
Else $\mathrm{curSS}(lr) \leftarrow \mathrm{curSS}(lr) \cup \{i\}$
If $\mathrm{curSS} = lr.\mathsf{SS} \cap \mathsf{HS}$ then
  $\mathrm{S} \leftarrow \mathrm{S} \cup \{lr.\mathsf{msg}\}$
$(\boldsymbol{r}_j)_{j\in[0..\ell]} \leftarrow \mathsf{st}_i.\mathrm{mapPP}(pp_i)$
$\mathsf{st}_i.\mathrm{mapPP}(pp_i) \leftarrow \perp$
$(\boldsymbol{R}, c, (b_j)_{j\in[\ell]}) \leftarrow \mathsf{CompPar}(\mathsf{pk}, lr)$
$(\boldsymbol{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[n]}) \leftarrow \mathsf{sk}_i$
$\mathsf{mask} \leftarrow \sum_{j\in lr.SS} \mathsf{PRF}(\mathsf{seed}_{i,j}, (\mathsf{pk}, lr))$
$\mathsf{mask}' \leftarrow \sum_{j\in lr.SS} \mathsf{PRF}(\mathsf{seed}_{j,i}, (\mathsf{pk}, lr))$
$\boldsymbol{z} \leftarrow \boldsymbol{r}_{0,[m-k]} + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{r}_{j,[m-k]}$
  $+2c \cdot L_{lr.\mathsf{SS},j} \cdot \boldsymbol{s}_j + \mathsf{mask} - \mathsf{mask}' \bmod q$
Return $(\boldsymbol{R}, \boldsymbol{z})$

Oracle $\mathrm{RO}(x)$:

Return $\mathrm{H}(x)$

Fig. 13. The $\mathbf{G}_0$ game, where the algorithms $\mathsf{CompPar}$ and Vf are defined in Figure 10.

We now argue that BNonce occurs with a negligible probability. Since $\mathsf{st}_i.\mathrm{mapPP}(pp_i)$ is set to $\perp$ after the query (i, lr), BNonce occurs only if the PPO oracle generates a new token $(\boldsymbol{R}_0, \ldots, \boldsymbol{R}_\ell)$ that is exactly the same as pp_i. Therefore, by the following lemma from [DKM+24], the probability that this occurs is at most $\mathsf{q}_s^2 \cdot 2^{-N+1}$, and thus, since $N \geqslant 2\kappa$,

$$\mathsf{Adv}^{\mathbf{G}_1}(\mathcal{A}, \kappa) \geqslant \mathsf{Adv}^{\mathbf{G}_0}(\mathcal{A}, \kappa) - \mathsf{q}_s^2 \cdot 2^{-2\kappa+1} \ . \tag{8}$$

Lemma 12 (Lemma 3.8, [DKM+24]). *For any integers $q, m, k > 0$, any real number $\sigma > 0$ and any matrix $A \in R_q^{k\times m}$, denote a distribution $\mathcal{D}(A) := \{[A|\mathbb{I}_k] \cdot \boldsymbol{s} \mid \boldsymbol{s} \leftarrow\!\!\$\ \mathscr{D}_\sigma^{m+k}\}$. If $\sigma > 2N \cdot q^{\frac{1}{k+m}+\frac{1}{Nm}}$,*

$$\mathsf{Pr}_{A \leftarrow\!\!\$\ R_q^{k\times m}}[H_\infty(\mathcal{D}(A)) \geqslant N-1] \geqslant 1 - 2^{-N+1} \ ,$$

where $H_\infty(\mathcal{D}(A)) := -\log_2(\max_{\boldsymbol{x}'\in R_q^k} \mathsf{Pr}_{\boldsymbol{x} \leftarrow\!\!\$\ \mathcal{D}(A)}[\boldsymbol{x} = \boldsymbol{x}'])$ denotes the min-entropy of $\mathcal{D}(A)$.[8]

$\mathbf{G}_2$: The same as $\mathbf{G}_1$ except in the oracle $\textsc{PSigno}$, the aggregated mask $\mathsf{mask}_{\mathsf{h}}$ computed from honest parties' seeds are replaced with a uniformly random value. The game is formally defined in Figure 14. We can show this game is computationally close to $\mathbf{G}_1$ by first replacing each $\mathsf{PRF}(\mathsf{seed}_{i,j}, \cdot)$ for honest parties i and j with a truly random function, which incurs at most a

[8] We omit the bit dropping from the original lemma, as it is not needed here and it only reduces the min-entropy.

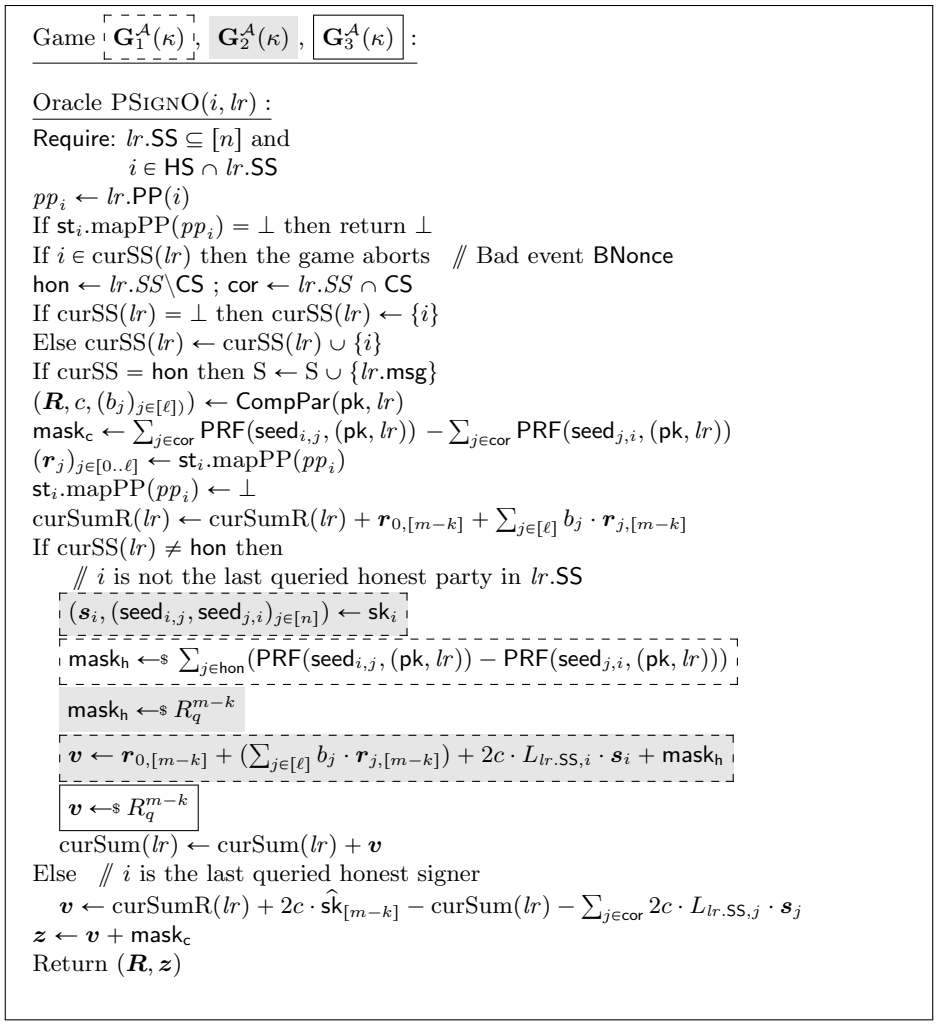

Fig. 14. The PSIGNO oracle of the games $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, where $\mathbf{G}_1$ only contains dashed boxes, $\mathbf{G}_2$ only contains highlighted boxes, and $\mathbf{G}_3$ only contains solid boxes. In addition, each entry of the tables curSum and curSumR is initialized to 0. The rest of each game is identical to $\mathbf{G}_0$.

reduction loss of $n^2 \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}}(\mathcal{C}, \kappa)$. Then, the game is identical to $\mathbf{G}_1$, since for each lr, denoting $\mathsf{hon} = lr.\mathsf{SS} \cap \mathsf{HS}$ and $\mathsf{mask_h}^{(i)} = \sum_{j\in\mathsf{hon}} (\mathsf{PRF}(\mathsf{seed}_{i,j}, (\mathsf{pk}, lr)) - \mathsf{PRF}(\mathsf{seed}_{j,i}, (\mathsf{pk}, lr)))$, we have $\{\mathsf{mask_h}^{(i)}\}_{i\in\mathsf{hon}\backslash\{i'\}}$ is uniformly distributed over $R_q^{k(|\mathsf{hon}|-1)}$ for any $i' \in \mathsf{hon}$. Therefore,

$$\mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) \geqslant \mathsf{Adv}^{\mathbf{G}_1}(\mathcal{A}, \kappa) - n^2 \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}}(\mathcal{C}, \kappa) \ . \tag{9}$$

$\mathbf{G}_3$: The same as $\mathbf{G}_2$ except the value $\boldsymbol{v}$ is uniformly sampled from R_q^{m-k} if i is not the last queried honest party in $lr.\mathsf{SS}$. The game is formally defined in Figure 14. Since $\mathsf{mask_h}$ is sampled uniformly from R_q^{m-k} and only used to mask $\boldsymbol{v}$, the distribution of $\boldsymbol{v}$ is identical in both games. Therefore, we have

$$\mathsf{Adv}^{\mathbf{G}_3}(\mathcal{A}, \kappa) = \mathsf{Adv}^{\mathbf{G}_2}(\mathcal{A}, \kappa) \ . \tag{10}$$

We construct an Ideal-TUF adversary $\mathcal{B}$ as follows. To start with, after receiving A from the Ideal-TUF game, $\mathcal{B}$ initializes st_i.mapPP for $i \in [n]$, tables curSS and curSum following the game $\mathbf{G}_3$, then initializes a map curLR to an empty map, recording whether a lr request has made, and runs $\mathcal{A}$ with input A and access to oracles $\widetilde{\text{INIT}}$, $\widetilde{\text{PPO}}$, $\widetilde{\text{PSIGNO}}$ and $\widetilde{\text{RO}}$, which are simulated as follows.

$\widetilde{\text{Init}}(\mathsf{CS})$: $\mathcal{B}$ queries $\mathsf{pk} \leftarrow \text{INIT}(\mathsf{CS})$. For each $i \in \mathsf{CS}$, $\mathcal{B}$ samples $\boldsymbol{s}_i \leftarrow\!\!\$\ R_q^{m-k}$ and $\mathsf{seed}_{i,j} \leftarrow\!\!\$\ \{0,1\}^\kappa$ for each $j \in [n]$. Finally, $\mathcal{B}$ returns $(\boldsymbol{s}_i, (\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i})_{j\in[n]})_{i\in\mathsf{CS}}$.

$\widetilde{\text{PPO}}, \widetilde{\text{RO}}$: $\mathcal{B}$ forwards queries directly to PPO and RO respectively.

$\widetilde{\text{PSignO}}(i, lr)$: The same as $\text{PSIGNO}(i, lr)$ in $\mathbf{G}_3$ except that when $\text{curSS}(lr) = \mathsf{hon}$, $\mathcal{B}$ queries $\hat{\boldsymbol{z}} \leftarrow \text{SIGNO}(lr)$ and sets $\boldsymbol{z} \leftarrow \hat{\boldsymbol{z}} - \text{curSum}(lr) - \sum_{j\in\mathsf{cor}} 2c \cdot L_{lr.\mathsf{SS},j} \cdot \boldsymbol{s}_j + \mathsf{mask}_\mathsf{c}$. Also, $\mathcal{B}$ does not need to retrieve $\{\boldsymbol{r}_{i,j}\}_{j\in[0..\ell]}$ and update curSumR, as the table curSumR is not used anymore.

After $\mathcal{A}$ returns, $\mathcal{B}$ outputs the output of $\mathcal{A}$.

We observe that $\mathcal{B}$ wins the Ideal-TUF game if $\mathcal{A}$ wins the game $\mathbf{G}_3$, since the message $lr.\mathsf{msg}$ is added to S in $\mathbf{G}_3$ if and only if $\text{curSS}(lr) = \mathsf{hon}$, which is exactly the scenario where $\mathcal{B}$ makes a SIGNO query on lr. Also, since $\mathcal{B}$ simulates the game $\mathbf{G}_3$ perfectly, it follows that $\mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{B}, \kappa) \geqslant \mathsf{Adv}^{\mathbf{G}_3}(\mathcal{A}, \kappa)$. Therefore, we can conclude the lemma by Equations (8) to (10). □

6.4 Proof of Lemma 11

Let $\mathcal{A}$ be a Ideal-TUF adversary as described in the lemma. W.l.o.g. we assume that $\mathcal{A}$ is deterministic. Also, we assume if $\mathcal{A}$ returns $(\mu^*, (\boldsymbol{R}^*, \boldsymbol{z}^*))$, the RO query $\text{H}_2(\mathsf{pk}, \mu^*, \boldsymbol{R}^*)$ was made by $\mathcal{A}$, which adds at most one RO query. Also, since the game makes at most one RO query to H_1 and H_2 respectively for each signing query, the total number of RO queries to each of H_1 and H_2 is bounded $\mathsf{q} = \mathsf{q}_h + \mathsf{q}_s + 1$. We now construct an algorithm $\mathcal{C}$ compatible with the syntax in Lemma 9 and construct $\mathcal{B}$ from $\mathsf{Fork}^{\mathcal{C}}$.

CONSTRUCTION OF $\mathcal{C}$. The input of $\mathcal{C}$ consists of $A, (\boldsymbol{t}_i)_{i\in[1+\mathsf{q}_s(\ell+1)]}$, and a list of hash values $h_1, \ldots, h_{2\mathsf{q}}$, where $(A, (\boldsymbol{t}_i)_{i\in[1+\mathsf{q}_s(\ell+1)]})$ are sampled following the AOM-MISIS game, and for each $i \in [\mathsf{q}]$, h_{2i-1} is sampled uniformly from $\mathcal{S}_\mathsf{b}$ and h_{2i} is sampled uniformly from $\mathcal{S}_\mathsf{c}$. To start with, $\mathcal{C}$ initializes st_i.mapPP $\leftarrow ()$ for $i \in [n]$, and in addition, initializes a counter $\text{ctr}_h \leftarrow 0$ for counting the number of random oracle queries. Then, $\mathcal{C}$ runs $\mathcal{A}$ on input A with access to oracles $\widetilde{\text{INIT}}$, $\widetilde{\text{PPO}}$, $\widetilde{\text{PSIGNO}}$ and $\widetilde{\text{RO}}$, which are simulated as follows.

$\widetilde{\text{Init}}(\mathsf{CS})$: The same as $\text{INIT}(\mathsf{CS})$ except $\mathcal{C}$ sets $\mathsf{pk} \leftarrow \lfloor 2\boldsymbol{t}_0 \rceil_{\nu_\mathsf{pk}}$.

$\widetilde{\text{PPO}}(i)$: For the j-th query, $\mathcal{C}$ sets $\boldsymbol{R}_{\hat{j}} \leftarrow \boldsymbol{t}_{1+(j-1)(\ell+1)+\hat{j}+1}$ for $\hat{j} \in [0..\ell]$ and sets

$$\mathsf{st}_i.\text{mapPP}((\boldsymbol{R}_{\hat{j}})_{\hat{j}\in[0..\ell]}) \leftarrow j\ .$$

Note that since $\mathcal{C}$ does not sample $(\boldsymbol{r}_{\hat{j}})_{\hat{j}\in[0..\ell]}$, $\mathcal{C}$ uses st_i.mapPP to store the index j instead.

$\widetilde{\text{SignO}}(lr)$: The same as $\text{SIGNO}(lr)$ except that $\mathcal{C}$ computes $\boldsymbol{z} \leftarrow \text{PI}(\boldsymbol{d})$, where

$$d_{\hat{j}} = \begin{cases} 2c\ , & \hat{j} = 1\ , \\ 1\ , & \hat{j} = 1 + (j-1)(\ell+1) + 1\ , j \in \mathsf{honR} \\ b_{j'}\ , & \hat{j} = 1 + (j-1)(\ell+1) + 1 + j', j' \in [\ell], j \in \mathsf{honR}\ , \\ 0\ , & o.w.\ , \end{cases} \tag{11}$$

and $\mathsf{honR} := \{\mathsf{st}_i.\text{mapPP}(lr.\text{PP}(i))\}_{i\in\mathsf{hon}}$.

$\widetilde{\mathbf{RO}}$ query $\mathrm{H}_1(x)$: If $\mathrm{H}_1(x) \neq \perp$, $\mathcal{C}$ returns $\mathrm{H}_1(x)$. Otherwise, parse x as $(\widetilde{\mathsf{pk}}, lr)$. If the parsing fails or $\widetilde{\mathsf{pk}} \neq \mathsf{pk}$, $\mathcal{C}$ sets $\mathrm{H}_1(x) \leftarrow\!\!\$\ \mathcal{S}_{\mathsf{b}}^{\ell}$ and returns $\mathrm{H}_1(x)$. Otherwise, $\mathcal{C}$ increases ctr_h by 1, sets $\mathrm{H}_1(x) \leftarrow h_{2\mathrm{ctr}_h-1}$. Also, $\mathcal{C}$ computes $\boldsymbol{R} \leftarrow \sum_{i\in lr.\mathsf{SS}}(\boldsymbol{R}_{i,0} + \sum_{j\in[\ell]} b_j \cdot \boldsymbol{R}_{i,j})$, where $(\boldsymbol{R}_{i,j})_{j\in[0..\ell]} \leftarrow lr.\mathsf{PP}(i)$ and $\{b_j\}_{j\in[\ell]} \leftarrow h_{2\mathrm{ctr}_h-1}$. If $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R}) = \perp$, $\mathcal{C}$ sets $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R}) \leftarrow h_{2\mathrm{ctr}_h}$. Finally, $\mathcal{C}$ returns $\mathrm{H}_1(x)$.

$\widetilde{\mathbf{RO}}$ query $\mathrm{H}_2(x)$: If $\mathrm{H}_2(x) \neq \perp$, $\mathcal{C}$ returns $\mathrm{H}_2(x)$. Otherwise, parse x as $(\widetilde{\mathsf{pk}}, \mu, \boldsymbol{R})$. If the parsing fails or $\widetilde{\mathsf{pk}} \neq \mathsf{pk}$, $\mathcal{C}$ sets $\mathrm{H}_2(x) \leftarrow\!\!\$\ \mathcal{S}_{\mathsf{c}}$. Otherwise, $\mathcal{C}$ increases ctr_h by 1 and sets $\mathrm{H}_2(x) \leftarrow h_{2\mathrm{ctr}_h}$. Finally, $\mathcal{C}$ returns $\mathrm{H}_2(x)$.

After receiving the output $(\mu^*, (c^*, \boldsymbol{z}^*, \boldsymbol{h}^*))$ from $\mathcal{A}$, $\mathcal{C}$ aborts if $\mathcal{A}$ does not win the Ideal-TUF game. Otherwise $\mathcal{C}$ computes $\boldsymbol{R}^* \leftarrow \left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}} \cdot c^* \cdot \mathsf{pk} \right\rceil_{\nu_{\mathsf{r}}} + \boldsymbol{h}^*$ and finds the index I such that $\mathrm{H}_2(\mathsf{pk}, \mu^*, \boldsymbol{R}^*)$ is set to h_I during the simulation. By our assumption of $\mathcal{A}$, we know such I must exist. Then, $\mathcal{C}$ returns $(I, \mathrm{Out} = (\mu^*, \boldsymbol{R}^*, c^*, \boldsymbol{z}^*, \boldsymbol{h}^*))$.

Analysis of $\mathcal{C}$. To use Lemma 9, we define $S := \{2j\}_{j\in[\mathsf{q}]}$ and IG as the algorithm that samples $(A, (\boldsymbol{t}_i)_{i\in[1+\mathsf{q}_s(\ell+1)]})$ following the AOM-MISIS game, and HG as the algorithm that samples h_{2i-1} uniformly from $\mathcal{S}_{\mathsf{b}}$ and samples h_{2i} uniformly from $\mathcal{S}_{\mathsf{c}}$ for each $i \in [\mathsf{q}]$. From the simulation, we know that the output index I of $\mathcal{C}$ is always in S. Also, it is not hard to check that $\mathcal{C}$ simulates the game Ideal-TUF perfectly, which implies $\mathrm{acc}(\mathcal{C}) \geqslant \mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{A}, \kappa)$. By Lemma 9, we have that

$$\mathrm{acc}(\mathsf{Fork}^{\mathcal{C}}) \geqslant \mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{A}, \kappa)^2/\mathsf{q}\ .$$

Construct $\mathcal{B}$ from $\mathsf{Fork}^{\mathcal{C}}$. We now construct the AOM-MISIS adversary $\mathcal{B}$ using $\mathsf{Fork}^{\mathcal{C}}$. To start with, $\mathcal{B}$ receives $(A, \{\boldsymbol{t}_i\}_{i\in[Q]})$ from the AOM-MISIS game with $Q = K + 1 + \mathsf{q}_s(\ell + 1)$ and runs $\mathsf{Fork}^{\mathcal{C}}(A, \{\boldsymbol{t}_i\}_{i\in[Q]})$ with access to the PI oracle from the AOM-MISIS game. If $\mathsf{Fork}^{\mathcal{C}}$ outputs $(I, \mathrm{Out} = (\mu^*, \boldsymbol{R}^*, c^*, \boldsymbol{z}^*, \boldsymbol{h}^*), \widetilde{\mathrm{Out}} = (\widetilde{\mu}^*, \widetilde{\boldsymbol{R}}^*, \widetilde{c}^*, \widetilde{\boldsymbol{z}}^*, \widetilde{\boldsymbol{h}}^*))$ and $c^* \neq \widetilde{c}^*$, $\mathcal{B}$ sets $\hat{\boldsymbol{s}} \leftarrow (\boldsymbol{z}^* - \widetilde{\boldsymbol{z}}^*, 2(c^* - \widetilde{c}^*)\boldsymbol{t}_0 - \widehat{A}(\boldsymbol{z}^* - \widetilde{\boldsymbol{z}}^*))$ and $\hat{\boldsymbol{b}} \leftarrow (2(c^* - \widetilde{c}^*), 0, \ldots, 0)$. Otherwise, $\mathcal{B}$ aborts. It is clear that $A\hat{\boldsymbol{s}} = \widehat{A}(\boldsymbol{z}^* - \widetilde{\boldsymbol{z}}^*) + 2(c^* - \widetilde{c}^*)\boldsymbol{t}_0 - \widetilde{A}(\boldsymbol{z}^* - \widetilde{\boldsymbol{z}}^*) = 2(c^* - \widetilde{c}^*)\boldsymbol{t}_0 = \sum_{i\in[Q]} \hat{b}_i \boldsymbol{t}_i$.

We now show how $\mathcal{B}$ sets $\boldsymbol{u}$ such that $\hat{\boldsymbol{b}}^T\boldsymbol{u} \neq 0$ and $\boldsymbol{d}^T\boldsymbol{u} = 0 \mod q$ for any oracle query $\mathrm{PI}(\boldsymbol{d})$. Note that $\mathcal{B}$ only makes PI queries while simulating oracle $\widetilde{\mathrm{SIGN}}\mathrm{O}$. We set $u_1 = 1$ and thus $\hat{\boldsymbol{b}}^T\boldsymbol{u} = 2(c^* - \widetilde{c}^*) \neq 0$.

Enumerating j from 1 to q_s, $\mathcal{C}$ sets $u_{1+(j-1)(\ell+1)+[\ell+1]}$ such that

$$\sum_{i\in[1+j(\ell+1)]} u_i d_i = 0 \text{ for each PI query } \boldsymbol{d} \text{ with } d_{1+[j(\ell+1)]} \neq \boldsymbol{0}\ . \tag{12}$$

To simplify notation in the following analysis, we use $\boldsymbol{v}$ to denote the vector $u_{1+(j-1)(\ell+1)+[\ell+1]} \in R^{\ell+1}$. Concretely, $\mathcal{C}$ sets $\boldsymbol{v}$ as follows. Suppose Equation (12) holds for $j-1$ (except when $j = 1$, i.e., no condition is required for the case $j = 1$). We say a $\widetilde{\mathrm{SIGN}}\mathrm{O}$ query lr corresponds to the j-th token if and only if it is the valid query with $\mathsf{st}_i.\mathrm{mapPP}(lr.\mathsf{PP}(i)) = j$, where a valid query means the $\widetilde{\mathrm{SIGN}}\mathrm{O}$ oracle does not return $\perp$. From the simulation, there is at most one $\widetilde{\mathrm{SIGN}}\mathrm{O}$ query corresponding to the j-th token during each execution of $\mathcal{A}$. Therefore, there are the following cases:

Case 1: No query corresponds to the j-th token during both executions. In this case, $\mathcal{B}$ set $\boldsymbol{v} \leftarrow \boldsymbol{0}$.

Case 2: Only one query corresponds to the j-th token during the two executions. Denote $\boldsymbol{d}$ as the PI query made during the execution of the $\widetilde{\mathrm{SIGN}}\mathrm{O}$ query corresponding to the j-th token,

where $\boldsymbol{d}$ follows the form given in Equation (11). $\mathcal{B}$ sets $\boldsymbol{v} \leftarrow (-\sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i, 0, \ldots, 0)$, and it follows $\sum_{i\in[1+j(\ell+1)]} u_i d_i = \sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i + v_1 = 0$. Also, since $\boldsymbol{d}$ is the PI query with $d_{1+(j-1)(\ell+1)+[\ell+1]} \neq \mathbf{0}$, it follows that Equation (12) holds.

Case 3: There is one query corresponding to the j-th token during each of the two executions. Denote $\boldsymbol{d}$ (resp. $\widetilde{\boldsymbol{d}}$) as the PI query made during the execution of the $\widetilde{\text{SIGNO}}$ query corresponding to the j-th token before (resp. after) rewinding. If $\boldsymbol{d} = \widetilde{\boldsymbol{d}}$, then $\mathcal{C}$ sets $\boldsymbol{v}$ in the same way as Case 2.

Otherwise, let $\hat{k} \in [\ell]$ be the index such that $d_{1+(j-1)(\ell+1)+1+\hat{k}} \neq \widetilde{d}_{1+(j-1)(\ell+1)+1+\hat{k}}$. (If such $\hat{k}$ does not exist, $\mathcal{B}$ aborts.) Denote $b := d_{1+(j-1)(\ell+1)+1+\hat{k}}$ and $\widetilde{b} := d_{1+(j-1)(\ell+1)+1+\hat{k}}$. Since Equation (12) holds for $j-1$, we have either $\sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i = 0$ or $\sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i = d_1 \in 2\mathcal{S}_{\mathsf{c}}$ by Equation (11), where $2\mathcal{S}_{\mathsf{c}} := \{2c \mid c \in \mathcal{S}_{\mathsf{c}}\}$. Therefore, denote $\Delta := \frac{1}{2}\sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i$ and $\widetilde{\Delta} := \frac{1}{2}\sum_{i\in[1+(j-1)(\ell+1)]} u_i \widetilde{d}_i$. Since $b, \widetilde{b} \in \mathcal{S}_{\mathsf{b}}$, by Lemma 1, there exists $\gamma \in R$ such that $\gamma(b - \widetilde{b}) = 2 \mod q$. $\mathcal{C}$ sets

$$v_{\hat{j}} \leftarrow \begin{cases} -2\Delta + b\gamma(\Delta - \widetilde{\Delta}) , & \hat{j} = 1 , \\ -\gamma(\Delta - \widetilde{\Delta}) , & \hat{j} = \hat{k} , \\ 0 , & o.w. \end{cases}$$

Then, it holds that $\sum_{i\in[1+j(\ell+1)]} u_i d_i = \sum_{i\in[1+(j-1)(\ell+1)]} u_i d_i + v_1 + b v_{\hat{k}} = 2\Delta - 2\Delta + b\gamma(\Delta - \bar{\Delta}) - b\gamma(\Delta - \bar{\Delta}) = 0$ and $\sum_{i\in[1+j(\ell+1)]} u_i \widetilde{d}_i = \sum_{i\in[1+(j-1)(\ell+1)]} u_i \widetilde{d}_i + v_1 + \widetilde{b} v_{\hat{k}} = 2\widetilde{\Delta} - 2\Delta + b\gamma(\Delta - \widetilde{\Delta}) - \widetilde{b}\gamma(\Delta - \widetilde{\Delta}) = 2\widetilde{\Delta} - 2\Delta + (b - \widetilde{b})\gamma(\Delta - \widetilde{\Delta}) = 2\widetilde{\Delta} - 2\Delta + 2(\Delta - \widetilde{\Delta}) = 0$. Finally, since $\boldsymbol{d}$ and $\widetilde{\boldsymbol{d}}$ are the PI queries with $d_{1+(j-1)(\ell+1)+[\ell+1]} \neq \mathbf{0}$, it follows that Equation (12) holds.

Analysis of $\mathcal{B}$. Denote $\mathsf{BadHash}$ as the event that $h_1, \widetilde{h}_1, \ldots, h_{2\mathsf{q}}, \widetilde{h}_{2\mathsf{q}}$ are *not* all distinct. We now show that $\mathcal{B}$ wins the AOM-MISIS game if $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur.

We first show that if $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur, $\mathcal{B}$ does not abort. (The following argument is similar to the one provided in the security reduction of the CTZ protocol.) Suppose $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur. Then, the only step where $\mathcal{B}$ might abort is in Case 3 above. In Case 3, suppose $\boldsymbol{d} \neq \widetilde{\boldsymbol{d}}$ and $d_{1+(j-1)(\ell+1)+1+[\ell]} = \widetilde{d}_{1+(j-1)(\ell+1)+1+[\ell]}$ (in which case $\mathcal{B}$ aborts). Let lr be the $\widetilde{\text{SIGNO}}$ query that corresponds to the j-th token *before* rewinding. Then, there exists $J \in [\mathsf{q}]$ such that $h_{2J-1} = \mathrm{H}_1(\mathsf{pk}, lr) = d_{1+(j-1)(\ell+1)+1+[\ell]}$. Since $\mathsf{BadHash}$ does not occur, the only possibility is that the $2J - 1 < I$ and lr is also the $\widetilde{\text{SIGNO}}$ query that corresponds to the j-th token *after* rewinding. Let $\boldsymbol{R}$ be the aggregated nonce computed from $\mathsf{CompPar}(\mathsf{pk}, lr)$. Denote $\hat{J} \in [\mathsf{q}]$ be the index such that $h_{2\hat{J}} = \mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R})$ before rewinding. From the simulation of the random oracles, it holds that $\hat{J} \leqslant J$ and thus $2\hat{J} \leqslant 2J \leqslant I$. Also, since $lr.\mathsf{msg} \neq \mu^*$ (o.w., $\mathcal{C}$ would not win the game), we have $2\hat{J} \neq I$ and thus $2\hat{J} < I$. Therefore, $\mathrm{H}_2(\mathsf{pk}, lr.\mathsf{msg}, \boldsymbol{R})$ in the second execution of $\mathcal{C}$ is also $h_{2\hat{J}}$. This implies $\boldsymbol{d} = \bar{\boldsymbol{d}}$, which contradicts our assumption.

Suppose $\mathcal{B}$ does not abort. From the way $\mathcal{B}$ sets $\boldsymbol{u}$, Equation (12) holds for $j = \mathsf{q}_s$. Then, since for each PI query $\boldsymbol{d}$, the component $d_{1+[\mathsf{q}_s(\ell+1)]} \neq \mathbf{0}$, it follows that $\boldsymbol{d}^T \boldsymbol{u} = \sum_{i\in[1+\mathsf{q}_s(\ell+1)]} u_i d_i = 0$. It is left to bound the norms of vectors $\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{b}}$ and $\boldsymbol{u}$.

We first bound the norm of $\hat{\boldsymbol{s}}$. We have

$$\|\hat{\boldsymbol{s}}[m-k]\| = \|\boldsymbol{z}^* - \widetilde{\boldsymbol{z}}^*\| \leqslant 2\beta_{\mathsf{z}} \tag{13}$$

and, by Lemma 13 (proved in [EKT24]),

$$
\begin{aligned}
\|\hat{\boldsymbol{s}}[(m-k+1)..m]\| &= \left\|2(c^*-\widetilde{c}^*)\boldsymbol{t}_0 - \widehat{A}(\boldsymbol{z}^*-\widetilde{\boldsymbol{z}}^*)\right\| \\
&\leqslant \left\|(c^*-\widetilde{c}^*)2^{\nu_{\mathsf{pk}}}\overline{\mathsf{pk}} - \widehat{A}(\boldsymbol{z}^*-\widetilde{\boldsymbol{z}}^*)\right\| + \left\|(c^*-\widetilde{c}^*)\cdot(2\boldsymbol{t}_0 - 2^{\nu_{\mathsf{pk}}}\overline{\mathsf{pk}})\right\| \\
&\leqslant \left\|(c^*-\widetilde{c}^*)2^{\nu_{\mathsf{pk}}}\mathsf{pk} - \widehat{A}(\boldsymbol{z}^*-\widetilde{\boldsymbol{z}}^*)\right\| + \beta_{\mathsf{c}}2^{\nu_{\mathsf{pk}}+1}\sqrt{Nk} \quad \text{(by Lemma 13)} \\
&\leqslant \left\|2^{\nu_{\mathsf{r}}}\overline{\left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} - 2^{\nu_{\mathsf{r}}}\overline{\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} \mod q\right\| \\
&\quad + \left\|(\widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}) - 2^{\nu_{\mathsf{r}}}\overline{\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}}\right\| \\
&\quad + \left\|(\widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}) - 2^{\nu_{\mathsf{r}}}\overline{\left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}}\right\| + \beta_{\mathsf{c}}2^{\nu_{\mathsf{pk}}+1}\sqrt{Nk} \\
&\leqslant \left\|2^{\nu_{\mathsf{r}}}\left(\overline{\left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} - \overline{\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}}\right) \mod q\right\| \\
&\quad + 2^{\nu_{\mathsf{r}}+1}\sqrt{Nk} + \beta_{\mathsf{c}}2^{\nu_{\mathsf{pk}}+1}\sqrt{Nk}\,. \quad \text{(by Lemma 13)}
\end{aligned}
\tag{14}
$$

Lemma 13 (Lemma 3.14 [EKT24]). *For any integers $v \geqslant 4$ and $q > 2^v$, let $q_v = \lfloor q/2^v \rceil$. Moreover, assume q and v satisfies $q_v = \lfloor q/2^v \rceil$. Then, for any $x \in \mathbb{Z}_q$, we have*

$$\left|x - 2^v \cdot \overline{\lfloor x \rceil_v}\right| \leqslant 2^v - 1\,.$$

Since $\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}} + \boldsymbol{h}^* = \boldsymbol{R}^* = \widetilde{\boldsymbol{R}}^* = \left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}} + \widetilde{\boldsymbol{h}}^*$, there exists $\boldsymbol{\delta} \in R^k$ such that $\|\boldsymbol{\delta}\|_\infty \leqslant 2$ and $\overline{\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} + \overline{\boldsymbol{h}^*} = \overline{\left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} + \overline{\widetilde{\boldsymbol{h}}^*} + q_{\nu_{\mathsf{r}}}\cdot\boldsymbol{\delta}$. Therefore,

$$
\begin{aligned}
&\left\|2^{\nu_{\mathsf{r}}}\left(\overline{\left\lfloor \widehat{A}\widetilde{\boldsymbol{z}}^* - 2^{\nu_{\mathsf{pk}}}\widetilde{c}^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}} - \overline{\left\lfloor \widehat{A}\boldsymbol{z}^* - 2^{\nu_{\mathsf{pk}}}c^*\mathsf{pk}\right\rceil_{\nu_{\mathsf{r}}}}\right) \mod q\right\| \\
&\leqslant \left\|2^{\nu_{\mathsf{r}}}\overline{\boldsymbol{h}^*} \mod q\right\| + \left\|2^{\nu_{\mathsf{r}}}\overline{\widetilde{\boldsymbol{h}}^*} \mod q\right\| + \left\|2^{\nu_{\mathsf{r}}}q_{\nu_{\mathsf{r}}}\cdot\boldsymbol{\delta} \mod q\right\| \\
&\leqslant 2\beta_{\mathsf{z}} + 2^{\nu_{\mathsf{r}}}\|\boldsymbol{\delta}\| \leqslant 2\beta_{\mathsf{z}} + 2^{\nu_{\mathsf{r}}+1}\sqrt{Nk}\,.
\end{aligned}
\tag{15}
$$

Therefore, by Equations (13) to (15), $\|\hat{\boldsymbol{s}}\| \leqslant \left(2^{\nu_{\mathsf{r}}+2} + \beta_{\mathsf{c}}\cdot 2^{\nu_{\mathsf{pk}}+1}\right)\cdot\sqrt{Nk} + 4\beta_{\mathsf{z}} \leqslant \beta_{\mathsf{s}}$.

Also, $\left\|(\hat{b}_1\cdot\sigma_1, \ldots, \hat{b}_Q\cdot\sigma_Q)\right\|_1 = \|2(c^*-\widetilde{c}^*)\sigma_{\mathsf{sk}}\|_1 \leqslant 4\sigma_{\mathsf{sk}}\beta_{\mathsf{c}} \leqslant \beta_{\mathsf{b}}$.

It is left to bound $\|(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)\|$. For $j \in [\mathsf{q}_s]$, denote $\boldsymbol{v} := u_{K+1+(j-1)(\ell+1)+[\ell+1]}$. There are three cases as mentioned in the construction of $\mathcal{B}$. For the first case, $\|\boldsymbol{v}\| = 0$. For the second case, since $v_1 \in 2\mathcal{S}_{\mathsf{c}} \cup \{0\}$, $\|\boldsymbol{v}\| = \|\boldsymbol{v}\| \leqslant 2\sqrt{\beta_{\mathsf{c}}}$. For the third case, since $\Delta, \widetilde{\Delta} \in \mathcal{S}_{\mathsf{c}} \cup \{0\}$, $\|\Delta\|_1 \leqslant \beta_{\mathsf{c}}$ and $\left\|\widetilde{\Delta}\right\|_1 \leqslant \beta_{\mathsf{c}}$. Since $\|\gamma\| = \sqrt{N}$ (by lemma 1),

$$
\begin{aligned}
\|\boldsymbol{v}\|^2 &= \left\|\gamma(\widetilde{b}\Delta - b\widetilde{\Delta}), 0, \ldots, 0, -\gamma(\Delta - \widetilde{\Delta}), 0, \ldots, 0)\right\|^2 \\
&\leqslant 2(\|\Delta\|_1 + \left\|\bar{\Delta}\right\|_1)^2\|\gamma\|^2 \leqslant 8N\beta_{\mathsf{c}}^2\,.
\end{aligned}
$$

Therefore, $\|(u_1/\sigma_1, \ldots, u_Q/\sigma_Q)\| \leqslant u_1/\sigma_{\mathsf{sk}} + \sqrt{\sum_{j\in[\mathsf{q}_s]} \|\boldsymbol{v}/\sigma_{\mathsf{r}}\|} \leqslant 1/\sigma_{\mathsf{sk}} + \beta_{\mathsf{c}}\sqrt{8N\mathsf{q}_s}/\sigma_{\mathsf{r}} \leqslant \beta_{\mathsf{u}}$. The above shows that $\mathcal{B}$ wins the AOM-MISIS game, given that $\mathsf{Fork}^{\mathcal{C}}$ returns and $\mathsf{BadHash}$ does not occur.

Finally, since each of $h_1, h_3, \ldots, h_{2\mathsf{q}-1}$ and $h_2, h_4, \ldots, h_{2\mathsf{q}}$ are sampled uniformly from $\mathcal{S}_{\mathsf{b}}^{\ell}$ and $\mathcal{S}_{\mathsf{c}}$ respectively, $\mathsf{Pr}[\mathsf{BadHash}] \leqslant (2\mathsf{q})^2/|\mathcal{S}_{\mathsf{b}}|^{\ell} + (2\mathsf{q})^2/|\mathcal{S}_{\mathsf{c}}| \leqslant 8\mathsf{q}^2 2^{-2\kappa}$. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\text{aom-misis}}_{\mathsf{par}}(\mathcal{B}, \kappa) &\geqslant \mathrm{acc}(\mathsf{Fork}^{\mathcal{C}}) - \mathsf{Pr}[\mathsf{BadHash}] \\ &\geqslant \mathsf{Adv}^{\text{ideal-tuf}}(\mathcal{A}, \kappa)^2/\mathsf{q} - 8\mathsf{q}^2 2^{-2\kappa} ,\end{aligned}$$

which concludes the lemma. □

Acknowledgments

This research was partially supported by NSF grants CNS-2026774, CNS-2154174, CNS-2426905, a gift from Microsoft, and a Stellar Development Foundation Academic Research Award.

References

AGHS13. Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 97–116. Springer, Berlin, Heidelberg, December 2013.

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.

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, Berlin, 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, Cham, August 2022.

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, Cham, December 2021.

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, Cham, August 2018.

BGGK17. Dan Boneh, Rosario Gennaro, Steven Goldfeder, and Sam Kim. A lattice-based universal thresholdizer for cryptographic systems. Cryptology ePrint Archive, Report 2017/251, 2017.

BKL+24. Cecilia Boschini, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Akira Takahashi, and Mehdi Tibouchi. Ringtail: Practical two-round threshold signatures from learning with errors. Cryptology ePrint Archive, Paper 2024/1113, 2024.

BL90. Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 27–35. Springer, New York, August 1990.

BLT+24. Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from DDH with full adaptive security. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 429–459. Springer, Cham, May 2024.

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.

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.

BTT22. Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi. MuSig-L: Lattice-based multi-signature with single-round online phase. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 276–305. Springer, Cham, August 2022.

BTZ22. Mihir Bellare, Stefano Tessaro, and Chenzhi Zhu. Stronger security for non-interactive threshold signatures: BLS and FROST. Cryptology ePrint Archive, Report 2022/833, 2022.

CATZ24. Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Partially non-interactive two-round lattice-based threshold signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 268–302. Springer, Singapore, December 2024.

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, Cham, August 2023.

CKGW22. Deirdre Connolly, Chelsea Komlo, Ian Goldberg, and Christopher A. Wood. Two-Round Threshold Schnorr Signatures with FROST. Internet-Draft draft-irtf-cfrg-frost-10, Internet Engineering Task Force, September 2022. Work in Progress.

Des88. Yvo Desmedt. Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 120–127. Springer, Berlin, Heidelberg, August 1988.

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

DKM+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, Cham, May 2024.

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, Cham, May 2021.

dPKPR24. Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. Raccoon: A masking-friendly signature proven in the probing model. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 409–444. Springer, Cham, August 2024.

EKT24. Thomas Espitau, Shuichi Katsumata, and Kaoru Takemure. Two-round threshold signature from algebraic one-more learning with errors. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 387–424. Springer, Cham, August 2024.

ENP24. Thomas Espitau, Guilhem Niot, and Thomas Prest. Flood and submerse: Distributed key generation and robust threshold signature from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 425–458. Springer, Cham, August 2024.

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Cham, May 2020.

GKS23. Kamil Doruk Gur, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice signatures from threshold homomorphic encryption. Cryptology ePrint Archive, Paper 2023/1318, 2023. https://eprint.iacr.org/2023/1318.

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.

HKL19. Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Cham, May 2019.

HKLN20. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 500–529. Springer, Cham, August 2020.

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, Cham, October 2020.

KM08. Neal Koblitz and Alfred Menezes. Another look at non-standard discrete log and diffie-hellman problems. *Journal of Mathematical Cryptology*, 2(4):311–326, 2008.

KRT24. Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 459–491. Springer, Cham, August 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.

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, Berlin, Heidelberg, May 2014.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009.

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

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, Berlin, Heidelberg, December 2005.

Natnt. National Institute of Standards and Technology. Post-Quantum Cryptography: Additional Digital Signature Schemes, 2022–Present. https://csrc.nist.gov/projects/pqc-dig-sig.

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, Cham.

Oka93. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Berlin, Heidelberg, August 1993.

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.

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, Berlin, Heidelberg, May 1997.

TT15. Katsuyuki Takashima and Atsushi Takayasu. Tighter security for efficient lattice cryptography via the Rényi divergence of optimized orders. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec 2015*, volume 9451 of *LNCS*, pages 412–431. Springer, Cham, November 2015.

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, Cham, April 2023.

A Linear secret sharing

We adopt the following definition from [CATZ24], simplifying the small coefficient property by bounding only the term required in the security proof (Section 5.3).

Definition 2 (Linear Threshold Secret Sharing with Small Coefficients). *Let* $1 < t \leqslant n$ *and* B_{ss} *be positive integers and* $\mathbb{G}$ *be an abelian group. A* t*-out-of-*n *linear threshold secret sharing scheme* $\mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ *for* $\mathbb{G}$ *consists of two algorithms* $(\mathsf{Share}, \mathsf{Recon})$ *with the following syntax:*

- $\mathsf{Share}(s \in \mathbb{G}; \boldsymbol{\rho} \in \mathbb{G}^K) \Rightarrow (\mathsf{ss}_j)_{j\in[L]} \in \mathbb{G}^L$*: takes as input a secret* $s \in \mathbb{G}$ *and a randomness vector* $\boldsymbol{\rho} \in \mathbb{G}^K$ *(sampled uniformly from* $\mathbb{G}^K$*), and returns the secret shares* $(\mathsf{ss}_j)_{j\in[L]}$*. We note that each party* $i \in [n]$ *has a subset of indices* $T_i \subseteq [L]$ *such that the share of party* i *is* $(\mathsf{ss}_j)_{j\in T_i}$*. We say that the individual share size of party* i *is* $|T_i|$*, the* ***total share size*** *is* L*, and the* ***randomness size*** *is* K*.*
- $\mathsf{Recon}(U, (\mathsf{ss}_j)_{j\in\bigcup_{i\in U} T_i}) \Rightarrow s \in \mathbb{G}$*: takes as input a set* $U \subseteq [n]$ *with* $|U| \geqslant t$ *and the secret shares corresponding to each party in* U*, and returns the reconstructed secret* s*.*

We require that $\mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$ *satisfies the following properties:*

- ***Linearity****: The sharing algorithm* Share *can be written as an integer matrix* $M \in \mathbb{Z}^{L\times(K+1)}$ *mapping a vector* $\boldsymbol{v} = (s, \rho_1, \ldots, \rho_K)^T \in \mathbb{G}^{K+1}$ *to* $M\boldsymbol{v} \in \mathbb{G}^L$*. We refer to* M *as the* ***sharing matrix*** *of* $\mathsf{SecSha}_{t,n,B_{\mathsf{ss}}}$*. Moreover, for any* $U \subseteq [n]$ *denote* M_U *as the matrix* M *restricted to the rows indexed with* $\bigcup_{i\in U} T_i$*, the following is also true:*
  - *For any* $U \subseteq [n], |U| \geqslant t$*, there exists a* ***reconstruction coefficient*** *vector* $\boldsymbol{\lambda}^U \in \mathbb{Z}^L$ *such that* $\lambda^U_j = 0$ *for* $j \notin \bigcup_{i\in U} T_i$ *and* $(\boldsymbol{\lambda}^U)^T M = (1, 0, \ldots, 0)$*. Then, the output of* $\mathsf{Recon}(U, \cdot)$ *on input* $(\mathsf{ss}_j)_{j\in\bigcup_{i\in U} T_i}$ *can be written as* $\sum_{i\in U}\sum_{j\in T_i} \lambda^U_j \mathsf{ss}_j$*. Hence, for* $(\mathsf{ss}_j)_{j\in[L]} \leftarrow \mathsf{Share}(s; \boldsymbol{\rho})$ *for any* $s \in \mathbb{G}$ *and* $\boldsymbol{\rho} \in \mathbb{G}^K$*, we have that* $\sum_{i\in U}\sum_{j\in T_i} \lambda^U_j \mathsf{ss}_j = s$*.*
  - *For any* $\mathsf{CS} \subseteq [n]$ *with* $|\mathsf{CS}| < t$*, there exists a vector* $\boldsymbol{w}^{\mathsf{CS}} \in \mathbb{Z}^{K+1}$ *such that* $w_1 = 1$ *and* $M_{\mathsf{CS}}\boldsymbol{w}^{\mathsf{CS}} = \boldsymbol{0}$*. We call such* $\boldsymbol{w}^{\mathsf{CS}}$ *the* ***sweeping vector*** *of* M_{CS}*.*
- ***Small Coefficients****: For any* $U \subseteq [n]$ *with* $|U| \geqslant t$ *and any* $\mathsf{CS} \subset [n]$ *with* $|\mathsf{CS}| < t$*, it holds that* $\sum_{(\hat{i},\hat{j})\in T_i\times[K+1]} \lambda^U_{\hat{i}} M_{\hat{i},\hat{j}} w^{\mathsf{CS}}_{\hat{j}} \leqslant B_{\mathsf{ss}}$*.*

Also, [CATZ24] shows the existence of such secret sharing scheme from the generic construction by Benaloh and Leichter [BL90], which can be stated as the following lemma.

Lemma 14 ([CATZ24]). *For any* $1 < t \leqslant n$*, there exists a* t*-out-of-*n *linear threshold secret sharing with small coefficients with total share size* $L = O(t'^{4.3} n \log n)$ *making the individual share size* $|T_i| \leqslant O(t'^{4.3} n \log n)$ *for* $t' = \min(t, n - t)$ *and the small coefficient bound* $B_{\mathsf{ss}} = O(t'^{4.3} n (\log n)^2)$*.*