LoTRS: Practical Post-Quantum Structured Threshold Ring Signatures from Lattices

Nikai Jagganath
nikai.jagganath@monash.edu
Monash University
CSIRO's Data61
Melbourne, Victoria, Australia

Muhammed F. Esgin
muhammed.esgin@monash.edu
Monash University
Melbourne, Victoria, Australia

Ron Steinfeld
ron.steinfeld@monash.edu
Monash University
Melbourne, Victoria, Australia

Amin Sakzad
amin.sakzad@monash.edu
Monash University
Melbourne, Victoria, Australia

Markku-Juhani O. Saarinen
markku-juhani.saarinen@tuni.fi
Tampere University
Tampere, Finland

Dongxi Liu
dongxi.liu@data61.csiro.au
CSIRO's Data61
Sydney, New South Wales, Australia

Abstract

Threshold ring signatures (TRS) enable a quorum of T users to jointly sign a message while hiding which T of the N ring members participated, supporting privacy-preserving endorsement in ad-hoc settings. That said, many deployments do not need anonymity over *every* T-subset of a ring: when the approval pattern is already public, a structured ring can be sufficient. In this work, we first formalize this setting as a *structured threshold ring signature* (sTRS) and introduce LoTRS, a lattice-based sTRS that avoids a dedicated leader and keeps interaction to the optimal number of two rounds by separating the *threshold signing relation* from the *anonymity mechanism*. To the best of our knowledge, LoTRS is the first construction in which a TRS variant is obtained by combining: (i) an aggregated signing layer: a two-round lattice-based multisignature protocol producing an aggregated signature relation, with (ii) a selection-hiding layer: a 1-out-of-N proof that hides the chosen ring element supporting that relation. While it is natural to use a T-out-of-N proof to build a TRS, our LoTRS exploits a 1-out-of-N proof to significantly improve efficiency. LoTRS concretely instantiates the aggregated signing layer using DualMS (Crypto'23) and the selection-hiding layer arising from Esgin et al.'s lattice-based one-out-of-many proof (IEEE S&P'22).

Our $(T, N \cdot T)$-LoTRS construction achieves polylog(N, T) signature size and outperforms (T, N)-TRS schemes significantly. For example, for $N = 100$ and $T = 50$, our signature size is only 36 KB, which is $\approx 3.5\times$ smaller than the previously best performing lattice-based scheme LastRings by Jeon et al (ISC'25). Our Rust reference implementation further supports practicality: for $T = 16$ and $N = 32$, i.e., structured ring size $T \cdot N = 512$, it produces 25 KB signatures, with mean signing time 149 ms and verification time 43 ms in a release build on a Ryzen AI 9 HX 370 laptop.

Keywords

Post-Quantum, Lattice, Threshold Ring Signature, Multi-Signature, Zero-Knowledge Proof

1 Introduction

Threshold ring signatures (TRS) introduced in the seminal work by Bresson et al. [6] strengthen both ring signatures and threshold signatures by combining a quorum requirement with signer anonymity in an *ad-hoc* setting. Informally, a TRS enables any T-out-of-N users to interactively produce a single signature on a message such that (i) the verifier is convinced that at least T distinct secret keys were used, yet (ii) the identities of those T signers remain hidden from outsiders among the N public keys forming the ring. The *ring* itself is formed spontaneously: the N public keys can be selected on the fly from a larger universe of available keys, without a group manager, enrollment authority, or centralized join procedure, and users may join freely with keys of their choice—hence the terminology "ring" rather than "group." Ring signatures, introduced by Rivest, Shamir, and Tauman [38] correspond to the special case $T = 1$, while the threshold variant was introduced to tolerate partial corruption: even if an adversary compromises fewer than T secret keys, it should still be infeasible to forge a valid signature. TRS are therefore well-suited for privacy-preserving application such as electronic voting, whistleblowing [34], and blockchain/cryptocurrency settings [22], where a statement or transaction should be endorsed by a quorum without revealing which members approved it. In such deployments, the ring must be sufficiently large to provide cover, and compact signatures are especially important for practical scalability.

At the same time, this flexibility makes threshold anonymity expensive to realize efficiently, especially in the post-quantum setting. In particular, once one insists on anonymity with respect to all size-T subsets of a ring, the privacy layer naturally tends toward a dedicated T-out-of-N zero-knowledge proof.

In many practical deployments, however, the relevant approval structure is not fully ad-hoc. Instead, the signer set is often constrained by a public organizational pattern: one representative per institution, one delegate per seat, one approver from each department, or one signer from each bucket of a fixed partition. In such settings, the verifier does not need anonymity with respect to *every* size-T subset of the ring; rather, the goal is to hide the actual signer set *within a public family of admissible subsets*. This motivates the study of a practically meaningful primitive, which we call a *structured threshold ring signature (sTRS)*.

At a high level, a sTRS retains the threshold-authenticity guarantee of a TRS, but replaces full ad-hoc threshold anonymity by anonymity relative to a public structure. Concretely, a public ring instance R comes equipped with a public descriptor Ψ that induces a family $\mathcal{F}_\Psi(R)$ of admissible signer sets. The signing protocol is

correct only for signer sets in $\mathcal{F}_\Psi(R)$, and anonymity is required only between pairs of admissible signer sets in this family. This relaxation allows the anonymity layer to focus on hiding *which structured hidden aggregate* was used, rather than proving membership of an *arbitrary* size-T subset. This type of structure is already familiar from RingCT-style protocols [18, 19, 35], though there it appears only in the single-signer case, without threshold signing. By contrast, in our sTRS, the true T-signer set is concealed among N candidate T-subsets of PK, of which one is real and the remaining $N-1$ are decoys; the proof certifies validity of one such subset without revealing which one (we discuss more on this in Sec. 1.2 and Fig. 1).

1.1 Related Work

Since its introduction a variety of attempts have been made to improve on the inefficient (signature size asymptotically $O(N \log N)$) TRS of Bresson et al. [6]. Liu et al. [26] presented a TRS scheme that allowed the use of both RSA-based and DL-based public keys at the same time, while other schemes ([36, 40, 42]) relied on the DDH problem. However, with the focus now on post-quantum or quantum-resistant cryptography, building TRS schemes from post-quantum assumptions has become the goal. One such path to construct post-quantum TRS schemes is coding theory [32], [10], however these turn out to be inefficient. For example, Lin et al. [25] point out that [32] has a 20MB signature. In a different case, Jeon et al. [24] point out that the recently proposed work by Chiang et al. [9] based on the VOLE-in-the-head paradigm and AES encryption has a signature size of over 200KB for $T = 200$. The most promising route in the post-quantum setting is arguably lattice-based cryptography with earlier works from Cayrel et al. [7] and Bettaieb et al. [3], and most recently LastRings [24], and the generic construction GC-TRS [25], which has two instantiations, namely LTRS and CTRS.

Lin et al.'s GC-TRS framework constructs lattice-based threshold ring signatures by having the T signers first interactively produce a single aggregated signature, and then attaching a dedicated T-out-of-N zero-knowledge proof showing that this aggregate was generated by T distinct members of the public ring. Their lattice instantiations are explicitly leaderless and require only two rounds of interaction, with signature size scaling either linearly in the ring size N (LTRS) or as $T \log N$ (CTRS); however, while CTRS achieves asymptotically compact sizes depending on N, it is noted that its concrete signature sizes are not yet ideal (see Table 1). In contrast, LastRings follows a different paradigm: signers first produce individual NIST standardized Falcon [20] signatures, and the final TRS is obtained by giving a succinct argument of knowledge that T valid signatures exist and correspond to members of the ring. To preserve signer privacy, this argument must be zero-knowledge; since LaBRADOR [4] is not inherently zero-knowledge, LastRings combines it with a separate lattice-based zero-knowledge layer, LNP [29], achieving logarithmic scaling in the ring size and reporting signatures of 122 KB for $N = 100$ and $T = N/2$. Their stated $O(\log T)$ signature bound should, however, be read with some care. In their concrete parameterization, the signature size still appears to depend on both the ring size and the threshold, suggesting that a more faithful joint asymptotic is at least $O(\log N + \log T)$. Nevertheless, these results still boast large signature sizes and make use of heavy machinery (e.g. LaBRADOR). Wu et al. [41] propose a rejection-sampling-free lattice-based threshold ring signature (RFTRS) and use it to construct a post-quantum, receipt-free, publicly verifiable e-voting scheme for untrusted cloud settings. Their main improvement is to replace Stern-like techniques with Gaussian convolution via the $G + G$ identification protocol [12], which removes rejection-sampling and avoids abort-and-restart repetition. To realize the T-out-of-N threshold, they use Lagrange interpolation: pseudo-signatures are simulated for non-signers, while the real signers' shares are embedded into an $(N-T)$-degree polynomial, giving threshold correctness and anonymity in a unified framework. The resulting RFTRS signature size is approximately $1677 \cdot N$ bytes.

This state of affairs hints at a useful distinction between two design goals. The first is the classical TRS goal: hide an arbitrary size-T subset of the ring. The second, weaker goal is to hide the signer set only within a public, application-driven family of admissible subsets. The latter still captures meaningful privacy in many structured workflows, while opening the door to significantly cheaper proof systems.

1.2 Our Contributions

A structured variant of TRS. We formalize a new variant of TRS, which we call a *structured threshold ring signature* (sTRS). In an sTRS, a public ring instance R is equipped with a public structure descriptor Ψ that induces a family $\mathcal{F}_\Psi(R)$ of admissible signer sets. Correctness is required only for signer sets in $\mathcal{F}_\Psi(R)$, and anonymity is defined only between pairs of signer sets in this family. This captures real world settings in which the approval pattern is public and constrained. In practice, an sTRS can be used in place of any TRS when the relaxed anonymity guarantee is acceptable. Our sTRS proposal, LoTRS, has the structure as depicted in Figure 1 where $M = T \cdot N$ public keys are viewed as a $T \times N$ table and all the real signers are in the ℓ-th column. Due to this structure, we refer to LoTRS to be a $(T, T \cdot N)$- or (T, M)-sTRS scheme and explicitly use different notations M and N to denote the full ring sizes in sTRS and TRS, respectively. We discuss the impact of this structure on anonymity at the end of this section. To the best of our knowledge, this structured-threshold approach is novel in the context of TRS, and we show that it can lead to substantial efficiency gains in TRS constructions.

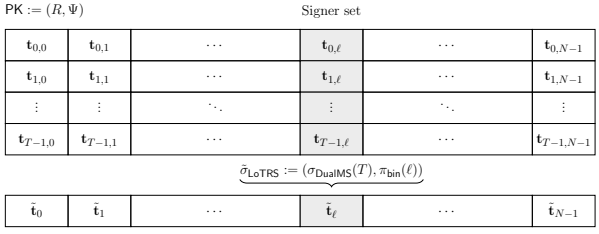

Figure 1: LoTRS **overview; structured ring** PK **and signature** $\tilde{\sigma}$**, with the signer column indexed by** ℓ **highlighted. We have** $\tilde{\mathbf{t}}_i := \sum_{u=0}^{T-1} \alpha_u \mathbf{t}_{u,i}$ **for random** α_u**'s.**

sTRS construction from a multisignature + a one-out-of-many proof. We present our efficient structured threshold ring signature scheme Lord of The RingS, denoted LoTRS, as our design indeed has "One Ring (Proof) to Rule them all". As shown in Table 1, for $N = 100$ and $T = N/2$ our threshold ring signature construction LoTRS outperforms the other schemes significantly; our signature size is $\approx 3.5\times$ smaller than the previously best performing TRS scheme LastRings.

We highlight the following technical aspects that, taken together, distinguish our construction from prior threshold ring signature designs. To the best of our knowledge, LoTRS is the first construction in which a TRS variant is obtained by combining (i) a two-round lattice-based multisignature protocol producing an aggregated signature relation, with (ii) a one-out-of-many proof that hides the chosen ring element supporting that relation. This design differs from the recent GC-TRS frameworks [25] that pair aggregated signing with an explicit T-out-of-N zero-knowledge proof, and from LastRings that prove knowledge of many underlying signatures via succinct arguments. In particular, due to the reliance on heavyweight T-out-of-N proofs, GC-TRS frameworks fall far behind practical expectations with signature sizes around MBs.

Turning T-out-of-N anonymity into a 1-out-of-N selection proof. A more obvious construction enforces threshold membership by proving, in zero knowledge, that an aggregated signing relation was produced by some subset of size T drawn from a ring of size N, which naturally leads to a T-out-of-N proof as the main privacy primitive (e.g., [25]). Our construction takes a different route: we enforce the threshold structure at the protocol layer (by arranging public keys as a table and aggregating across the T participating signers), so that the remaining privacy task is only to hide which ring element is "extracted" by the verifier's check. Consequently, our proof layer reduces to a one-out-of-many (1-out-of-N) selection proof: the verifier is convinced that the verification equation holds relative to *one* hidden (aggregated) ring component, while the threshold requirement is guaranteed by the multi-party signing structure.

Implementations and practical evaluation. We provide an artifact with two independent implementations of the concrete κ=1 instantiation ($N = \beta^\kappa$): a Python reference implementation used for readability and deterministic test-vector generation, and a performance-oriented Rust implementation that is cross-validated byte-for-byte against the Python vectors. The implementations cover the full protocol pipeline, including setup, key generation, aggregation, signing, verification, serialization, ring arithmetic, and the CDT and FACCT-style Gaussian samplers needed by the concrete parameter sets. They also include a SageMath-based estimator that reproduces the parameters and size calculations used in the paper. Our Rust benchmarks show that the construction is practical at the headline setting: for $N = 100$ and $T = 50$ (structured ring size $M = 5000$), signatures are about 36 KiB, signing takes about 0.8 s, and verification takes about 0.25 s in a release build with the per-signer loops parallelized via `rayon`. We give the full implementation methodology, benchmark table, and artifact description in §6.1.

Table 1: Comparison of our sTRS construction with other TRS schemes for ($N = 100$, $T = N/2$). Note that the full ring size in LoTRS is $M = N \cdot T$. See 'Comparing TRS and sTRS anonymity' in text for a discussion on the anonymity tradeoff between TRS and sTRS.

Work	Asymptotic Sig. Size	Sig. Size (analytic)	Security
⋆ Melchor et al.[32]	$O(N)$	2 MB	80 bit
⋆ Cayrel et al.[7]	$O(N)$	47 MB	111 bit
† Cayrel et al.[7]	$O(N)$	45 MB	111 bit
† Bettaieb et al.[3]	$O(NT)$	13 MB	111 bit
† GC-TRS(LTRS) [25]	$O(N)$	–	–
† GC-TRS(CTRS) [25]	$O(T \log N)$	–	–
† RFTRS [41]	$O(N)$	168 KB	120 bit
† LastRings [24]	$O(\log(NT))$	122 KB	128 bit
† LoTRS (**This work**)	polylog(N, T)	36 KB	128 bit
$*(N = 32, T = N/2)$		23 KB	128 bit
$*(N = 16, T = N/2)$		20 KB	128 bit

Key: ⋆ Code-based, † Lattice-based, $*$ smaller N estimates for LoTRS. Sizes are analytic estimates from each work's parameter selection; the codec-emitted wire sizes for LoTRS are reported in Appendix F and are slightly larger due to Golomb–Rice overhead and the binary KiB convention.

Comparing TRS and sTRS anonymity. When comparing our scheme with earlier TRS, we find it useful to distinguish between two notions of anonymity. The first is *signer-set anonymity*, namely the probability of guessing *all* real signers correctly. The second is *per-signer anonymity*, namely the probability of guessing a *single* real signer correctly. This distinction matters because anonymity is already lost once even a single signer is identified, so comparing schemes only through signer-set anonymity can be misleading. In the structured setting emphasized in this paper, a (T, M)-sTRS arranges the M public keys into T rows and $M/T := N$ columns of a table, so the true signing set is one column hidden among N candidate columns; in particular, each signer is hidden among N candidates within its row. Hence, a $(T, T \cdot N)$-sTRS achieves the same level of $1/N$ anonymity for both signer-set and per-signer anonymity notions. By contrast, a (T, N)-TRS may have a larger collection of admissible T-subsets, leading to $1/\binom{N}{T}$ signer-set anonymity. However, its per-signer anonymity may be quite small, i.e., T/N.

For example, with $T = 20$, a table-structured $(20, 2000)$-sTRS has 100 candidate columns, so both the signer-set anonymity and the per-signer anonymity are $1/100$. By contrast, an unstructured $(20, 100)$-TRS may admit up to $\binom{100}{20} \approx 5.4 \times 10^{20}$ candidate signer sets, while its per-signer anonymity is only $1/5$. Thus, a comparison based only on signer-set anonymity would strongly favor the latter, even though the former hides each signer much better. Our point is therefore not that our anonymity is uniformly stronger or weaker, but rather that it reflects a different tradeoff: better performance and stronger per-signer anonymity, at the cost of reduced signer-set anonymity. In practice, compromising one (or a few) signer(s) may be all an adversary needs because the adversary may force the compromised signer(s) to give out the names of other signers. We

focus on the table structure here because it is the main concrete setting of the paper, although the same perspective extends to other structured signer families.

In practice, both anonymity sets and signing quorums are often quite small. Monero, for example, currently fixes the ring size at 16, i.e., 15 decoys plus one real input [39]. Small-quorum threshold signing is likewise common in deployed wallet infrastructure: BitGo's MPC wallets use threshold signature schemes in a 2-of-3 configuration [5], while Frostsnap documents a simple 2-of-3 threshold setup and explicitly allows different thresholds such as 3-of-5 [21]. These examples suggest that small structured thresholds (such as the last-two rows of Table 1) are practically relevant rather than merely theoretical.

1.3 Technical Overview

Splitting signing and anonymity mechanisms. Our construction is split into two parts: we separate the threshold signing algebra (via a lattice multisignature) from the ring anonymity mechanism (via a lattice one-out-of-many proof), and then compose them in a way that yields a sTRS. As is expected when it comes to the lattice setting, "composing" protocols is not as straightforward as it sounds; in particular, a technical subtlety arises in the unforgeability reduction. Our proof does not extract only a constant number of accepting forks; instead, it must obtain $2(\kappa+1)$ accepting transcripts with common prefix. As a consequence, the extractor is an expected-time procedure, and its expected running time depends explicitly on the adversary's forgery probability. Accordingly, Lemma 5.6 is phrased as a time-success tradeoff rather than as a standard advantage-only bound.

*Two-round lattice multisignatures (*DualMS [8]*).* As our threshold signing backbone we build on the two-round lattice-based multisignature protocol DualMS [8]. At a high level, a multisignature lets a set of signers jointly authenticate a common message and output a single *aggregated* signature that is shorter than sending all individual signatures. DualMS is especially well-suited for our setting because it is explicitly designed for a two-round "exchange-and-aggregate" workflow and supports standard key aggregation techniques. Technically, its security proof highlights a simulation requirement that is specific to multisignatures: in the first round the signer must emit a commitment before the protocol's challenge is fixed (since that challenge depends on all parties' commitments), so the reduction must be able to simulate the signer *in-order* (called *straight-line simulation* in the multisignature literature) rather than generating transcripts in an arbitrary order as is common for single-signer Fiat–Shamir signatures. DualMS achieves this via a trapdoor-free *dual signing simulation* whose simulated signatures follow essentially the same structure as honest signatures (analogous DualMS proof in Appendix D.2). In our sTRS, LoTRS, we adapt this two-round aggregated-signature relation as the mechanism that enforces the quorum requirement: a valid aggregate can only be produced when at least T distinct secret keys contribute to the combined response.

Lattice one-out-of-many proofs. For ring anonymity, we build on the lattice-based one-out-of-many proof paradigm of Esgin et al. [16–19] through our subroutine $\mathsf{Sign}_{\mathsf{bin}}$ (Fig. 5). A standard one-out-of-many proof lets a prover convince a verifier that it knows a valid witness for *one* public item in a set, without revealing which one; under Fiat–Shamir, this is the usual route to lattice-based ring signatures with polylogarithmic communication and no trusted setup. The main difficulty in our setting is that we do not want to hide a *single* signer. Instead, we must hide a set of T independently keyed signers, while still ensuring that signing fails unless all T parties participate. A straightforward T-out-of-N zero-knowledge membership proof would lose the efficiency advantage of the one-out-of-many paradigm. Our key observation is that, in the table structure used by LoTRS, the hidden object is not an arbitrary threshold subset, but one admissible signer set—namely, one column of a public $T \times N$ table. We therefore use the one-out-of-many layer purely for *selection hiding*: it hides which column is used, while the fact that this column corresponds to T real signers is enforced separately by the aggregate signing algebra rather than by an explicit threshold-membership proof.

Structured ring instance and aggregate-compatible hiding. In LoTRS, a public ring instance consists of a fixed table $\mathsf{PK} = (\mathbf{t}_{u,i})$ with rows $u \in [0, T-1]$ and columns $i \in [0, N-1]$ of $T \cdot N$ public keys. The real signers occupy one hidden column $\ell \in [0, N-1]$, namely the set $\{\mathbf{t}_{u,\ell}\}_{u\in[0,T-1]}$. This structure lets us compress the T-signer statement into a one-column statement. Instead of proving knowledge of T secret keys among TN public keys, we derive random aggregation coefficients $(\alpha_u)_{u\in[0,T-1]}$ and form, for each column i, the aggregated public component $\tilde{\mathbf{t}}_i := \sum_{u=0}^{T-1} \alpha_u \mathbf{t}_{u,i}$. Each actual signer still holds and uses its own secret key, but the verifier checks only a *single* aggregate relation. This is why a multisignature-style combination is sufficient here: unlike classical threshold signatures, there is no fixed global public key whose secret key must be shared or reconstructed. The ring is formed ad hoc from individual public keys, and the hidden statement is that the signers controlling one column jointly authenticate the corresponding aggregate $\tilde{\mathbf{t}}_\ell$.

Compatibility of the multisignature and anonymity layers. The main technical challenge is that a naïve combination of a one-out-of-many proof with multisignature aggregation does not by itself guarantee that the anonymous selection layer and the signing layer refer to the *same* signer set. In principle, one could imagine an anonymity proof selecting one aggregate while the linear signing relation is satisfied with respect to a different combination of public keys. We solve this compatibility problem by modifying both the first-round commitments and the final responses of DualMS. Let $N = \beta^\kappa$. In Sign_1, each signer $u \in [0, T-1]$ samples masking randomness and broadcasts κ commitments $\mathbf{w}_{u,j}$, for $j \in [0, \kappa-1]$. Unlike in DualMS, we define each $\mathbf{w}_{u,j}$ with the additional term $\alpha_u \sum_{i=0}^{N-1} p_{i,j} \mathbf{t}_{u,i}$. We add this term so that, after aggregation across signers, the degree-j public-key contribution in the multisignature verification equation already matches the degree-j coefficient required by the selector polynomial. Without this extra term, the lower-degree part of the multisignature relation would not line up with the lower-degree part $\sum_{m=0}^{\kappa-1} \mathbf{C}_m x^m$ of the one-out-of-many identity (further discussion below), and the two layers could point

to different signer sets. We then show that, despite this extra public-key term, the modified commitments remain statistically close to uniform under our regularity condition.

In Sign_2, the signers first aggregate these commitments into $\tilde{\mathbf{w}}_j := \sum_{u=0}^{T-1} \mathbf{w}_{u,j}$, and then run an *identical* binary subproof $\mathsf{Sign}_{\mathsf{bin}}$ on the common tuple $(\tilde{\mathbf{w}}_j)_{j\in[0,\kappa-1]}$, obtaining a shared Fiat–Shamir challenge x. Because $\mathsf{Sign}_{\mathsf{bin}}$ takes only common inputs and a shared seed ρ, every signer would derive bit-identical π; the protocol description gives each signer's view as π_u, but a single shared π is computed once and broadcast in any concrete deployment (the implementation reported in Appendix F does exactly this, with SAgg asserting that the T transcripts agree on π). We then specifically engineer each signer's multisignature response $\mathbf{z}_u$ so that it contains the term $x^\kappa \alpha_u \mathbf{s}_u$; we make the analogous compatibility change on the dual side for $\mathbf{r}_u$. This degree-κ term is absent in pure DualMS, but we need it here because the selector proof isolates the hidden column exactly in degree κ.

More precisely, for each $i \in [0, N-1]$, write $i = (i_0, \ldots, i_{\kappa-1})$ in base β, then the binary proof certifies verifier-side values $f_{j,i_j} = a_{j,i_j} + x\, b_{j,i_j}$ that encode the hidden column index $\ell = (\ell_0, \ldots, \ell_{\kappa-1})$ blockwise in base β. Define

$$P_i(X) := \prod_{j=0}^{\kappa-1} \big(a_{j,i_j} + X\, b_{j,i_j}\big).$$

Then $P_i(X)$ is a degree-κ polynomial whose leading coefficient is 1 exactly when $i = \ell$, and 0 otherwise. Consequently, the verifier obtains the selector identity

$$\sum_{i=0}^{N-1} \Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big) \tilde{\mathbf{t}}_i = x^\kappa \tilde{\mathbf{t}}_\ell + \sum_{m=0}^{\kappa-1} \mathbf{C}_m x^m,$$

so only the hidden column ℓ survives in degree κ, while all other columns contribute only lower-degree terms (see Appendix D.4). We place the secret-key contribution in degree κ precisely so that, after aggregation, the multisignature equation authenticates the unique leading term $x^\kappa \tilde{\mathbf{t}}_\ell$, while the masking terms absorb the lower-degree coefficients $\mathbf{C}_m$.

This is the key novelty in the composition: we do not merely combine a one-out-of-many proof with a multisignature. We modify the DualMS commitments and responses so that the anonymity layer and the signing layer are forced to agree on the *same* hidden column ℓ. As a result, validity still requires contributions from all T secret keys in column ℓ, while anonymity comes from hiding which column is selected. This is exactly what lets LoTRS exploit the efficiency of a lattice 1-out-of-N proof without resorting to an explicit T-out-of-N zero-knowledge proof.

2 Preliminaries

We work over the cyclotomic polynomial rings $\mathcal{R} := \mathbb{Z}[X]/(f(X))$ and its modular analogues $\mathcal{R}_q := \mathbb{Z}_q[X]/(f(X))$, $\mathcal{R}_{\hat{q}} := \mathbb{Z}_{\hat{q}}[X]/(f(X))$ where $f(X) = X^d + 1$ for d a power of two (the $2d$-th cyclotomic polynomial), and $q, \hat{q}$ are primes chosen with $q, \hat{q} \equiv 5 \pmod 8$. Ring elements in $\mathcal{R}_q$ are represented with centered coefficients in $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$. For a vector of ring elements $\mathbf{v} \in \mathcal{R}^n$, we define $\|\mathbf{v}\|_p$ as the ℓ_p-norm of the length-nd coefficient embedding (i.e., concatenate all d coefficients of each component). A key technical fact is that most elements of $\mathcal{R}_q$, $\mathcal{R}_{\hat{q}}$ chosen in this way are invertible (Lemma A.1).

For index x, we write $\phi_x B_x$ for the standard deviation of the discrete Gaussian used to sample the corresponding masking vector. Thus, $\mathbf{y}_x \leftarrow D^m_{\phi_x B_x}$ means that each coefficient of $\mathbf{y}_x$ is sampled independently with standard deviation $\sigma_x = \phi_x B_x$. Here B_x bounds the relevant secret-dependent shift, while $\phi_x > 0$ is a rejection-sampling slack parameter chosen so that $\mathbf{y}_x + \mathbf{h}_x$ remains statistically close to the original Gaussian distribution after rejection sampling; details in Appendix A.2.

All logarithms are base 2 unless explicitly stated otherwise. A function negl is called negligible if $\mathsf{negl}(\lambda) < 1/\lambda^c$ for every $c > 0$ and all sufficiently large λ. For all integers within the interval $a \le b$, we write $[a, b] := \{a, a+1, \ldots, b\}$.

3 Structured Threshold Ring Signatures

In this section we introduce the definition of a structured threshold ring signature, a slight variant of the adversarial model of a threshold ring signature scheme as defined in [23].

Definition 3.1 (Structured Threshold Ring Signature Scheme). Define the ring $P := \{\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{pk}_2, \ldots\}$ to be the set of all public keys that are added to the system. Let $R \subseteq P$, $|R| = M$ be a public subring instance of registered public keys with public auxiliary descriptor Ψ. We always enumerate the elements of R as $0, \ldots, M-1$. The structure Ψ can be a table layout, partitioning of R, or any other public combinatorial object, so that admissible signer sets correspond to one approved pattern of selections across this structure. Let

$$\mathcal{F}_\Psi(R) \subseteq_\Psi \{U \subseteq R : |U| = T\}$$

be the public family of admissible signer sets induced by Ψ. For each $U \in \mathcal{F}_\Psi(R)$, let

$$S_U := \{\mathsf{sk}_i \mid \mathsf{pk}_i \in U\}$$

denote the corresponding set of signer secret keys. A (T, M)-*structured threshold ring signature scheme* is a tuple $\mathsf{sTRS} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$. On input the security parameter, output public parameters pp.
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})$. On input pp, generate a public/private key pair for user i.
- $\sigma \leftarrow \mathsf{Sign}(\mu, S_U, (R, \Psi))$. An (interactive) signing procedure run by the parties holding the secret keys in S_U, for some admissible signer set $U \in \mathcal{F}_\Psi(R)$, producing a signature σ on message μ w.r.t. the structured ring instance (R, Ψ).
- $b \leftarrow \mathsf{Vf}(pp, \mu, \sigma, (R, \Psi)) \in \{0, 1\}$. Deterministic verification that accepts iff σ is a valid structured threshold ring signature on μ w.r.t. (R, Ψ).

Remark 1. *Ordinary threshold ring signatures are recovered as the special case in which $\Psi = \perp$ and $\mathcal{F}_\perp(R) = \{U \subseteq_\Psi R : |U| = T\}$. In that case, anonymity ranges over all size-T subsets of R.*

Auxiliary descriptor Ψ. More concretely, Ψ should be understood as public structural data that specifies how the ring instance R is organized, and hence which size-T subsets are admissible. In our concrete LoTRS construction, Ψ describes a $T \times N$ table arrangement

of the public keys in R; writing the table as $(\mathsf{pk}_{u,i})_{u\in[0,T-1],\, i\in[0,N-1]}$, the admissible signer sets are exactly the columns, i.e.,

$$\mathcal{F}_\Psi(R) = \Big\{\{\mathsf{pk}_{0,i}, \mathsf{pk}_{1,i}, \ldots, \mathsf{pk}_{T-1,i}\} : i \in [0, N-1]\Big\}.$$

Thus, Ψ does not merely annotate the ring; it restricts the anonymity space to a structured family of candidate signer sets. This can be beneficial both conceptually and algorithmically: conceptually, it captures settings where the valid signing patterns are constrained by public system rules, and algorithmically, it allows a construction to exploit that structure instead of proving membership in an arbitrary size-T subset of R. As another example, Ψ could specify a partition $R = R_0 \sqcup R_1 \sqcup \cdots \sqcup R_{T-1}$, with admissible signer sets required to contain exactly one public key from each block R_j. Such a partitioned form could model, e.g., one signer per role, department, or region, while still allowing anonymity within each block.

Definitions of completeness, unforgeability, and anonymity for sTRS, analogous to those for TRS, are provided in Appendix B.

Remark 2. *A potential concern in the structured setting is that, if the same structural pattern is used repeatedly, the resulting signatures may exhibit temporal correlations because the same subset of signers appears together across multiple executions. This concern is legitimate, but it is not specific to the structured setting. Indeed, in a standard threshold ring signature scheme, repeated use of largely the same signers likewise induces cross-instance correlation, since an observer may combine information from multiple signatures by intersecting the candidate signer sets consistent with the corresponding rings. Thus, this should be viewed as the usual correlation issue of repeated threshold ring usage, rather than as a structural weakness of sTRS.*

4 LoTRS: Our Lattice-based Structured Threshold Ring Signature

In this section, we explain our construction, see Figs. 2-7. The main parameters referred to in the algorithms can be found in Table 2. To keep the protocol description readable, we omit from the algorithm listings a number of routine derived quantities, including several rejection sampling slack factors and norm bounds ($\phi_0, B_0, \phi_a, B_a, \phi_b, B_b$ etc.), and related parameters. These terms are important for concrete instantiation and for the formal correctness/unforgeability analyses, but they are not essential for following the core protocol mechanics. We therefore introduce only the quantities needed immediately in the algorithms, and defer the complete definitions and bound derivations to Appendix C.

Deterministic expansion. We write $\mathsf{Expand}(\rho, \tau; \mathcal{D})$ for a deterministic, domain-separated seed-expansion procedure (modeled as a random oracle). Concretely, Expand takes as input a seed ρ, a label τ, and a target distribution $\mathcal{D}$. It first computes a domain-separated byte stream from $\rho \,\|\, \tau$ using a fixed extendable-output function (XOF), and then deterministically parses this stream into an element of $\mathcal{D}$. Thus, equal inputs $(\rho, \tau, \mathcal{D})$ yield equal outputs, while distinct labels provide domain separation across different uses of the sampler. When $\mathcal{D}$ is a probability distribution (e.g. a discrete Gaussian or a bounded set like S_1^m), the parsing rule is the canonical sampler for $\mathcal{D}$ driven by the derived XOF stream; when $\mathcal{D}$ is a finite set, the parsing rule is uniform sampling over that set.

Table 2: Parameters of LoTRS.

Parameter(s)	Description
T	Threshold number of signers.
$N = \beta^\kappa$	Number of columns in structured ring instance PK.
d	Ring dimension (degree) of the cyclotomic ring.
$f(X) = X^d + 1$	The $2d$-th cyclotomic polynomial.
$\mathcal{R} = \mathbb{Z}[X]/f(X)$	Cyclotomic ring.
$\mathcal{R}_q = \mathbb{Z}_q[X]/f(X)$	Ring (for multisignature).
$\mathcal{R}_{\hat{q}} = \mathbb{Z}_{\hat{q}}[X]/f(X)$	Ring (for binary proof).
$q\ ;\ \hat{q}$	Modulus; binary proof modulus.
$\mathbf{A} \in \mathcal{R}_q^{k\times l}$	Public matrix of dimension $k \times l$.
$\mathbf{B} \in \mathcal{R}_q^{k\times l'}$	Public matrix of dimension $k \times l'$.
$\bar{\mathbf{A}} = [\mathbf{A} \mid \mathbf{I}] \in \mathcal{R}_q^{k\times(l+k)}$	Augmented matrix used for public key.
$\bar{\mathbf{B}} = [\mathbf{B} \mid \mathbf{I}] \in \mathcal{R}_q^{k\times(l'+k)}$	Augmented matrix used for dual key.
$K_A, K_B, K_{w,0}, \ldots, K_{w,\kappa-1}$	Quantities of bits dropped.

Signing session seed. In any TRS, a standard session-initialization phase must occur to identify the T participating signers for the current session and establish the ring R. This step is not part of Setup or KGen; rather, it is a standard session-initialization phase in TRS schemes in which the participating signers determine the communication endpoints for the current run and, in this case, establish the session seed ρ and column index $ell \in [0, N-1]$. For simplicity, we describe the basic realization in which one designated co-signer samples $\rho \in \{0,1\}^{256}$ and privately sends it to the other participating signers over authenticated confidential channels. The seed ρ is *ephemeral*: it is used only for the current signing session, is not part of any long-term secret key, and is discarded when the session ends. Rejection-sampling restarts do not require a new ρ; instead, fresh masking values are derived by expanding $\rho \,\|\, \mathrm{ctr}$ with an updated counter. A fresh independent ρ is shared only if the entire signing session is aborted and restarted.

The sole purpose of ρ is to synchronize the deterministic values derived through Expand, namely shared masks used in $\mathsf{Sign}_{\mathsf{bin}}$ and coefficients $p_{i,j}$ that are already needed in Sign_1. In particular, only these Expand-derived quantities become deterministic once ρ is fixed. The remaining signer-local masks $(\mathbf{y}_{u,j})_j$ and $(\mathbf{r}_{u,j})_j$ are still sampled freshly by each signer, so Sign_1 is not deterministic as a whole. The standard threshold ring signature anonymity notion, including ours, does not attempt to hide the signer set from a participating malicious co-signer: such a party already learns the co-signer identities directly from the protocol execution and can trivially reveal them. Accordingly, we only require ρ to support selection hiding against non-participating adversaries. On the other hand, adversarial choice of ρ does not by itself yield a forgery: the seed affects only the deterministic coefficients and masks in the one-out-of-many proof layer, whereas a valid signature still requires a valid aggregate DualMS relation over T secret-key contributions. In this sense, ρ is session randomness for the anonymity layer, not a source of threshold authenticity.

Key set and challenge space. We define the key set $S_\eta = \{s \in \mathcal{R} : \|s\|_\infty \le \eta\}$ and the challenge space $C := \{x \in \mathcal{R} : \|x\|_\infty = 1 \wedge \|x\|_1 = w\}$. Under assumptions in Section 5.1, every nonzero

difference $x - x'$ with $x, x' \in C$ is invertible over $\mathcal{R}_q$ and $\mathcal{R}_{\hat{q}}$.

Setup(1^λ)

$\rho_{\mathsf{pp}} \xleftarrow{\$} \{0,1\}^{256}$
return $\mathsf{pp} := \rho_{\mathsf{pp}}$

KGen(pp)

$\mathbf{A} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k \times l})$
$\bar{\mathbf{A}} := [\mathbf{A} \mid \mathbf{I}] \in \mathcal{R}_q^{k \times (l+k)}$
$\mathbf{s} \xleftarrow{\$} S_\eta^{l+k}$
$\mathbf{t} := \bar{\mathbf{A}}\,\mathbf{s} \in \mathcal{R}_q^k$
return $(\mathsf{sk} := \mathbf{s},\ \mathsf{pk} := \mathbf{t})$

KAgg$(\mathsf{PK} := (\mathbf{t}_{u,i}))$

for $u = 0, \ldots, T-1$ **do**
  $\alpha_u := H_{\mathrm{agg}}(\mathsf{PK}, u)$
endfor
for $i = 0, \ldots, N-1$ **do**
  $\tilde{\mathbf{t}}_i := \sum_{u=0}^{T-1} \alpha_u\, \mathbf{t}_{u,i}$
endfor
return $(\tilde{\mathbf{t}}_0, \ldots, \tilde{\mathbf{t}}_{N-1})$

Figure 2: Setup, key generation and key aggregation.

Structured ring instance. Fix a public structured ring instance $\mathsf{PK} := (R, \Psi)$, where R contains $M := T \cdot N$ independently generated public keys and Ψ specifies that these keys are arranged as a $T \times N$ table. Under this structure, the admissible family $\mathcal{F}_\Psi(R)$ consists exactly of the N column-sets, each containing one public key from every row. Thus, a signing session is associated with some signer set $S \in \mathcal{F}_\Psi(R)$, equivalently with some hidden column index $\ell \in [0, N-1]$. For actual signer in row u, only the entry $\mathbf{t}_{u,\ell}$ corresponds to its secret key $\mathbf{s}_u$; the remaining entries $\mathbf{t}_{u,i}$ for $i \neq \ell$ are decoy public keys belonging to other users in the public ring instance.

Setup and key generation [Fig. 2]. The setup algorithm samples a uniform public seed $\rho_{\mathsf{pp}} \xleftarrow{\$} \{0,1\}^{256}$, and sets $\mathsf{pp} := \rho_{\mathsf{pp}}$. All public matrices used in the scheme are then derived deterministically from ρ_{pp} via Expand with domain-separated labels. For key generation, a user constructs $\mathbf{A} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k \times l})$, and the augmented matrix $\bar{\mathbf{A}} := [\mathbf{A} \mid \mathbf{I}] \in \mathcal{R}_q^{k \times (l+k)}$. Each user then samples a short secret $\mathbf{s} \xleftarrow{\$} S_\eta^{l+k}$, and computes its public key $\mathbf{t} := \bar{\mathbf{A}}\mathbf{s} \in \mathcal{R}_q^k$. The key pair is $(\mathsf{sk}, \mathsf{pk}) := (\mathbf{s}, \mathbf{t})$.

Key aggregation (per column in PK*)* [Fig. 2]. Given $\mathsf{PK} = (\mathsf{PK}_i)_{i \in [0, N-1]}$, where column PK_i contains public keys $(\mathbf{t}_{u,i})_{u \in [0, T-1]}$, the aggregator first computes, for each row u, the coefficient $\alpha_u := H_{\mathrm{agg}}(\mathsf{PK}, u)$ i.e., by hashing PK together with the row index u. It then outputs the aggregated column keys $\tilde{\mathbf{t}}_i := \sum_{u=0}^{T-1} \alpha_u \mathbf{t}_{u,i}$ for all $i \in [0, N-1]$. The resulting $(\tilde{\mathbf{t}}_0, \ldots, \tilde{\mathbf{t}}_{N-1})$ is used by verification, and implicitly by the signing algebra.

Signature generation. Formally, our scheme exposes a single interactive signing algorithm $\mathsf{Sign}(\mu, S_U, (R, \Psi))$ that encapsulates the entire T-party signing session. The concrete procedures Sign_1 and Sign_2 should therefore be viewed only as signer-local round subprocedures inside this global Sign call. In particular, their inputs are all either determined by or generated during the execution of $\mathsf{Sign}(\mu, S_U, (R, \Psi))$. We write these local inputs explicitly in the protocol description, and omit the formal interface for simplicity. Every signer performs same local steps to generate a signature, so we describe the procedure for a single signer. Fix a message μ, and a signer in row u and hidden column ℓ of PK, with $\mathsf{sk}_u := \mathbf{s}_u$ and $\mathsf{pk}_u := \mathbf{t}_{u,\ell}$.

$\mathsf{Sign}_1(\mu, \mathbf{s}_u, \ell, \mathsf{PK}; \rho)$

Require: $\mathsf{PK} := \big((\mathbf{t}_{u,i})_{u,i}\big)$
$\alpha_u := H_{\mathrm{agg}}(\mathsf{PK}, u)$
$(a_{j,i})_{j,\, i \in [1, \beta-1]} := \mathsf{Expand}(\rho, \mathsf{a}|\mathsf{ctr}; D_{\phi_a B_a}^{\kappa(\beta-1)})$
  / Line 3: Derive the shared selector-polynomial coefficients.
Set $a_{j,0} := -\sum_{i=1}^{\beta-1} a_{j,i},\ \forall j$
$\mathbf{B} := H_{\mathrm{com}}(\mathsf{PK}, \mu) \in \mathcal{R}_q^{k \times l'}$
$\mathbf{y}_{u,1}, \ldots, \mathbf{y}_{u,\kappa-1} \xleftarrow{\$} D_{\phi_s B_s}^{l+k-1}$ $\mathbf{y}_{u,0} \xleftarrow{\$} D_{\phi_0 B_0}$
$\mathbf{r}_{u,1} \ldots, \mathbf{r}_{u,\kappa-1} \xleftarrow{\$} D_{\phi'_s B'_s}^{l'+k-1}$ $\mathbf{r}_{u,0} \xleftarrow{\$} D_{\phi_{0'} B_{0'}}$
$\Big(\mathbf{w}_{u,j} := \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \alpha_u \sum_{i=0}^{N-1} p_{i,j}\mathbf{t}_{u,i} \ \in \mathcal{R}_q^k\Big)_j$
  / Line 8: Signer commits via $w_{u,j}$'s to local contribution.
  / $p_{i,j}$'s derived from $a_{j,i}$'s (see Remark 3).
$\mathsf{st}_u := \Big(\ell, \mu, \mathsf{PK}, \mathbf{s}_u, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\Big)$
$\mathsf{Com}_u := (\mathbf{w}_{u,j})_j$
return $(\mathsf{st}_u,\ \mathsf{Com}_u)$

Figure 3: Signing, Round 1. Note that unless specified, $u \in [0, T-1]$ and $j \in [0, \kappa-1]$.

Signing, Round 1 [Fig. 3]. Assume that the T signers have already obtained the common session seed ρ. Once ρ is fixed, all quantities derived through Expand are fixed as well; however, Sign_1 is still randomized overall, since each signer samples fresh local masks. The signer in row u first computes its row-aggregation coefficient $\alpha_u := H_{\mathrm{agg}}(\mathsf{PK}, u)$, and expands ρ to obtain Gaussian coefficients $(a_{j,i})_{j \in [0, \kappa-1],\, i \in [1, \beta-1]} \leftarrow \mathsf{Expand}(\rho, \mathsf{a}|\mathsf{ctr}, D_{\phi_a B_a}^{\kappa(\beta-1)})$. For each j, it then sets $a_{j,0} := -\sum_{i=1}^{\beta-1} a_{j,i}$, so that $\sum_{i=0}^{\beta-1} a_{j,i} = \mathbf{0}$. Here, ctr denotes a restart counter. It is initialized to 0 in the beginning of a session and incremented after each rejection-sampling failure, so that $\mathsf{Expand}(\rho, \text{“}\cdot|\mathsf{ctr}\text{”}, \cdot)$ yields a fresh set of coefficients on every iteration, while remaining deterministic for fixed ρ.
Next, the signer derives the public matrix $\mathbf{B} := H_{\mathrm{com}}(\mathsf{PK}, \mu) \in \mathcal{R}_q^{k \times l'}$ and forms $\bar{\mathbf{B}} := [\mathbf{B} \mid \mathbf{I}]$. It then samples fresh Gaussian masks $(\mathbf{y}_{u,j})_{j \in [0, \kappa-1]}$ and $(\mathbf{r}_{u,j})_{j \in [0, \kappa-1]}$ from the respective distributions specified in Fig. 3. For each $j \in [0, \kappa-1]$, it computes the round-one commitment

$$\mathbf{w}_{u,j} \;:=\; \bar{\mathbf{A}}\mathbf{y}_{u,j} \;+\; \bar{\mathbf{B}}\mathbf{r}_{u,j} \;+\; \alpha_u \sum_{i=0}^{N-1} p_{i,j}\mathbf{t}_{u,i} \;\in\; \mathcal{R}_q^k,$$

where $p_{i,j}$'s are computed as in Remark 3. Finally, it sends to the other signers $\mathsf{Com}_u := (\mathbf{w}_{u,j})_{j \in [0, \kappa-1]}$, and stores the local state st_u, containing $(\ell, \mu, \mathsf{PK}, \mathbf{s}_u, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho)$, for use in round 2.

Remark 3 (Pre-computation of the coefficients $p_{i,j}$ [17]). *In* $\mathsf{Sign}_{\mathsf{bin}}$ *(Fig. 5), for each* $j \in [0, \kappa-1]$ *the prover commits to the*

vector $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})$ and proves that it is binary with Hamming weight one. Upon receiving the challenge x, the prover computes $f_{j,i_j} := x \cdot \delta_{\ell_j,i_j} + a_{j,i_j}$. Now consider the product $p_i(x) := \prod_{j=0}^{\kappa-1} f_{j,i_j}$. For every $i \in [0, N-1]$, one has

$$p_i(x) = \prod_{j=0}^{\kappa-1} \left(x\,\delta_{\ell_j,i_j} + a_{j,i_j}\right) = \delta_{\ell,i}\,x^{\kappa} + \sum_{j=0}^{\kappa-1} p_{i,j}\,x^{j},$$

for coefficients $p_{i,j}$ that depend only on ℓ and the values a_{j,i_j}. In particular, $(p_{i,j})_{j\in[0,\kappa-1]}$ are exactly the coefficients of degree strictly less than κ obtained by expanding $\prod_{j=0}^{\kappa-1}(x\,\delta_{\ell_j,i_j} + a_{j,i_j})$ as a polynomial in x. Hence they can be computed by the prover before the challenge x is known.

$\mathsf{Sign}_2(\mathsf{st}_u, \{\mathsf{Com}_{u'}\}_{u'\in[0,T-1]\setminus\{u\}}, \mathsf{PK})$

Require: $\mathsf{st}_u := \left(\ell, \mu, \mathsf{PK}, \mathbf{s}_u, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\right)$

Require: $\{\mathsf{Com}_{u'} := (\mathbf{w}_{u',j})_j\}_{u'\in[0,T-1]\setminus\{u\}}$

$\left(\tilde{\mathbf{w}}_j := \mathbf{w}_{u,j} + \sum_{u'\in[0,T-1]\setminus\{u\}} \mathbf{w}_{u',j}\right)_j$

/ Line 3: Aggregate first-round commitments of T participating signers.

for $j = 0, \ldots, \kappa-1$ **do**

  Decompose $\tilde{\mathbf{w}}_j := 2^{K_{w,j}}\tilde{\mathbf{w}}_j^{(1)} + \tilde{\mathbf{w}}_j^{(0)}$

endfor

$\pi_u := \mathsf{Sign}_{\mathsf{bin}}\left(\ell, \mu, (\tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}), \mathsf{PK}; \rho\right)$

/ Line 7: Prove signer set corresponds to one valid column of PK.

Parse π_u to obtain x

$M_w := \sum_{j=1}^{\kappa-1} \|x^j\|_1\, 2^{K_{w,j}-1}$

if $\|\tilde{\mathbf{w}}_0^{(0)}\|_\infty > 2^{K_{w,0}-1} - M_w$ **then** Restart

/ for $\kappa = 1$, $M_w = 0$.

$\mathbf{z}_u := x^{\kappa}\alpha_u\mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j\mathbf{y}_{u,j} \in \mathcal{R}^{l+k}$

/ Line 12: Main signer-local response under common Fiat–Shamir challenge.

if $\mathsf{Rej}\left(\mathbf{z}_u, x^{\kappa}\alpha_u\mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j\mathbf{y}_{u,j}, \phi_0, B_0\right)$ **then** Restart

$\mathbf{r}_u := -\sum_{j=0}^{\kappa-1} x^j\mathbf{r}_{u,j} \in \mathcal{R}^{l'+k}$

/ Line 14: Form auxiliary response used in Vf linear consistency check.

return $\sigma_u := (\pi_u,\ \mathbf{z}_u,\ \mathbf{r}_u)$

Figure 4: Signing, Round 2. Note that unless specified, $u \in [0, T-1]$ and $j \in [0, \kappa-1]$.

Signing, Round 2 [Fig. 4]. Let st_u denote signer u's local state output by Sign_1, namely $(\ell, \mu, \mathsf{PK}, \mathbf{s}_u, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho)$ for $j \in [0, \kappa-1]$. Let $\{\mathsf{Com}_{u'}\}_{u'\in[0,T-1]\setminus\{u\}}$ denote the round-one broadcasts of the other signers, where $\mathsf{Com}_{u'} := (\mathbf{w}_{u',j})_{j\in[0,\kappa-1]}$ contains the round-one commitments. Signer u first forms the aggregated commitments $\tilde{\mathbf{w}}_j := \sum_{u'=0}^{T-1}\mathbf{w}_{u',j}$ for all $j \in [0, \kappa-1]$, and then writes the coefficient-wise centered decomposition $\tilde{\mathbf{w}}_j = 2^{K_{w,j}}\tilde{\mathbf{w}}_j^{(1)} + \tilde{\mathbf{w}}_j^{(0)}$. Next, signer u runs the binary selection proof $\mathsf{Sign}_{\mathsf{bin}}$ on the secret column index ℓ, using the message μ, the high-bit aggregated commitments $(\tilde{\mathbf{w}}_j^{(1)})_{j\in[0,\kappa-1]}$, and the table PK, to obtain a transcript π. Since all honest signers use the same inputs $(\ell, \mu, \mathsf{PK}, (\tilde{\mathbf{w}}_j^{(1)})_j, \rho)$, this transcript is identical for all of them. Intuitively, $\mathsf{Sign}_{\mathsf{bin}}$ proves that ℓ is encoded as a block-wise one-hot vector in base β (for $N = \beta^{\kappa}$), so that verifier's selector polynomial isolates the unique column ℓ while keeping its value hidden.

$\mathsf{Sign}_{\mathsf{bin}}(\mathsf{pp}, \ell, \mu, (\tilde{\mathbf{w}})_{j\in[0,\kappa-1]}, \mathsf{PK}; \rho)$

Require: $\mathsf{PK} := \left(\mathsf{PK}_i := (\mathbf{t}_{u,i})_{u\in[0,T-1]}\right)_i,\ N = \beta^{\kappa}$

$\mathbf{G} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{G}; \mathcal{R}_{\hat{q}}^{\hat{n}\times(\hat{n}+\hat{k}+2\kappa\beta)})$

Decompose $\ell = (\ell_0, \ldots, \ell_{\kappa-1})$ with $\ell_j \in [0, \beta-1]$

/ Line 3: Encode column index as a block-wise one-hot vector in base β.

$\mathbf{b} := (\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})_j \in \{0,1\}^{\kappa\beta}$

$\mathbf{b}_1 := (\delta_{\ell_j,1}, \ldots, \delta_{\ell_j,\beta-1})_j \in \{0,1\}^{\kappa(\beta-1)}$

$(a_{j,i})_{j,\,i\in[1,\beta-1]} := \mathsf{Expand}(\rho, \mathsf{a}|\mathsf{ctr}; D_{\phi_a B_a}^{\kappa(\beta-1)})$

Set $a_{j,0} := -\sum_{i=1}^{\beta-1} a_{j,i}\ \forall j$

$\mathbf{a} := (a_{j,0}, \ldots, a_{j,\beta-1})_j,\ \mathbf{d} := (-(a_{j,i})^2)_{j,i},\ \mathbf{c} := \mathbf{a} \circ (\mathbf{1} - 2\mathbf{b})$

$\mathbf{r}_b := \mathsf{Expand}(\rho, \mathsf{rb}|\mathsf{ctr}; S_1^{\hat{n}+\hat{k}})$

$\mathbf{r}_a := \mathsf{Expand}(\rho, \mathsf{ra}|\mathsf{ctr}; D_{\phi_b B_b}^{\hat{n}+\hat{k}})$

$B_{\mathsf{bin}} := \mathbf{G}\left(\mathbf{r}_b, \mathbf{b}, \mathbf{c}\right)^{\top} \in \mathcal{R}_{\hat{q}}^{\hat{n}}$

Decompose $B_{\mathsf{bin}} := 2^{K_B}B_{\mathsf{bin}}^{(1)} + B_{\mathsf{bin}}^{(0)}$ for 2^{K_B}

/ K_B chosen experimentally.

$A_{\mathsf{bin}} := \mathbf{G}\left(\mathbf{r}_a, \mathbf{a}, \mathbf{d}\right)^{\top} \in \mathcal{R}_{\hat{q}}^{\hat{n}}$

Decompose $A_{\mathsf{bin}} := 2^{K_A}A_{\mathsf{bin}}^{(1)} + A_{\mathsf{bin}}^{(0)}$ for $2^{K_A} \approx \hat{n}d\,w\,2^{K_B}$

$x := H(\mu, A_{\mathsf{bin}}^{(1)}, B_{\mathsf{bin}}^{(1)}, \tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK})$.

$\mathbf{z}_b := \mathbf{r}_a + x\mathbf{r}_b$

if $\mathsf{RejOp}(\mathbf{z}_b, x\mathbf{r}_b, \phi_b, B_b)$ **then Restart**

$f_{j,i} := x\,\delta_{\ell_j,i} + a_{j,i}\ \forall j,\ i \in [0, \beta-1]$

$\mathbf{f}_1 := (f_{j,1}, \ldots, f_{j,\beta-1})_j \in \mathcal{R}^{\kappa(\beta-1)}$

if $\mathsf{Rej}\left(\mathbf{f}_1,\ x\mathbf{b}_1,\ \phi_a,\ B_a\right)$ **then** Restart

if $\|\mathbf{f}_1\|_\infty > B_{f_1}\ \vee\ \|\mathbf{f}_0\|_\infty > B_{f_0}$ **then** Restart

$g_{j,i} := f_{j,i}(x - f_{j,i})\ \forall j,\ i \in [0, \beta-1]$

$\mathbf{g}_0 := (g_{0,0}, \ldots, g_{\kappa-1,0}),\ \mathbf{g}_1 := (g_{j,i})_{j\in[0,\kappa-1],\,i\in[1,\beta-1]}$

if $\|\mathbf{g}_0\|_\infty > B_{g_0}\ \vee\ \|\mathbf{g}_1\|_\infty > B_{g_1}$ **then** Restart

$\hat{A}_{\mathsf{bin}} := \mathbf{G}(\mathbf{z}_b, \mathbf{f}, \mathbf{g})^{\top} - x2^{K_B}B_{\mathsf{bin}}^{(1)}$

/ Line 25: Reconstruct compressed commitment.

Decompose $\hat{A}_{\mathsf{bin}} := 2^{K_A}\hat{A}_{\mathsf{bin}}^{(1)} + \hat{A}_{\mathsf{bin}}^{(0)}$

if $\|\hat{A}_{\mathsf{bin}}^{(0)}\|_\infty > 2^{K_A-1} - w2^{K_B-1}$ **then** Restart

/ Line 27: Check dropped bits remain within range.

return $\pi := (B_{\mathsf{bin}}^{(1)},\ (\tilde{\mathbf{w}}_1^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}),\ x,\ \mathbf{f}_1,\ \mathbf{z}_b)$

/ for $\kappa = 1$, no $\tilde{\mathbf{w}}_j^{(1)}$ terms are sent

Figure 5: Binary selection proof. Note that unless specified, $u \in [0, T-1], i \in [0, N-1]$ and $j \in [0, \kappa-1]$.

Binary selection proof [Fig. 5]. $\mathsf{Sign}_{\mathsf{bin}}(\mathsf{pp}, \ell, \mu, (\tilde{\mathbf{w}}_j^{(1)})_j, \mathsf{PK}; \rho)$ is deterministic once the session seed ρ is fixed. It decomposes the hidden column index as $\ell = (\ell_0, \ldots, \ell_{\kappa-1})$ in base β, and forms the block-wise one-hot vector $\mathbf{b} \in \{0,1\}^{\kappa\beta}$, where block j contains a single 1 at position ℓ_j. Using the same Gaussian masks $(a_{j,i})$ expanded from ρ as in Sign_1, it commits to $\mathbf{b}$ and the derived values $\mathbf{c}$ and $\mathbf{d}$ through the public matrix $\mathbf{G}$, which is deterministically expanded from the public seed ρ_{pp}. Concretely, it computes

$$B_{\mathsf{bin}} := \mathbf{G}(\mathbf{r}_b, \mathbf{b}, \mathbf{c})^\top, \qquad A_{\mathsf{bin}} := \mathbf{G}(\mathbf{r}_a, \mathbf{a}, \mathbf{d})^\top,$$

and then writes the coefficient-wise centered decompositions

$$B_{\mathsf{bin}} = 2^{K_B} B_{\mathsf{bin}}^{(1)} + B_{\mathsf{bin}}^{(0)}, \qquad A_{\mathsf{bin}} = 2^{K_A} A_{\mathsf{bin}}^{(1)} + A_{\mathsf{bin}}^{(0)},$$

using the Bai–Galbraith compression technique [2].
It next derives the Fiat–Shamir challenge

$$x := H(\mu, A_{\mathsf{bin}}^{(1)}, B_{\mathsf{bin}}^{(1)}, \tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK}),$$

forms the masked responses, including the coefficients $(f_{j,i})$ encoding $x\mathbf{b}+\mathbf{a}$, and enforces the target response distribution by rejection sampling together with explicit norm checks.
The proof output is the transcript $\pi := \big(B_{\mathsf{bin}}^{(1)}, (\tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}), x, \mathbf{f}_1, \mathbf{z}_b\big)$, which is included only once in the final aggregated signature. Let x be the Fiat–Shamir challenge output by $\mathsf{Sign}_{\mathsf{bin}}$. Each signer then computes its main response

$$\mathbf{z}_u := x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j} \in \mathcal{R}^{l+k},$$

and applies rejection sampling to mask the term $x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}$, restarting if the test rejects. Although the vectors $x^j \mathbf{y}_{u,j}$ are fresh local masks rather than long-term secrets, they must be included together with the secret term in the hidden component passed to rejection sampling in order to match the form required by Lemma A.4. Finally, signer u forms the auxiliary response

$$\mathbf{r}_u := -\sum_{j=0}^{\kappa-1} x^j \mathbf{r}_{u,j} \in \mathcal{R}^{l'+k},$$

and outputs its round-two share $\sigma_u := (\pi, \mathbf{z}_u, \mathbf{r}_u)$.

Signature aggregation [Fig. 6]. At this stage, for each signer $u \in [0, T-1]$ we have a round 2 share $\sigma_u := (\pi, \mathbf{z}_u, \mathbf{r}_u)$ output by Sign_2, where the binary transcript π is common to all honest signers and the responses satisfy $\mathbf{z}_u \in \mathcal{R}^{l+k}$ and $\mathbf{r}_u \in \mathcal{R}^{l'+k}$. The aggregator first splits each $\mathbf{z}_u$ as $(\mathbf{z}_u'^\top, \mathbf{z}_u''^\top) := \mathbf{z}_u^\top$, with $\mathbf{z}_u' \in \mathcal{R}^l$ and $\mathbf{z}_u'' \in \mathcal{R}^k$, and similarly splits each $\mathbf{r}_u$ as $(\mathbf{r}_u'^\top, \mathbf{r}_u''^\top) := \mathbf{r}_u^\top$, with $\mathbf{r}_u' \in \mathcal{R}^{l'}$ and $\mathbf{r}_u'' \in \mathcal{R}^k$. It then forms the aggregated responses by summing component-wise,

$$\tilde{\mathbf{z}} := \sum_{u=0}^{T-1} \mathbf{z}_u', \qquad \tilde{\mathbf{r}} := \sum_{u=0}^{T-1} \mathbf{r}_u', \qquad \tilde{\mathbf{e}} := \sum_{u=0}^{T-1} (\mathbf{z}_u'' + \mathbf{r}_u'').$$

Finally, it outputs the structured threshold ring signature $\tilde{\sigma} := (\pi, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}})$, where π is the shared $\mathsf{Sign}_{\mathsf{bin}}$ transcript and $(\tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}})$ are the aggregated responses.

$\mathsf{SAgg}(\{\sigma_0, \ldots, \sigma_{T-1}\})$

Require: $\sigma_u = (\pi_u, \mathbf{z}_u, \mathbf{r}_u)$
where $\pi_u := (B_{\mathsf{bin},u}^{(1)}, (\tilde{\mathbf{w}}_{1,u}^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1,u}^{(1)}), x_u, \mathbf{f}_{1,u}, \mathbf{z}_{b,u})$
if $\exists u \in [T]$:
  $\Big(x_u \neq x_0 \vee B_{\mathsf{bin},u}^{(1)} \neq B_{\mathsf{bin},0}^{(1)} \vee \mathbf{f}_{1,u} \neq \mathbf{f}_{1,0} \vee \mathbf{z}_{b,u} \neq \mathbf{z}_{b,0}$
  $\vee\ (\tilde{\mathbf{w}}_{1,u}^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1,u}^{(1)}) \neq (\tilde{\mathbf{w}}_{1,0}^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1,0}^{(1)})\Big)$
  then return 0
  / Check signers produce same binary transcript before aggregating responses.
for $u = 0, \ldots, T-1$ **do**
  $(\mathbf{z}_u'^\top, \mathbf{z}_u''^\top) := \mathbf{z}_u^\top, \qquad (\mathbf{r}_u'^\top, \mathbf{r}_u''^\top) := \mathbf{r}_u^\top$
endfor
$\tilde{\mathbf{z}} := \sum_{u=0}^{T-1} \mathbf{z}_u' \in \mathcal{R}^l, \ \tilde{\mathbf{r}} := \sum_{u=0}^{T-1} \mathbf{r}_u' \in \mathcal{R}^{l'}, \ \tilde{\mathbf{e}} = \sum_{u=0}^{T-1} (\mathbf{z}_u'' + \mathbf{r}_u'') \in \mathcal{R}^k$
return $\tilde{\sigma} := (\pi_0, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}})$

Figure 6: Signature aggregation

Verification [Fig. 7]. Given the public parameters $\mathsf{pp} = \rho_{\mathsf{pp}}$, the message μ, the public-key table PK, and a purported signature $\tilde{\sigma} = (\pi, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}})$, the verifier first reconstructs the public matrices $\mathbf{A} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k\times l})$ and $\mathbf{G} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{G}; \mathcal{R}_{\hat{q}}^{\hat{n}\times(\hat{n}+\hat{k}+2\kappa\beta)})$. It then recomputes the commitment matrix $\mathbf{B} := H_{\mathsf{com}}(\mathsf{PK}, \mu) \in \mathcal{R}_q^{k\times l'}$, parses π to obtain $B_{\mathsf{bin}}^{(1)}$, the values $(\tilde{\mathbf{w}}_j^{(1)})_{j\in[1,\kappa-1]}$, the Fiat–Shamir challenge x, and the auxiliary values needed to reconstruct the selector polynomials from the $f_{j,i}$'s. From these it reconstructs $\hat{A}_{\mathsf{bin}}^{(1)}$ exactly as in Fig. 7. The verifier accepts if and only if the following hold:
(i) the norm-bound checks on $\mathbf{f}_1, \mathbf{z}_b, \mathbf{g}_0, \mathbf{g}_1, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}$, and $\tilde{\mathbf{e}}$ are all satisfied;
(ii) writing $(\tilde{\mathbf{t}}_0, \ldots, \tilde{\mathbf{t}}_{N-1}) := \mathsf{KAgg}(\mathsf{PK})$, the verifier computes

$$\widehat{\mathbf{w}}_0 := \sum_{i=0}^{N-1} \Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big) \tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) - \sum_{j=1}^{\kappa-1} x^j \, 2^{K_{w,j}} \tilde{\mathbf{w}}_j^{(1)},$$

decomposes $\widehat{\mathbf{w}}_0 = 2^{K_{w,0}} \widehat{\mathbf{w}}_0^{(1)} + \widehat{\mathbf{w}}_0^{(0)}$ and checks that the Fiat–Shamir hash recomputes the same challenge, namely $x' := H(\mu, \hat{A}_{\mathsf{bin}}^{(1)}, B_{\mathsf{bin}}^{(1)}, \widehat{\mathbf{w}}_0^{(1)}, \tilde{\mathbf{w}}_1^{(1)} \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK})$ satisfies $x = x'$.

4.1 Efficiency and Sizes

We now estimate the sizes of core components of LoTRS.

Encoding model. An element of $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ is represented by its d coefficients in $[0, q-1]$, requiring $b_q := \lceil \log_2 q \rceil$ bits per coefficient. Thus a ring element costs $d \cdot b_q$ bits and a vector in $\mathcal{R}_q^m$ costs $m \cdot d \cdot b_q$ bits. An analogous analysis applies to $\mathcal{R}_{\hat{q}}$. For small-norm values with an explicit ℓ_∞ bound $\|\cdot\|_\infty \le B$, we can use signed encoding in $[-B, B]$, requiring $b_B := \lceil \log_2(2B+1) \rceil$ bits per coefficient. For a (discrete) Gaussian sample from D_σ, we can use an entropic bound of $\log_2(4.13\sigma)$ bits as detailed in [15].

$\mathsf{Vf}(\mathsf{pp}, \mu, \tilde{\sigma}, \mathsf{PK})$

1 : **Require:** $\tilde{\sigma} = \big(\pi := (B^{(1)}_{\mathsf{bin}}, (\tilde{\mathbf{w}}^{(1)}_1, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}), x, \mathbf{f}_1, \mathbf{z}_b), \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}\big)$.
2 : **Require:** $\mathsf{PK} := \big(\mathsf{PK}_i := (\mathbf{t}_{u,i})_u\big)_i$ with $N = \beta^\kappa$
3 : $\mathbf{A} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k\times l})$
4 : $\mathbf{G} := \mathsf{Expand}\left(\rho_{\mathsf{pp}}, \mathsf{G}; \mathcal{R}_{\hat{q}}^{\hat{n}\times(\hat{n}+\hat{k}+2\kappa\beta)}\right)$
5 : **if** $\big(\|\mathbf{z}_b\|_\infty > 6\phi_b B_b\big)$ **then return** 0
6 : **if** $(\|\tilde{\mathbf{z}}\| > B_{\tilde{z}} \vee \|\tilde{\mathbf{r}}\| > B_{\tilde{r}} \vee \|\tilde{\mathbf{e}}\| > B_{\tilde{e}})$ **then return** 0
7 : Parse $\mathbf{f}_1 := (f_{j,1}, \ldots, f_{j,\beta-1})_j$
8 : Set $f_{j,0} := x - \sum_{i=1}^{\beta-1} f_{j,i},\ \forall j$
9 : $\mathbf{f} := (f_{j,0}, \ldots, f_{j,\beta-1})_{j\in[0,\kappa-1]}$
10 : **if** $\big(\|\mathbf{f}_1\|_\infty > B_{f_1} \vee \|\mathbf{f}_0\|_\infty > B_{f_0}\big)$ **then return** 0
11 : Define $g_{j,i} := f_{j,i}(x - f_{j,i})\ \forall\ j \in [0,\kappa-1],\ i \in [0,\beta-1]$
12 : Let $\mathbf{g}_0 := (g_{0,0}, \ldots, g_{\kappa-1,0})$ and $\mathbf{g}_1 := (g_{j,i})_{j\in[0,\kappa-1],\, i\in[1,\beta-1]}$
13 : **if** $\|\mathbf{g}_0\|_\infty > B_{g_0}$ **or** $\|\mathbf{g}_1\|_\infty > B_{g_1}$ **then return** 0
14 : $\mathbf{g} := (g_{j,i})_{j\in[0,\kappa-1],\, i\in[0,\beta-1]}$
15 : $\hat{A}_{\mathsf{bin}} := \mathbf{G}\big(\mathbf{z}_b, \mathbf{f}, \mathbf{g}\big)^\top - x2^{K_B} B^{(1)}_{\mathsf{bin}} \in \mathcal{R}_{\hat{q}}^{\hat{n}}$
16 : Decompose $\hat{A}_{\mathsf{bin}} := 2^{K_A}\hat{A}^{(1)}_{\mathsf{bin}} + \hat{A}^{(0)}_{\mathsf{bin}}$
17 : **if** $\|\hat{A}^{(0)}_{\mathsf{bin}}\|_\infty > 2^{K_A-1} - w2^{K_B-1}$ **then return** 0
/ Line 7– 17: Reconstruct and check binary proof data.
18 : $\mathbf{B} := H_{\mathsf{com}}(\mathsf{PK}, \mu) \in \mathcal{R}_q^{k\times l'}$
19 : $(\tilde{\mathbf{t}}_0, \ldots, \tilde{\mathbf{t}}_{N-1}) := \mathsf{KAgg}(\mathsf{PK})$
20 : $\widehat{\mathbf{w}}_0 := \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big)\tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) - \sum_{j=1}^{\kappa-1} x^j\, 2^{K_{w,j}}\tilde{\mathbf{w}}^{(1)}_j$
21 : Decompose $\widehat{\mathbf{w}}_0 := 2^{K_{w,0}}\widehat{\mathbf{w}}^{(1)}_0 + \widehat{\mathbf{w}}^{(0)}_0$
22 : **if** $x \neq H\big(\mu, \hat{A}^{(1)}_{\mathsf{bin}}, B^{(1)}_{\mathsf{bin}}, \widehat{\mathbf{w}}^{(1)}_0, \tilde{\mathbf{w}}^{(1)}_1, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}, \mathsf{PK}\big)$ **then** return 0
23 : **return** 1.

Figure 7: Verifier. Note that unless specified, $u \in [0, T-1], i \in [0, N-1]$ **and** $j \in [0, \kappa-1]$

Public parameters. The public parameters are $\mathsf{pp} := \rho_{\mathsf{pp}}$. The matrix $\mathbf{A}$ is reconstructed as $\mathbf{A} := \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k\times l})$, and $\bar{\mathbf{A}} := [\mathbf{A} \mid \mathbf{I}]$ is derived from it as needed.

Key sizes. A public key is $\mathbf{t} = \bar{\mathbf{A}}\mathbf{s} \in \mathcal{R}_q^k$, hence $|\mathsf{pk}| = kdb_q$ bits. A secret key $\mathbf{s} \in S_\eta^{l+k}$ can be stored as a compact key generation seed value or with $b_\eta := \lceil \log_2(2\eta+1) \rceil$ bits per coefficient, hence $|\mathsf{sk}| = (l+k)db_\eta$ bits. The ephemeral session seed ρ is established during the session initialization for a single signing session and is not part of the long-term secret key; hence it is not included in $|\mathsf{sk}|$.

Signature size. A LoTRS signature is $\tilde{\sigma} = (\pi, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}})$ where from $\mathsf{Sign}_{\mathsf{bin}}$, we have $\pi = (B^{(1)}_{\mathsf{bin}}, (\tilde{\mathbf{w}}^{(1)}_1, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}), x, \mathbf{f}_1, \mathbf{z}_b)$. Therefore,

$$|\tilde{\sigma}| = |B^{(1)}_{\mathsf{bin}}| + \sum_{j=1}^{\kappa-1} |\tilde{\mathbf{w}}^{(1)}_j| + |x| + |\mathbf{f}_1| + |\mathbf{z}_b| + |\tilde{\mathbf{z}}| + |\tilde{\mathbf{r}}| + |\tilde{\mathbf{e}}|.$$

Using the encoding model above yields $\sum_{j=1}^{\kappa-1} |\tilde{\mathbf{w}}^{(1)}_j| = \sum_{j=1}^{\kappa-1} kd(b_q - K_{w,j})$, $|\tilde{\mathbf{z}}| = ld\log_2(4.13\sigma_z)$, $|\tilde{\mathbf{r}}| = l'd\log_2(4.13\sigma_r)$, $|\tilde{\mathbf{e}}| = kd\log_2(4.13\sigma_e)$, $|\mathbf{f}_1| = \kappa(\beta-1)d\log_2(4.13\sigma_f)$, $|\mathbf{z}_b| = (\hat{n}+\hat{k})d\log_2(4.13\sigma_{z_b})$, $|B^{(1)}_{\mathsf{bin}}| = \hat{n}d(b_{\hat{q}} - K_B)$, with $|x| \approx 128$. Note that x can be stored as an intermediate 128-bit hash value which is expanded into a challenge distribution sample when required.

Interactive communication. Sign_1 transmits $\mathsf{Com}_u = (\mathbf{w}_{u,j})_{j\in[0,\kappa-1]}$ from each signer, hence $|\mathsf{Com}_u| = \kappa kdb_q$ bits and total round 1 broadcast is $T\kappa kdb_q$ bits. Sign_2 transmits a shared π once and per-signer responses $(\mathbf{z}_u, \mathbf{r}_u)$, so total round 2 communication is approximately $|\pi| + T(|\mathbf{z}_u| + |\mathbf{r}_u|) \le |\pi| + Tdb_q(l + l' + 2k)$ bits.
The above communication counts are per signing attempt. To obtain expected communication, they should be multiplied by the expected number of attempts from Theorem 5.1.

Asymptotic communication size. Since we rely on the one-out-of-many proof of [16, 18], we get a polylog(N) dependence for the signature size (more concretely about $\log^{1.6} N$ complexity; see [16, App. F.4]). Also, since we compress the whole PK table into a single row, the dependence on T is minimal. In particular, since we sum T terms (for aggregation) each following a Gaussian distribution, their bit length depends on $\log(\sqrt{T})$. Overall, we get polylog(N, T) signature size.

Implementation Encoding Sizes. The implementation benchmarks report the actual emitted wire size in KiB, while Table 3 reports the analytic KB estimate used during parameter selection; small differences come from Rice-code sample variation, byte alignment, and the decimal-vs-binary unit convention.

5 Security of LoTRS

In this section, we first prove the correctness of LoTRS and list all required underlying lattice assumptions required for security analysis and proofs. We finally provide all the security properties of LoTRS along with their detailed security reductions.

5.1 Assumptions and correctness

To ensure the security of our LoTRS, we make the following assumptions regarding the parameters in Tables 2 and 5.

(1) $q, \hat{q} \equiv 5 \mod 8$ with $\|x - x'\|_\infty < \sqrt{\hat{q}/2}, \sqrt{q/2}$ (Lemma A.1)
(2) $\hat{q} > \max\{d(2 + 12\phi_a B_a)^2, 2N^2\}$ (Lemma D.1)
(3) $\sigma_s := \phi_s B_s > (2d/\sqrt{2\pi}) \cdot q^{k/(l+k)+2/(d(l+k))}$ (Lemma A.8)
(4) $\sigma_{s'} := \phi_{s'} B_{s'} > (2d/\sqrt{2\pi}) \cdot q^{k/(l'+k)+2/(d(l'+k))}$ (Lemma A.8)
(5) $\mathsf{MSIS}^\infty_{\hat{q},\hat{n},\,2\kappa\beta+\hat{n},\,2\gamma^*_{\mathsf{com}}}$ is hard (Lemma 5.5), where

$$\gamma^*_{\mathsf{com}} := \max\Big(B_{g_0},\ B_{g_1},\ 6\phi_a B_a,\ 6\phi_b B_b,\ 2^{K_B-1},\ 2^{K_A-1},\ 2^{K_A} - 2w2^{K_B-1},\ 2w\big(2^{K_A-1} - w2^{K_B-1}\big)\Big).$$

For parameter setting:
$\mathsf{AMSIS}_{\hat{q},\hat{n};\,(m_{f_1},m_{f_0},m_{g_1},m_{g_0},m_{\hat{z}_b},m_{\mathrm{BG}});\,(B_{f_1},B_{f_0},B_{g_1},B_{g_0},B_{\hat{z}_b},B_{\mathrm{BG}})}$
is hard, where $m_{f_1} = \kappa(\beta-1), m_{f_0} = \kappa, m_{g_1} = \kappa(\beta-1), m_{g_0} = \kappa, m_{\hat{z}_b} = \hat{n} + \hat{k}, m_{\mathrm{BG}} = 4\hat{n}$, and $B_{f_1} := 1 + \tau_{f_1}\phi_a\sqrt{\kappa w}$, $B_{f_0} := 1 + \tau_{f_0}\sqrt{\beta-1}\,\phi_a\sqrt{\kappa w}$,

$B_{g_1} := wB_{f_1} + dB_{f_1}^2$, $B_{g_0} := wB_{f_0} + dB_{f_0}^2$, $B_{\hat{z}_b} := 6\phi_b\sqrt{d(\hat{n}+\hat{k})}\,w$,
$B_{\mathrm{BG}} := \max\Big(2^{K_b-1}, 2^{K_a-1}, 2^{K_a} - 2w2^{K_b-1}, 2w\big(2^{K_a-1} - w2^{K_b-1}\big)\Big)$.

(6) $\mathsf{MLWE}_{\hat{q},\hat{n},\hat{k},1}$ is hard (Theorem 5.7).
(7) $\mathsf{MSIS}^{\infty}_{q,k,\,1+l+l'+2k,\,\beta^{\infty}_{\mathsf{SIS}}}$ is hard (Lemma 5.6), where

$$\beta^{\infty}_{\mathsf{SIS}} := \max\Big(2B_{\mathrm{det}},\ 2(\kappa+1)B_\Gamma B_{\hat{z}},\ 2(\kappa+1)B_\Gamma B_{\hat{r}},\ 2(\kappa+1)B_\Gamma B_{\hat{e}},\ 2(\kappa+1)B_\Gamma B_{\eta,w}\Big).$$

For parameter setting:
$\mathsf{AMSIS}_{q,k;\,(m_\alpha,m_{\hat{z}},m_{\hat{r}},m_{\hat{e}},m_{\hat{\eta}_w});\,(B_\alpha,B_{\hat{z}},B_{\hat{r}},B_{\hat{e}},B_{\hat{\eta}_w})}$ is hard, where $m_\alpha = 1, m_{\hat{z}} = l, m_{\hat{r}} = l', m_{\hat{e}} = k, m_{\hat{\eta}_w} = k$, and $B_\alpha := 2B_{\mathrm{det}}$, $B_{\hat{z}} := 2(\kappa+1)B_\Gamma B_{\hat{z}}$, $B_{\hat{r}} := 2(\kappa+1)B_\Gamma B_{\hat{r}}$, $B_{\hat{e}} := 2(\kappa+1)B_\Gamma B_{\hat{e}}$, $B_{\widehat{\eta}_w} := 2(\kappa+1)B_\Gamma B_{\eta,w}$.
(8) $\mathsf{MLWE}_{q,k,l,\eta}$ is hard (Lemma 5.6).
(9) $\mathsf{MLWE}_{q,k,l'-1,\eta'}$ is hard (Lemma 5.4).

Theorem 5.1 (LoTRS Completeness). LoTRS *has correctness error at most*

$$1 - \left(\frac{1}{\mu(\phi)}\right)^T + \left(1 - \frac{1}{\mu(\phi_a)}\right) + \left(1 - \frac{1}{\mu(\phi_b)}\right) + (1 - p_{\mathrm{BG}}) + (1 - p_w) + \varepsilon_{\mathrm{tot}} + \epsilon_{\mathrm{agg}}.$$

when defined with the parameters in Tables 2 and 5. Under the same independence heuristic, the expected number of signing attempts is approximately $\mu(\phi)^T \cdot \mu(\phi_a) \cdot \mu(\phi_b) \cdot \mu_{\mathrm{BG}} \cdot \mu_w \cdot \mu_{fg}$, *where* $\mu_{\mathrm{BG}} := p_{\mathrm{BG}}^{-1}$, $\mu_w := p_w^{-1}$, $\mu_{fg} := \frac{1}{1-\varepsilon_{\mathrm{tot}}}$.

We prove LoTRS is complete in Appendix C.

5.2 LoTRS Security Model

Our proof in the random oracle model (ROM) extends on the proof of DualMS [8] and we show how to adapt the unforgeability of a multisignature to the threshold ring signature setting.

Assumptions on random-oracle queries. We assume w.l.o.g. the adversary $\mathcal{A}$ makes its random-oracle and signing queries in the following canonical order:

- Any $\mathcal{A}$ queries $H_{\mathrm{com}}(\mathsf{PK}, \mu)$ before any query $H(\mu, (\cdot), \mathsf{PK})$.
- Any $\mathcal{A}$ queries $H_{\mathrm{agg}}(\mathsf{PK}, u)$ for every $u \in \{0, \ldots T-1\}$ before it queries $\mathsf{OSign}_1(\mu, \ell, \mathsf{PK})$.
- In any signing session, $\mathcal{A}$ queries $H(\mu, (\cdot), \tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK})$ before it queries $\mathsf{OSign}_2(\mathsf{st}_u, \{(\mathbf{w}_{u',j})_j\}_{u' \in [0,T-1]\setminus\{u\}})$.
- $\mathcal{A}$ makes all hash queries needed to verify its eventual forgery before outputting it.

These assumptions incur only a standard, inessential loss: from any adversary making at most Q_k key generating queries, Q_h queries to each random oracle and initiating Q_s signing sessions, we can build a "random-oracle middleman" adversary $\mathcal{A}'$ that satisfies the above order, succeeds with the same probability, and makes at most $2Q_h + n(Q_s+1)$ queries to each random oracle.

Selective Security. We establish the unforgeability of LoTRS via a sequence of reductions. First, we reduce unforgeability under chosen-message attacks (UF-CMA) to its selective variant (sel-UF-CMA) in the random oracle model. Next, we reduce sel-UF-CMA to selective unforgeability under key-only attacks (sel-UF-KOA), together with the hardness of MLWE problem (Definition A.5). Finally, we reduce sel-UF-KOA to the hardness of MSIS (Definition A.6) and MLWE. Note that the security games for (sel-)UF-KOA are exactly the same as those for (sel-)UF-CMA, but without access to signing oracles (see Fig. 9).

For ease of notation $\mathsf{Exp}^{(\mathsf{sel})\text{-}\mathsf{UF}\text{-}\mathsf{CMA}}_{\mathsf{LoTRS},\mathcal{A}}(\lambda) :=$ (sel)-UF-CMA of LoTRS. In the selective security setting, the adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ operates in two stages as shown in Fig. 9. As the name suggests, $\mathcal{A}$ "selects" in advance a target index i^* on which it intends to forge a signature. Concretely for LoTRS, in the first stage, $\mathcal{A}_0$ outputs an index i^* corresponding to a future query to the random oracle H_{com}, without making any oracle queries itself. In other words, the target message-table pair (μ^*, PK^*) is defined to be the adversary's i^*-th fresh query to H_{com}.

In the second stage, $\mathcal{A}_1$ is given access to the random oracles and the signing oracles. It outputs a purported forgery consisting of a public key table instance PK^* and a signature σ^*, but no explicit message. $\mathcal{A}$ wins if (i) all the public keys in PK^* are distinct, (ii) (μ^*, PK^*) was not queried to the signing oracle, (iii) for every admissible signer set $S \in \mathcal{F}_{\Psi^*}(\mathsf{PK}^*)$, fewer than T keys in S belong to $P_{\mathrm{corr}} \cup P_{\mathrm{sig}}^{\mu^*}$, and (iv) the verifier check $\mathsf{Vf}(\mu^*, \sigma^*, \mathsf{PK}^*)$ succeeds. The proof of the following Lemma can be found in Appendix D.1.

Lemma 5.2 (UF-CMA to sel-UF-CMA [8]). *Let* $\mathcal{A}$ *be a* τ*-time adversary against* UF-CMA *security of* LoTRS *that makes at most* Q_h *queries to each random oracle. Then there exists an adversary* $\mathcal{B}$ *s.t.*

$$\mathsf{Adv}^{\mathsf{UF}\text{-}\mathsf{CMA}}_{\mathsf{LoTRS}}(\mathcal{A}) \ \le\ Q_h \cdot \mathsf{Adv}^{\mathsf{sel}\text{-}\mathsf{UF}\text{-}\mathsf{CMA}}_{\mathsf{LoTRS}}(\mathcal{B}).$$

and the running time of $\mathcal{B}$ *is approximately* τ.

In order to reduce sel-UF-CMA to sel-UF-KOA we need the following lemma. The results turn out to essentially be the same as those in DualMS since we primarily kept the same signature structure. We prove each of the claims separately and analogously to DualMS [8] in Appendix D.2.

Lemma 5.3. *Let* $k, l, l' \in \mathbb{Z}^+$ *be such that* $l + k, l' + k \le \mathrm{poly}(d)$. *Let* Rej *be the rejection sampling algorithm as defined in Fig. 8. Assume conditions 3, 4 from Section 5.1 hold. Let* Trans *and* Sim *be procedures with inputs and outputs as described in Fig. 11. Then with probability* $1 - 2^{\Omega(d)}$ *over the choices of* $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k\times l}$ *and* $\hat{\mathbf{B}} \xleftarrow{\$} \mathcal{R}_q^{k\times(l'-1)}$, *for any* $\mathbf{s} \in S_\eta^{l+k}$, $\mathbf{u} \in S_{\eta'}^{l'-1+k}$ *the following three claims hold:*

Claim 1. *The distributions of* out_1 *in* Trans *and* Sim *are identical.*

Claim 2. *For any* $c \in C'$ *and conditioned on any* out_1, *the following result holds for both* Trans *and* Sim*:*

$$|\Pr[\mathsf{out}_2 \neq \perp \mid \mathsf{out}_1]| \le\ 2^{-100} + 2^{-\Omega(d)}$$

Claim 3. *For any* $c \in C'$ *and conditioned on any* out_1, *the statistical distance between the distributions of* out_2 *in* Trans *and* Sim *is at most* $3 \cdot 2^{-100}/2\mu(\phi) + 2^{-\Omega(d)}$.

5.3 Reduction: sel-UF-CMA ⇒ sel-UF-KOA & MLWE

In this reduction, we work through a series of hybrid games that turns the CMA adversary into a KOA adversary. We argue via a standard hybrid sequence of games G_0, G_1, $\mathrm{G}_{2,0}, \ldots, \mathrm{G}_{2,Q_h}$, and G_3, as in DualMS in Appendix D.3.

Lemma 5.4 (sel-UF-CMA ⇒ sel-UF-KOA & MLWE). *For any τ-time adversary $\mathcal{A}$ against the* sel-UF-CMA *of* LoTRS *that makes at most Q_h queries to each of the random oracles and launches at most Q_s signing sessions, there exist algorithms $\mathcal{B}$ and $\mathcal{D}$ such that*

$$\mathsf{Adv}^{\text{sel-UF-CMA}}_{\text{LoTRS}}(\mathcal{A}) \le \mathsf{Adv}^{\text{sel-UF-KOA}}_{\text{LoTRS}}(\mathcal{B}) + (Q_h - 1)\,\mathsf{Adv}^{\text{MLWE}}_{q,k,l'-1,\eta'}(\mathcal{D}) + Q_s\Big(\frac{3\cdot 2^{-100}}{2\mu(\phi)} + 2^{-\Omega(N)} + \frac{Q_h}{|C|}\Big),$$

where ϕ is specified in Table 2 and running times of $\mathcal{B}$ and $\mathcal{D}$ are essentially τ.

5.4 Reduction: sel-UF-KOA ⇒ MSIS & MLWE

Finally, we can get a reduction to our hard MSIS problem. For the reduction to go through, we use the following selector-extraction lemma. It shows that, from three accepting forks of the underlying LoTRS proof with the same first-round data and distinct challenges, the extractor recovers a one-hot selector. Consequently, when the verifier computes the product over the reconstructed $f_{j,i}$ terms, the degree-κ term selects exactly one aggregated key $\tilde{\mathbf{t}}_\ell$, while all remaining terms are absorbed into lower-degree challenge powers.

Lemma 5.5 (Selector extraction). *Let $N = \beta^\kappa$. Suppose an extractor is given three accepting forks of the $\mathsf{Sign}_{\mathsf{bin}}$ proof with the same first-round data $(A^{(1)}_{\mathsf{bin}}, B^{(1)}_{\mathsf{bin}}, \tilde{\mathbf{w}}^{(1)}_0, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1})$ and with distinct challenges $x, x', x'' \in C$ whose pairwise differences are invertible. Let $\mathbf{f}_1, \mathbf{f}'_1, \mathbf{f}''_1$ be the corresponding selector-response vectors, and let $f_{j,v}, f'_{j,v}, f''_{j,v}$ denote their verifier-side reconstructed coordinates, for $j \in [0, \kappa-1]$ and $v \in [0, \beta-1]$. Then, except with negligible probability, there exist $a_{j,v} \in \mathcal{R}$ and $b_{j,v} \in \{0,1\} \subseteq \mathcal{R}$ such that $f_{j,v} = a_{j,v} + x b_{j,v}$, $f'_{j,v} = a_{j,v} + x' b_{j,v}$, $f''_{j,v} = a_{j,v} + x'' b_{j,v}$, and $\sum_{v=0}^{\beta-1} b_{j,v} = 1$ for every $j \in [0, \kappa-1]$. Consequently, for each j there is a unique $\ell_j \in [0, \beta-1]$ such that $b_{j,\ell_j} = 1$. Let $\ell := \sum_{j=0}^{\kappa-1} \ell_j \beta^j \in [0, N-1]$. Then there exist vector coefficients $\mathbf{C}_0, \ldots, \mathbf{C}_{\kappa-1} \in \mathcal{R}_q^k$ such that, writing $i = \sum_{j=0}^{\kappa-1} i_j \beta^j$, gives*

$$\sum_{i=0}^{N-1} \left(\prod_{j=0}^{\kappa-1} f_{j,i_j}\right) \tilde{\mathbf{t}}_i = x^\kappa \tilde{\mathbf{t}}_\ell + \sum_{m=0}^{\kappa-1} \mathbf{C}_m x^m.$$

The analogous identity holds for the other two forks after replacing $(x, f_{j,v})$ by $(x', f'_{j,v})$ or $(x'', f''_{j,v})$.

The proof follows from the exact three-challenge extraction argument (Theorem 1 of Esgin et al. [16]) for the compressed binary first message using the augmented commitment map $\mathsf{Com}_{\mathsf{G}^*}$; we give the full argument in Appendix D.4.

Lemma 5.6 (sel-UF-KOA ⇒ MSIS & MLWE). *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a $\tau_{\mathcal{A}}$-adversary that wins* sel-UF-KOA *with a non-negligible probability $\epsilon_{\mathcal{A}} := \mathsf{Adv}^{\text{sel-UF-KOA}}_{\text{LoTRS}}(\mathcal{A})$, makes at most Q_h to each random oracle, and at most Q_k key generating queries. Then there exists a $\tau_{\mathcal{D}}$-algorithm $\mathcal{D}$ and a $\tau_{\mathcal{B}}$-algorithm $\mathcal{B}$ with*

$$\frac{1}{4Q_k} \le \epsilon_{\mathcal{B}}, \tau_{\mathcal{B}} \le O\left(\frac{\kappa \tau_{\mathcal{A}}}{\epsilon_{\mathcal{A}} - \epsilon_{\mathcal{D}}}\right) \text{ and } \epsilon_{\mathcal{D}} := \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\eta}(\mathcal{D}), \tau_{\mathcal{D}} \approx \tau_{\mathcal{A}},$$

where $\mathcal{B}$ outputs a non-zero solution to an $\mathsf{MSIS}^\infty_{q,k,\,1+l+l'+2k,\,\beta^\infty_{\mathsf{SIS}}}$ instance with

$$\beta^\infty_{\mathsf{SIS}} := \max\big(2B_{\mathsf{det}},\ \kappa' B_\Gamma B_{\tilde{z}},\ \kappa' B_\Gamma B_{\tilde{r}},\ \kappa' B_\Gamma B_{\tilde{e}},\ \kappa' B_\Gamma B_{\eta,w}\big),\ \kappa' := 2(\kappa+1).$$

We cannot follow the double-forking proof style of DualMS [8] (Appendix A, proof of Lemma 10) since we need to extract from a degree-κ polynomial relation in the Fiat-Shamir challenge x. This requires extracting from $2(\kappa+1)$ accepting transcripts with distinct challenges, using the Tree-Finder algorithm in [11] and a Vandermonde/adjugate "one-shot" technique from [16]. The complete proof is in Appendix D.5.

Finally, we prove anonymity for our scheme. Note that in the anonymity game, the goal is to provide anonymity from adversaries outside the set of signers, unlike unforgeability where we may have a malicious signer.

Theorem 5.7 (Anonymity). *Let $\mathcal{A}$ be any probabilistic polynomial time (PPT) adversary against* LoTRS *making at most Q_h queries to each random oracle and launches at most Q_s signing session. Then*

$$\big|\Pr[\mathsf{Exp}^{\text{anon-0}}_{\text{LoTRS},\mathcal{A}} = 1] - \Pr[\mathsf{Exp}^{\text{anon-1}}_{\text{LoTRS},\mathcal{A}} = 1]\big| \le 2Q_s\left(\frac{2^{-100}}{\mu(\phi_b)} + \frac{2^{-100}}{\mu(\phi_a)} + \mathsf{Adv}^{\mathsf{MLWE}}_{\hat{q},\hat{n},\hat{k},1}(\mathcal{D}') + T\frac{2^{-100}}{\mu(\phi_0)} + T\epsilon(\kappa-1)\right).$$

To prove anonymity, fix an arbitrary PPT adversary $\mathcal{A}$ and condition on any well-formed anonymity challenge $(\mu^*, S_{U_0}, S_{U_1}, \mathsf{PK}^*)$ output by $\mathcal{A}$ in the anonymity experiment. Here S_{U_0}, S_{U_1} are two admissible candidate signer sets, each of size T, whose challenged signers are honest. The proof proceeds through a sequence of hybrid games that gradually transforms a signature generated for S_{U_0} into one generated for S_{U_1}; see Appendix D.6.

6 Concrete Instantiation

For our parameter setting, we need to satisfy conditions in Section 5.1. To find good parameters, we have searched over various values and filtered out those that satisfied all the required conditions. We estimate the hardness of MSIS and MLWE in practice against known attacks by following the methodology described in [14, Section 3.2.4] and also using the Lattice (a.k.a. LWE) estimator [1]. We also leverage the Asymmetric MSIS problem (see Definition A.7, [43]) for Assumptions 5 and 7 using the extended scripts from [18]. As in earlier works [16, 19], we aim for a "root-Hermite factor" of $\delta \approx 1.0045$ when aiming for 128-bit post-quantum security. We present in Table 3 an example parameter setting satisfying the requirements (Further details can be found in Appendix E).

Table 3: Concrete parameters for LoTRS. Sizes for public key pk and signature $\tilde{\sigma}$ are analytic estimates in decimal KB (10^3 bytes) from the parameter-selection scripts; the codec-emitted wire sizes (in binary KiB, 2^{10} bytes) are reported in Table 4 and Appendix F.

N	T	κ	β	d	k	l	l'	$\hat{n}$	$\hat{k}$	$q = \hat{q}$	pk	$\tilde{\sigma}$
100	50	1	100	128	12	5	6	11	8	$2^{38}-107$	7.13	35.06

At this point, our current construction of LoTRS gives very competitive signature sizes for the $N = 100, T = N/2$ setting. In the present version, we also incorporate standard compression techniques for the binary first message and for the aggregated commitment $\tilde{\mathbf{w}}_j$ ($\kappa = 1$ case), which further reduce the transmitted signature size while introducing only the explicit bounded residual terms handled in verification and in the security proof.

6.1 Implementation and Benchmarks

An implementation artifact is available from: https://github.com/lotrs-sig/lotrs. We provide two reference implementations of LoTRS: a Python "golden reference" (`lotrs-py`) for clarity and test-vector generation, and a Rust implementation (`lotrs-rs`) cross-validated byte-for-byte against the Python test vectors. Both cover the full κ=1 pipeline; a two-prime CRT-NTT backend supplies fast negacyclic multiplication at d=128 without altering $q, \hat{q}$. Table 4 reports wall-clock Sign / Verify means over 100 signatures per cell, with KeyGen / KAgg as deterministic setup measurements, on a Ryzen AI 9 HX 370 (Zen 5, 5.1 GHz, Debian Forky `Linux 7.0.4`, `rustc 1.95`, `rayon` multi-threaded); details in Appendix F

Table 4: LoTRS **end-to-end timings (`rayon` multi-threaded, Ryzen AI 9 HX 370). Sign is per accepted signature; $\tilde{\sigma}$ in KiB.**

N	T	Sign	Verify	KAgg	$\tilde{\sigma}$
32	4	120 ms	26 ms	10 ms	23.9
32	16	149 ms	43 ms	32 ms	25.0
100	5	417 ms	60 ms	23 ms	34.0
100	50	789 ms	250 ms	198 ms	35.8

At the production setting N=100, T=50, signing completes in ≈ 0.8 s, verification in ≈ 0.25 s, and the canonical Golomb–Rice serialized signature is 35.8 KiB; the deployment-scale cell N=32, T=16 yields 149 ms / 43 ms / 25.0 KiB. Empirical attempt counts run 5–7 versus the heuristic $\mu_{\text{tot}} \approx 3$, reflecting an additional $\tilde{\mathbf{w}}_0^{(1)}$-stability restart used by the κ=1 commitment-compression path. A full breakdown across all (N, T) cells, the T=1 ring-signature edge, and the N=1 DualMS edge is available via the artifact.

Acknowledgments

This work is partly supported by Australian Research Council Discovery Grants DP250100229 and DP260100245.

References

[1] Martin R. Albrecht, Rachel Player, and Sam Scott. 2015. On the concrete hardness of Learning with Errors. *J. Math. Cryptol.* 9, 3 (2015), 169–203.

[2] Shi Bai and Steven D Galbraith. 2014. An improved compression technique for signatures based on learning with errors. In *Cryptographers' Track at the RSA Conference*. Springer, 28–47.

[3] Slim Bettaieb and Julien Schrek. 2013. Improved lattice-based threshold ring signature scheme. In *International Workshop on Post-Quantum Cryptography*. Springer, 34–51.

[4] Ward Beullens and Gregor Seiler. 2023. LaBRADOR: compact proofs for R1CS from module-SIS. In *Annual International Cryptology Conference*. Springer, 518–548.

[5] BitGo. [n. d.]. Wallets Overview. https://developers.bitgo.com/docs/wallets-overview. States that BitGo MPC wallets use threshold signature schemes and a 2-of-3 configuration. Accessed: 2026-04-29.

[6] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. 2002. Threshold ring signatures and applications to ad-hoc groups. In *Annual International Cryptology Conference*. Springer, 465–480.

[7] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. 2010. A lattice-based threshold ring signature scheme. In *International Conference on Cryptology and Information Security in Latin America*. Springer, 255–272.

[8] Yanbo Chen. 2023. DualMS: Efficient lattice-based two-round multi-signature with trapdoor-free simulation. In *Advances in Cryptology – CRYPTO 2023*. Springer Nature Switzerland, 716–747.

[9] James Hsin-Yu Chiang, Ivan Damgård, William R Duro, Sunniva Engan, Sebastian Kolby, and Peter Scholl. 2025. Post-quantum threshold ring signature applications from VOLE-in-the-head. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*. 4664–4678.

[10] Léonard Dallot and Damien Vergnaud. 2009. Provably secure code-based threshold ring signatures. In *IMA International Conference on Cryptography and Coding*. Springer, 222–235.

[11] Rafaël Del Pino, Vadim Lyubashevsky, and Gregor Seiler. 2019. Short discrete log proofs for FHE and ring-LWE ciphertexts. In *IACR International Workshop on Public Key Cryptography*. Springer, 344–373.

[12] Julien Devevey, Alain Passelègue, and Damien Stehlé. 2023. G+ G: A fiat-shamir lattice signature based on convolved gaussians. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 37–64.

[13] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2018. Crystals-dilithium: A lattice-based digital signature scheme. *IACR transactions on cryptographic hardware and embedded systems* (2018), 238–268.

[14] Muhammed F. Esgin. 2020. *Practice-Oriented Techniques in Lattice-Based Cryptography*. Ph. D. Dissertation. Monash University. doi:10.26180/5eb8f525b3562 https://bridges.monash.edu/articles/Practice-Oriented_Techniques_in_Lattice-Based_Cryptography/12279728.

[15] Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. 2023. Efficient Hybrid Exact/Relaxed Lattice Proofs and Applications to Rounding and VRFs. In *CRYPTO (5) (Lecture Notes in Computer Science, Vol. 14085)*. Springer, 484–517.

[16] Muhammed F Esgin, Ron Steinfeld, Joseph K Liu, and Dongxi Liu. 2019. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *Annual International Cryptology Conference*. Springer, 115–146.

[17] Muhammed F Esgin, Ron Steinfeld, Amin Sakzad, Joseph K Liu, and Dongxi Liu. 2019. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *International Conference on Applied Cryptography and Network Security*. Springer, 67–88.

[18] Muhammed F Esgin, Ron Steinfeld, and Raymond K Zhao. 2022. MatRiCT+: More efficient post-quantum private blockchain payments. In *2022 ieee symposium on security and privacy (sp)*. IEEE, 1281–1298.

[19] Muhammed F Esgin, Raymond K Zhao, Ron Steinfeld, Joseph K Liu, and Dongxi Liu. 2019. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 567–584.

[20] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. 2018. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submission to the NIST's post-quantum cryptography standardization process* 36, 5 (2018), 1–75.

[21] Frostsnap. [n. d.]. Documentation. https://frostsnap.com/docs/. States that Frostsnap uses the FROST protocol and describes a 2-of-3 threshold setup. Accessed: 2026-04-29.

[22] Abida Haque, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. 2022. Logarithmic-Size (Linkable) Threshold Ring Signatures in the Plain Model. In *Public-Key Cryptography – PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, 437–467. doi:10.1007/978-3-030-97131-1_15

[23] Abida Haque and Alessandra Scafuro. 2020. Threshold ring signatures: new definitions and post-quantum security. In *IACR International Conference on Public-Key Cryptography*. Springer, 423–452.

[24] Sohyun Jeon, Calvin Abou Haidar, and Mehdi Tibouchi. 2025. LastRings: Lattice-Based Scalable Threshold Ring Signatures. In *International Conference on Information Security*. Springer, 152–172.

[25] Hao Lin, Mingqiang Wang, Weiqiang Wen, Shi-Feng Sun, and Kaitai Liang. 2025. Generic construction of threshold ring signatures and lattice-based instantiations: H. Lin et al. *Designs, Codes and Cryptography* (2025), 1–63.

[26] Joseph K Liu, Victor K Wei, and Duncan S Wong. 2003. A separable threshold ring signature scheme. In *International Conference on Information Security and Cryptology*. Springer, 12–26.

[27] Vadim Lyubashevsky. 2012. Lattice signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 738–755.

[28] Vadim Lyubashevsky and Gregory Neven. 2017. One-shot verifiable encryption from lattices. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 293–323.

[29] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. 2022. Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general. In *Annual International Cryptology Conference*. Springer, 71–101.

[30] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. 2021. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *IACR International Conference on Public-Key Cryptography*. Springer, 215–241.

[31] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2013. A toolkit for ring-LWE cryptography. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 35–54.

[32] Carlos Aguilar Melchor, Pierre-Louis Cayrel, Philippe Gaborit, and Fabien Laguillaumie. 2011. A new efficient threshold ring signature scheme based on coding theory. *IEEE Transactions on Information Theory* 57, 7 (2011), 4833–4842.
[33] Daniele Micciancio and Oded Regev. 2007. Worst-case to average-case reductions based on Gaussian measures. *SIAM journal on computing* 37, 1 (2007), 267–302.
[34] Alexander Munch-Hansen, Claudio Orlandi, and Sophia Yakoubov. 2021. Stronger Notions and a More Efficient Construction of Threshold Ring Signatures. In *Progress in Cryptology – LATINCRYPT 2021*, Patrick Longa and Carla Ràfols (Eds.). Springer International Publishing, Cham, 363–381.
[35] Shen Noether. 2015. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Report 2015/1098. ia.cr/2015/1098.
[36] Takeshi Okamoto, Raylin Tso, Michitomo Yamaguchi, and Eiji Okamoto. 2018. A k-out-of-n Ring Signature with Flexible Participation for Signers. *Cryptology ePrint Archive* (2018).
[37] Chris Peikert. 2010. An efficient and parallel Gaussian sampler for lattices. In *Annual Cryptology Conference*. Springer, 80–97.
[38] Ronald L Rivest, Adi Shamir, and Yael Tauman. 2001. How to leak a secret. In *International conference on the theory and application of cryptology and information security*. Springer, 552–565.
[39] The Monero Project. [n. d.]. Ring Size. https://www.getmonero.org/resources/moneropedia/ring-size.html. Moneropedia. Accessed: 2026-04-29.
[40] Patrick P Tsang, Victor K Wei, Tony K Chan, Man Ho Au, Joseph K Liu, and Duncan S Wong. 2004. Separable linkable threshold ring signatures. In *International Conference on Cryptology in India*. Springer, 384–398.
[41] Chunhui Wu, Youkang Zhou, Fangguo Zhang, Yusong Du, and Qiping Lin. 2025. An efficient rejection-free threshold ring signature from lattices and its application in receipt-free cloud e-voting. *Computer Standards & Interfaces* 94 (2025), 104008.
[42] Tsz Hon Yuen, Joseph K Liu, Man Ho Au, Willy Susilo, and Jianying Zhou. 2011. Threshold ring signature without random oracles. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 261–267.
[43] Jiang Zhang, Yu Yu, Shuqin Fan, Zhenfeng Zhang, and Kang Yang. 2020. Tweaking the asymmetry of asymmetric-key cryptography on lattices: KEMs and signatures of smaller sizes. In *IACR International Conference on Public-Key Cryptography*. Springer, 37–65.

Open Science

Artifacts required to reproduce major quantitative claims in this paper are released as a single self-contained repository at https://github.com/lotrs-sig/lotrs. The repository is organized as three independent components plus a sampler specification, each in its own subdirectory with a dedicated `README` and a top-level `Makefile` that wires them together.

- **Python reference implementation** (`lotrs-py`). A readable prototype of the full κ=1 pipeline (Setup, KGen, KAgg, Sign_1, Sign_2, SAgg, Vf) together with ring arithmetic and the auxiliary-prime CRT-NTT, the CDT and FACCT-style Gaussian samplers, the Golomb–Rice / fixed-width codec, and the parameter-derived bound checks. Ships with 134 unit and end-to-end tests across seven `test_*.py` modules, a reference parameter set matching Table 3, and a deterministic test-vector emitter (`vectors.py`) with seed-driven SHAKE-128 randomness. Requires Python 3.10+ and three pure-Python dependencies (`numpy`, `pycryptodome`, `mpmath`); no native extensions.
- **Rust reference implementation** (`lotrs-rs`). A performance-oriented implementation of the same protocol that is byte-exact compatible with the Python reference at the wire-format boundary. Ships with 46 unit tests, 12 Python/Rust interop tests that re-derive the Python `vectors.json` from the same seeds and compare `pp`, `sk`, `pk`, and full signatures byte-for-byte, and one sampler KAT that cross-checks the FACCT pipeline against an emitted Python reference. Requires a stable Rust toolchain ($\geq$ 1.75); a single `cargo run --release --example bench` regenerates the data underlying §6.1.
- **Parameter-selection scripts** (`estimator/`). Sage-based scripts (`lotrs_finder.py`, `lotrs_estimate.py`, `lotrs_param_checks.py`) that derive the bounds, signature size, expected attempt count, and security level reported in Table 3, together with bundled vendored copies of the lattice estimator [1] and the ASIS/MSIS estimator [18] pinned to the versions used for the paper's numbers. A reference run is committed under `estimator/` so the headline output can be diff-checked without rerunning Sage; `make reference` reproduces the diff.
- **Sampler specification** (`lotrs-facct-sampler.md`). A self-contained, language-agnostic description of the FACCT-style large-σ Gaussian sampler used at the production parameters, sufficient to re-implement the sampler from scratch and validate against the cross-language KAT.

Reproducibility. All randomness in both implementations derives from explicit seeds via SHAKE-128, so signing, key generation, and the test-vector emitter are bit-deterministic. The benchmark numbers in §6.1 were produced by a single back-to-back `cargo run --release --example bench` on a single machine, and the resulting `data.csv`, `summary.md`, and breakdown plots are committed to the repository under `lotrs-rs/bench-out/`. Aside from the Sage requirement of the estimator subdirectory, the artifact requires only an off-the-shelf Python interpreter, a Rust toolchain, and a LaTeX distribution to compile this paper from source. No proprietary datasets, models, hardware, or network services are required to evaluate any claim.

A Additional preliminaries

A.1 Invertibility in $\mathcal{R}_q$

Lemma A.1 (Invertibility in $\mathcal{R}_q$ [28], Lemma 2.2). *Let $d > 1$ be a power of two and let q be a prime with $q \equiv 5 \pmod 8$. Define the (power-of-two cyclotomic) ring $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d + 1)$. Then $\mathcal{R}_q$ contains exactly $2\,q^{d/2-1}$ non-invertible elements. Moreover, every nonzero $a \in \mathcal{R}_q$ with $\|a\|_\infty < \sqrt{q/2}$ is invertible in $\mathcal{R}_q$, where $\|\cdot\|_\infty$ denotes the maximum absolute value of the (centered) coefficients.*

A.2 Discrete Gaussian Distribution and Rejection Sampling

Gaussian parameterization and the $\sqrt{2\pi}$ conversion. We explicitly distinguish the two common parameterizations of (discrete) Gaussians for $\mathbf{x} \in \mathcal{R}^m$ centered at $\mathbf{0}$:

$$\rho_s(\mathbf{x}) := \exp\left(-\pi\|\mathbf{x}\|^2/s^2\right) \qquad \text{and} \qquad \rho_\sigma(\mathbf{x}) := \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right).$$

They coincide under the change of variables $s = \sqrt{2\pi}\,\sigma$, equivalently, $\sigma = s/\sqrt{2\pi}$. *Convention.* In the rejection sampling lemma Rej (Alg. 8) we use the standard deviation parameter σ (the $2\sigma^2$ denominator form).
For any $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ of a lattice Λ is the smallest $s > 0$ such that

$$\rho_{1/s}\left(\Lambda^* \setminus \{\mathbf{0}\}\right) \leq \varepsilon,$$

where Λ^* is the dual lattice of Λ. By Lemma 3.2 of [33], $\eta_\varepsilon(\mathcal{R}^m) \le \sqrt{dm}$ for $\varepsilon = 2^{-dm}$. In our parameter selection we choose s to exceed $\eta_\varepsilon(\mathcal{R}^m)$ by a factor at least $\sqrt{2}$, i.e.,

$$s \ \ge\ \sqrt{2}\,\eta_\varepsilon(\mathcal{R}^m) \qquad \Longleftrightarrow \qquad \sigma \ \ge\ \frac{\eta_\varepsilon(\mathcal{R}^m)}{\sqrt{\pi}}.$$

In this setting, the following lemma holds:

Lemma A.2 (Sum of independent discrete Gaussians [37]). *Let $m, n \ge 1$ and $\varepsilon > 0$ negligible. Let $\sigma_1, \dots, \sigma_n > 0$ satisfy the smoothing condition*

$$\sigma_i \ \ge\ (1/\sqrt{\pi}) \cdot \eta_\varepsilon(\mathcal{R}^m) \qquad \text{for all } i \in \{1, \dots, n\},$$

where $\eta_\varepsilon(\mathcal{R}^m)$ is the smoothing parameter of the lattice underlying $\mathcal{R}^m$. Sample independent $\mathbf{y}_i \leftarrow D^m_{\sigma_i}$ and set $\mathbf{y} := \sum_{i=1}^n \mathbf{y}_i$.
Let $\sigma := \sqrt{\sigma_1^2 + \cdots + \sigma_n^2}$. Then the distribution of $\mathbf{y}$ is within statistical distance at most 2ε of D^m_σ. In particular, if all $\sigma_i = \sigma_0$, then

$$\sum_{i=1}^{n} \mathbf{y}_i \ \approx_{2\varepsilon}\ D^m_{\sigma_0\sqrt{n}}.$$

Lemma A.3 (Gaussian tail bounds. [17], Lemma 4; [27], Lemma 4.4). *Let $n \ge 1$ and let $\sigma := \phi B$ for some $\phi > 0$ and $B > 0$. Sample $\mathbf{z} \xleftarrow{\$} D^n_\sigma$. For $t > 0$, and any z of $\mathbf{z}$,*

$$\Pr\big[|\ z\ | > t\sigma\big] \le 2\exp(-t^2/2), \quad \Pr\big[\|\mathbf{z}\|_\infty > t\sigma\big] \le 2n\exp(-t^2/2).$$

For any $t > 1$,

$$\Pr\big[\|\mathbf{z}\| > t\sigma\sqrt{n}\big] \ \le\ t^n \exp\Big(\tfrac{n}{2}(1 - t^2)\Big).$$

Note that when working over the rings $\mathcal{R}_q, \mathcal{R}_{\hat{q}}$, the sample $\mathbf{z} \xleftarrow{\$} D^n_\sigma$ will have an extra factor d when we look at the dimension over $\mathbb{Z}$.

$\mathrm{Rej}(\mathbf{z}, \mathbf{v}, \phi, K)$	$\mathrm{RejOp}(\mathbf{z}, \mathbf{v}, \phi, K)$
$\sigma := \phi K$	**if** $\langle \mathbf{z}, \mathbf{v}\rangle < 0$ **then return** 1
$\mu(\phi) := \exp\Big(\frac{12}{\phi} + \frac{1}{2\phi^2}\Big)$	$\sigma := \phi K$
$u \xleftarrow{\$} [0, 1)$	$\mu(\phi) := \exp\Big(\frac{1}{2\phi^2}\Big)$
$p := \frac{1}{\mu(\phi)} \cdot \exp\Big(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\Big)$	$u \xleftarrow{\$} [0, 1)$
if $u > p$ **then return** 1 / abort	$p := \frac{1}{\mu(\phi)} \cdot \exp\Big(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\Big)$
if $\|\mathbf{z}\|_\infty > 6\sigma$ **then return** 1	**if** $u > p$ **then return** 1
return 0	**if** $\|\mathbf{z}\|_\infty > 6\sigma$ **then return** 1
	return 0

Figure 8: Rejection sampling.

We use the rejection-sampling technique of [27] together with its optimized variant from [30]. The basic version Rej does not reveal any information about the secret vector $\mathbf{v}$. In contrast, the optimized variant RejOp explicitly exploits the fact that $\langle \mathbf{z}, \mathbf{v}\rangle \ge 0$, allowing for more efficient parameter selection. The security of this optimized approach relies on the Extended MLWE assumption; we refer the reader to [30] for further discussion. We also incorporate an infinity-norm check into the rejection-sampling procedures as a convenient shortcut in the protocol.

Lemma A.4 ([27], Lemma 3.3). *Let $n \ge 1$ and let h be a distribution over $V \subseteq \mathbb{Z}^n$ such that $\|\mathbf{v}\| \le K$ for all $\mathbf{v} \in V$. Sample $\mathbf{c} \leftarrow h$ and fix $\phi > 0$. Consider the randomized algorithm $\mathcal{F}$ that samples $\mathbf{y} \leftarrow D^n_\sigma$ for $\sigma = \phi K$ and sets $\mathbf{z} := \mathbf{y} + \mathbf{c}$, then outputs $\mathrm{Rej}(\mathbf{z}, \mathbf{c}, \phi, K)$. Then:*

(1) $\Big|\Pr[\mathcal{F}\ \text{outputs}\ 0] - \frac{1}{\mu(\phi)}\Big| \le 2^{-100}$, $\mu(\phi) := \exp\Big(\frac{12}{\phi} + \frac{1}{2\phi^2}\Big)$.
(2) Conditioned on $\mathcal{F}$ outputting 0, the statistical distance between the distribution of $\mathbf{z}$ and D^n_σ is at most 2^{-100}.

Acceptance bit convention. Our pseudocode for Rej returns 1 to abort and 0 to continue.

A.3 Hardness Assumptions

Definition A.5 ($\mathrm{MLWE}_{q,k,l,\chi}$*).* Let $\mathbf{A} \in \mathcal{R}_q^{k\times l}$ be uniform and let $\mathbf{s} \leftarrow S_\eta^l$ be a short secret. Let $\mathbf{e} \leftarrow \chi^k$ be an error distribution over $\mathcal{R}^k$. An MLWE sample is $(\mathbf{A}, \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e})$. The MLWE assumption states that MLWE samples are computationally indistinguishable from $(\mathbf{A}, \mathbf{u})$ where $\mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^k)$.

Definition A.6 ($\mathrm{MSIS}^\infty_{q,k,m,\beta}$*).* Given a uniformly random matrix $\mathbf{A} \in \mathcal{R}_q^{k\times m}$, the infinity-norm MSIS problem asks to find a nonzero short vector $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \in \mathcal{R}_q^k$ and $\|\mathbf{z}\|_\infty \le \beta$.

Definition A.7 ($\mathrm{AMSIS}^\infty_{q,k,\mathbf{m},\boldsymbol{\beta}}$ [43]*).* Let $\mathbf{m} = (m_1, \dots, m_s)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_s)$, where $s \ge 1$. Given a uniformly random matrix $\mathbf{A} \in \mathcal{R}_q^{k\times(m_1+\cdots+m_s)}$, the AMSIS problem asks to find a nonzero vector $\mathbf{z} = (\mathbf{z}_1^\top, \dots, \mathbf{z}_s^\top)^\top \in \mathcal{R}^{m_1+\cdots+m_s}$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \in \mathcal{R}_q^k$ and, for every $i \in \{1, \dots, s\}$, $\mathbf{z}_i \in \mathcal{R}^{m_i}$ and $\|\mathbf{z}_i\|_\infty \le \beta_i$.

AMSIS generalizes MSIS by allowing different blocks of the solution vector to satisfy different norm bounds, rather than imposing a single uniform bound on the whole vector. This gives finer parameter control: efficiency-critical components can be kept small, while other components can tolerate larger bounds to maintain the desired hardness level.

Lemma A.8 (Regularity [31]; [8], Lemma 5). *Let $k, l \in \mathbb{Z}^+$ be integers and $l + k \le \mathrm{poly}(d)$. Sample $\mathbf{A} \xleftarrow{\$} R_q^{k\times l}$ and define $\bar{\mathbf{A}} := [\mathbf{A} \mid \mathbf{I}] \in R_q^{k\times(l+k)}$. Let $\sigma := \phi B$. For $\sigma > 2d/\sqrt{2\pi} \cdot q^{k/(l+k)+2/d(l+k)}$ let $\mathbf{x} \leftarrow D^{l+k}_\sigma$. Then, with probability at least $1 - 2^{-\Omega(d)}$ over the choice of $\mathbf{A}$, the distribution of $\bar{\mathbf{A}}\mathbf{x} \in \mathcal{R}_q^k$ satisfies the following bound for every $\mathbf{y} \in \mathcal{R}_q^k$:*

$$\Pr[\bar{\mathbf{A}}\mathbf{x} = \mathbf{y}] \ \in\ \big(1 \pm 2^{-\Omega(d)}\big)\, q^{-dk}.$$

In particular, the distribution of $\bar{\mathbf{A}}\mathbf{x}$ is within statistical distance $2^{-\Omega(d)}$ of the uniform distribution over $\mathcal{R}_q^k$.

A.4 Technical Lemmas

The following technical lemmas are needed for our proofs.

Lemma A.9 ([19], Lemma D.1). *Let $a, b \in \mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$. Then:*

(1) $\|ab\| \le \sqrt{d}\,\|a\|\,\|b\|$.
(2) $\|a\| \le \sqrt{d}\,\|a\|_\infty$ *and* $\|a\| \le \|a\|_1 \le \sqrt{d}\,\|a\|$.
(3) $\|ab\|_\infty \le \|a\|\,\|b\|$.
(4) $\|ab\|_\infty \le \|a\|_1\,\|b\|_\infty$.
(5) For $a_1, \dots, a_n \in \mathcal{R}$: $\big\|\prod_{i=1}^n a_i\big\|_\infty \le \big(\prod_{i=1}^{n-1} \|a_i\|_1\big)\,\|a_n\|_\infty$.

A.5 Vandermonde matrices and some linear algebra

We briefly recall some linear-algebra facts from [16] needed for our unforgeability proof. Let $\mathbf{I}_n$ denote the n-dimensional identity matrix, and let all matrices below be defined over a ring $\mathcal{R}$. For an $n \times n$ matrix $\mathbf{A}$, we write $\det(\mathbf{A})$ for its determinant. The adjugate of $\mathbf{A}$, denoted $\mathrm{adj}(\mathbf{A})$ (the transpose of the cofactor matrix), satisfies

$$\mathrm{adj}(\mathbf{A})\mathbf{A} \;=\; \mathbf{A}\,\mathrm{adj}(\mathbf{A}) \;=\; \det(\mathbf{A})\,\mathbf{I}_n.$$

In particular, if $\mathbf{A}$ is nonsingular, then $\mathrm{adj}(\mathbf{A}) = \det(\mathbf{A})\mathbf{A}^{-1}$.

Vandermonde matrices. Fix elements $x^{(0)}, \ldots, x^{(\kappa)} \in \mathcal{R}$. The $(\kappa+1)$-dimensional Vandermonde matrix $\mathbf{V}$ is

$$\mathbf{V} \;=\; \begin{pmatrix} 1 & x^{(0)} & (x^{(0)})^2 & \cdots & (x^{(0)})^{\kappa} \\ 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{\kappa} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(\kappa)} & (x^{(\kappa)})^2 & \cdots & (x^{(\kappa)})^{\kappa} \end{pmatrix},$$

and

$$\det(\mathbf{V}) = \prod_{0 \le i < j \le \kappa} \left(x^{(j)} - x^{(i)}\right).$$

Fact 1. *The Vandermonde determinant* $\det(\mathbf{V})$ *is a product of* $\binom{\kappa+1}{2}$ *pairwise differences of the form* $x^{(j)} - x^{(i)}$ *with* $i \neq j$.

Inverse structure and adjugate row coefficients. As noted in [16], when $\mathbf{V}^{-1}$ exists it has the following structure:

$$\mathbf{V}^{-1} = \left(v_{a,b}\right)_{a,b \in [0,\kappa]}, \text{where, for each column } b \in [0,\kappa],$$

$$v_{a,b} = \frac{*_{a,b}}{\Delta_b},\; \Delta_b = \prod_{\substack{i \in [0,\kappa] \\ i \neq b}} \left(x^{(b)} - x^{(i)}\right),\; \text{and},\; v_{\kappa,b} \;=\; \frac{(-1)^{\kappa-b}}{\Delta_b},$$

where $*_{a,b}$ denotes some element in the ring computed from the $x^(i)'s$. From this form, $\mathbf{V}^{-1}$ exists over $\mathcal{R}$ if and only if every difference $x^{(j)} - x^{(i)}$ (for $0 \le i < j \le \kappa$) is invertible in $\mathcal{R}$. Since $\mathrm{adj}(\mathbf{V}) = \det(\mathbf{V})\mathbf{V}^{-1}$ whenever $\mathbf{V}$ is nonsingular, the row/column structure of $\mathrm{adj}(\mathbf{V})$ is obtained by cancelling the corresponding denominators against $\det(\mathbf{V})$. In particular, let $(\Gamma_0, \ldots, \Gamma_\kappa)$ denote the last row of $\mathrm{adj}(\mathbf{V})$. Then

$$\Gamma_i = (-1)^{i+\kappa} \prod_{\substack{0 \le \ell < j \le \kappa \\ j,\ell \neq i}} \left(x^{(j)} - x^{(\ell)}\right), \qquad \text{for } 0 \le i \le \kappa,$$

and each Γ_i is a product of exactly $\kappa(\kappa-1)/2$ such differences.

B More on Structured Threshold Ring Signatures

To model active adversaries, we consider an adversary $\mathcal{A}$ with access to the following oracles $\mathsf{O} := \{\mathsf{OKGen}, \mathsf{OSign}, \mathsf{OCorr}, \mathsf{OReg}, \mathsf{H}\}$. Let P be the set of honestly generated public keys, P_{corr} the set of corrupted (public) keys, and for each message μ let P^{μ}_{sig} be the set of public keys that have appeared in a signing-oracle query on message μ.

- $\mathsf{OKGen}(i)$: generate $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})$, return pk_i, and update $P \leftarrow P \cup \{\mathsf{pk}_i\}$.
- $\mathsf{OSign}(\mu, U, (R, \Psi))$: the adversary requests a signature on message μ with respect to structured ring instance (R, Ψ) and admissible signer set $U \in \mathcal{F}_\Psi(R)$. The oracle executes the honest parties' steps in Sign using the secret keys it knows for the honest signers in U, while $\mathcal{A}$ specifies the behavior of the corrupted participants; it outputs the resulting σ and updates $P^{\mu}_{\text{sig}} \leftarrow P^{\mu}_{\text{sig}} \cup U$.
- $\mathsf{OCorr}(i)$: if $\mathsf{pk}_i \notin P$ or $\mathsf{pk}_i \in P_{\text{corr}}$, return $\perp$; otherwise return sk_i and update corrupt keys: $P_{\text{corr}} \leftarrow P_{\text{corr}} \cup \{\mathsf{pk}_i\}$.
- $\mathsf{OReg}(i, \mathsf{pk})$: if $\mathsf{pk} \in P$, return $\perp$; otherwise set $\mathsf{pk}_i \coloneqq \mathsf{pk}$ and update registered keys as $P \leftarrow P \cup \{\mathsf{pk}\}$ and corrupt keys as $P_{\text{corr}} \leftarrow P_{\text{corr}} \cup \{\mathsf{pk}\}$.
- H: the set of scheme-dependent random oracles.

Definition 2.1 (Completeness). A structured threshold ring signature scheme sTRS is complete if for all probabalistic polynomial time (PPT) adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\begin{array}{l|l} \mathsf{Vf}(\mu, \sigma, & \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (R, \Psi)) = 0 & \{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})\}_{i \in |R|}, \\ & (\mu, U, (R, \Psi)) \leftarrow A(\mathsf{pp}), \\ & U \in \mathcal{F}_\Psi(R), \\ & \sigma \leftarrow \mathsf{Sign}(\mu, S_U, (R, \Psi)) \end{array}\right] \le \mathsf{negl}(\lambda).$$

Definition 2.2 (Unforgeability). A threshold ring signature scheme $\mathsf{sTRS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ satisfies unforgeability if, for every PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{sTRS},\mathcal{A}}(\lambda) = 1\right] \;\le\; \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{sTRS},\mathcal{A}}(\lambda)$ is defined in Fig. 9.

Definition 2.3 (Selective Unforgeability). A structured threshold ring signature scheme sTRS is *selectively unforgeable* if for every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\mathsf{Exp}^{\mathsf{sel\text{-}UF\text{-}CMA}}_{\mathsf{sTRS},\mathcal{A}}(\lambda) = 1\right] \le \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{Exp}^{\mathsf{sel\text{-}UF\text{-}CMA}}_{\mathsf{TRS},\mathcal{A}}(\lambda)$ is defined in Fig. 9.

Definition 2.4 (Anonymity). A structured threshold ring signature scheme sTRS satisfies anonymity if for all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that

$$\left|\Pr\left[\mathsf{Exp}^{\mathsf{anon\text{-}0}}_{\mathsf{sTRS},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{anon\text{-}1}}_{\mathsf{sTRS},\mathcal{A}}(\lambda) = 1\right]\right| \le \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{Exp}^{\mathsf{anon}\text{-}b}_{\mathsf{sTRS},\mathcal{A}}(\lambda)$ is defined in Fig. 10.

C Correctness of LoTRS

Proof. The verifier can reject an honestly generated signature only if some restart happens, or if an explicit norm check fails. Thus, by a union bound we upper-bound the rejection probability by the sum of these bad-event probabilities:

$$\begin{aligned}\Pr\left[\mathsf{Vf}(\mathsf{pp}, \mu, \tilde{\sigma}, \mathsf{PK}) = 0\right] \le\; & \Pr[E_{\mathsf{Rej}_z}] + \Pr[E_{\mathsf{Rej}_f}] + \Pr[E_{\mathsf{Rej}_{z_b}}] \\ & + \Pr[E_{\mathrm{BG}}] + \Pr[E_\infty] + \Pr[E_2] + \Pr[E_w],\end{aligned}$$

where E_* are one of the following bad events:

$\mathsf{Exp}^{\text{UF-CMA}}_{\text{sTRS},\mathcal{A}}(\lambda)$

1 : $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2 : $P \coloneqq \emptyset,\ P_{\text{corr}} \coloneqq \emptyset,\ P^{\mu}_{\text{sig}} \coloneqq \emptyset$ for all messages μ
3 : $(\mu^*, \sigma^*, (R^*, \Psi^*)) \leftarrow \mathcal{A}^{\mathsf{O}}(\mathsf{pp})$
4 : **if for every** $U \in \mathcal{F}_{\Psi^*}(R^*)$:
$\quad |U \cap (P_{\text{corr}} \cup P^{\mu^*}_{\text{sig}})| < T$
5 : $\quad \wedge$ **for no** $U \in \mathcal{F}_{\Psi^*}(R^*)$ was
$\quad (\mu^*, U, (R^*, \Psi^*))$ queried to OSign
6 : $\quad \wedge\ \mathsf{Vf}(\mathsf{pp}, \mu^*, \sigma^*, (R^*, \Psi^*)) = 1$
7 : **then return** 1
8 : **return** 0

$\mathsf{Exp}^{\text{sel-UF-CMA}}_{\text{sTRS},\mathcal{A}}(\lambda)$

1 : $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2 : $P \coloneqq \emptyset,\ P_{\text{corr}} \coloneqq \emptyset,\ P^{\mu}_{\text{sig}} \coloneqq \emptyset$ for all messages μ
3 : $i^* \leftarrow \mathcal{A}_0(\mathsf{pp})$
/ i^* indicates query index on H of future forgery on some μ^*
4 : $(\sigma^*, (R^*, \Psi^*)) \leftarrow \mathcal{A}_1^{\mathsf{O}}(\mathsf{pp})$
5 : **if for every** $U \in \mathcal{F}_{\Psi^*}(R^*)$:
$\quad |U \cap (P_{\text{corr}} \cup P^{\mu^*}_{\text{sig}})| < T$
6 : $\quad \wedge$ **for no** $U \in \mathcal{F}_{\Psi^*}(R^*)$ was
$\quad (\mu^*, U, (R^*, \Psi^*))$ queried to OSign
7 : $\quad \wedge\ \mathsf{Vf}(\mu^*, \sigma^*, (R^*, \Psi^*)) = 1$
8 : **then return** 1
9 : **return** 0

Figure 9: Unforgeability experiments for the security of structured threshold ring signatures. $\mathsf{Exp}^{\text{UF-CMA}}_{\text{sTRS},\mathcal{A}}$ **step 4 (resp.** $\mathsf{Exp}^{\text{sel-UF-CMA}}_{\text{sTRS},\mathcal{A}}$ **step 5) intuitively says that every admissible signer set** U **has fewer than** T **keys (i) controlled by** $\mathcal{A}$ **or (ii) already used with oracle help on the target message. In** $\mathsf{Exp}^{\text{sel-UF-CMA}}_{\text{sTRS},\mathcal{A}}$**, the value** i^* **denotes the adversary's pre-committed target random-oracle query (in our original proof, the eventual** H_{com}**-query on** μ^* **underlying the forgery), which allows the reduction to embed the hard instance at that designated point.**

$\mathsf{Exp}^{\text{anon-}b}_{\text{sTRS},\mathcal{A}}(\lambda)$

1 : $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2 : $P \coloneqq \emptyset,\ P_{\text{corr}} \coloneqq \emptyset,\ P^{\mu}_{\text{sig}} \coloneqq \emptyset$ for all messages μ
3 : $(\mu^*, U_0, U_1, (R^*, \Psi^*)) \leftarrow \mathcal{A}^{\mathsf{O}}(\mathsf{pp})$
4 : **if** $U_0, U_1 \in \mathcal{F}_{\Psi^*}(R^*)$
$\quad \wedge\ U_0 \cap P_{\text{corr}} = \emptyset\ \wedge\ U_1 \cap P_{\text{corr}} = \emptyset$
5 : **then**
6 : $\quad \sigma^* \leftarrow \mathsf{Sign}(\mu^*, S_{U_b}, (R^*, \Psi^*))$
7 : $\quad b^* \leftarrow \mathcal{A}(\sigma^*)$
8 : $\quad$ **if** $b^* = b$ **then return** 1
9 : **return** 0

Figure 10: Anonymity experiment for structured threshold ring signatures. $\mathsf{Exp}^{\text{anon-}b}_{\text{sTRS},\mathcal{A}}$ **step 4 requires admissible signer sets** U_0 **and** U_1 **to not have any corrupted keys.**

Event E_{Rej_z}**.** Let E_{Rej} be the event that Sign_2 restarts because Rej rejects the response $\mathbf{z}_u$. In Sign_2, each signer forms

$$\mathbf{z}_u \ :=\ x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j} \ \in\ \mathcal{R}^{k+l},$$

and then runs $\mathsf{Rej}(\mathbf{z}_u,\ x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j},\ \phi_0,\ B_0)$. Thus,

$$E_{\text{Rej}_z} \ :=\ \Big\{\mathsf{Rej}(\mathbf{z}_u,\ x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j},\ \phi_0,\ B_0) = 1\Big\}.$$

We require a bound on the secret-dependent shift:

$$\| \underbrace{x^\kappa \alpha_u \mathbf{s}_u}_{\text{secret term}} - \underbrace{\sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}}_{\text{random non-masking terms}} \| \ \le\ B_0.$$

By the triangle inequality, Lemma A.9 and the sum of a finite geometric series,

$$\begin{aligned}
\|x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j}\| \ &\le\ \|x^\kappa \alpha_u \mathbf{s}_u\| + \sum_{j=1}^{\kappa-1} \|x^j \mathbf{y}_{u,j}\| \\
&\le \sqrt{d(k+l)}\ \|x^\kappa \alpha_u \mathbf{s}_u\|_\infty + \sqrt{d(k+l)} \sum_{j=1}^{\kappa-1} \|x^j \mathbf{y}_{u,j}\|_\infty \\
&\le \sqrt{d(k+l)}\Big(\|x\|_1^\kappa\ \|\alpha_u\|_1\ \|\mathbf{s}_u\|_\infty + \sum_{j=1}^{\kappa-1} \|x\|_1^j \|\mathbf{y}_{u,j}\|_\infty\Big) \\
&\le \sqrt{d(l+k)}\Big(w^{\kappa+1}\eta + 6\phi_s B_s \sum_{j=1}^{\kappa-1} \|x\|_1^j\Big) \\
&\le \sqrt{d(l+k)}\Big(w^{\kappa+1}\eta + 6\phi_s B_s \cdot \frac{w^\kappa - w}{w-1}\Big), \text{ for } w \neq 1. \\
&\le \sqrt{d(l+k)}\Big(w^{\kappa+1}\eta + 6\phi_s B_s (\kappa-1) w^{\kappa-1}\Big), \qquad \text{for } w > 1. \\
&=: B_0.
\end{aligned}$$

Therefore, in Sign_2, each signer's $\mathbf{z}_u$ is accepted with probability $1/\mu(\phi_0) + \varepsilon_0$, where $|\varepsilon_0| \le 2^{-100}$ by Lemma A.4. So conditioned on acceptance, the distribution of $\mathbf{z}_u$ is statistically close to $D^{k+l}_{\phi_0 B_0}$. For ϕ_0, each signer restarts approximately $\mu(\phi_0)$ many times and so the expected number of restarts for T parties is $\mu(\phi_0)^T$. Therefore,

$$\Pr[E_{\text{Rej}_z}] \approx 1 - \Big(\frac{1}{\mu(\phi_0)}\Big)^T.$$

Remark 4. *In Table 5, $\phi_0 = \phi = \phi_{0'}$.*

Event E_{Rej_f}. In $\mathsf{Sign}_{\mathsf{bin}}$, the prover computes $f_{j,i_j} := x\,\delta_{\ell_j,i_j} + a_{j,i_j}$ for $j \in [0,\kappa-1]$ and $i_j \in [0,\beta-1]$, and sets

$$\mathbf{f}_1 := (f_{j,1},\ldots,f_{j,\beta-1})_{j\in[0,\kappa-1]} \in \mathcal{R}^{\kappa(\beta-1)},$$
$$\mathbf{b}_1 := (\delta_{\ell_j,1},\ldots,\delta_{\ell_j,\beta-1})_{j\in[0,\kappa-1]}.$$

Rejection sampling is applied as $\mathsf{Rej}(\mathbf{f}_1, x\mathbf{b}_1, \phi_a, B_a)$, and $\mathsf{Sign}_{\mathsf{bin}}$ restarts iff it rejects. We define

$$E_{\mathrm{Rej}_f} \;:=\; \big\{\mathsf{Rej}(\mathbf{f}_1, x\mathbf{b}_1, \phi_a, B_a) = 1\big\}.$$

To apply Lemma A.4, we need a bound on the shift:

$$\|x\mathbf{b}_1\| \;\le\; B_a.$$

Since $\mathbf{b}_1$ has at most one 1 per $\beta-1$-sized block (with the rest zero), and there are κ such blocks, there will be at most κw non-zero coefficients in $x\mathbf{b}_1$ where each coefficient is in $\{-1,0,1\}$:

$$\|x\mathbf{b}_1\| = \|x(\delta_{\ell_j,1},\ldots,\delta_{\ell_j,\beta-1})_{j\in[0,\kappa-1]}\| \;\le\; \sqrt{\kappa w}.$$

Hence it suffices to ensure $B_a \ge \sqrt{\kappa w}$. Therefore, by Lemma A.4, the acceptance probability is $1/\mu(\phi_a) + \varepsilon_a$ with $|\varepsilon_a| \le 2^{-100}$. Since each (honest) signer has exactly the same $\mathsf{Sign}_{\mathsf{bin}}$ execution, the success probability for all signers is equivalent to that of one signer per signing session. Therefore,

$$\Pr[E_{\mathrm{Rej}_f}] \approx 1 - \frac{1}{\mu(\phi_a)}.$$

Event $E_{\mathrm{Rej}_{z_b}}$. In $\mathsf{Sign}_{\mathsf{bin}}$, $\mathbf{z}_b := \mathbf{r}_a + x\mathbf{r}_b$ with $\mathbf{r}_a \leftarrow D^{\hat{n}+\hat{k}}_{\phi_b B_b}$ and $\mathbf{r}_b \leftarrow S_1^{\hat{n}+\hat{k}}$. By Lemma A.9,

$$\begin{aligned}\|x\mathbf{r}_b\| &\le \sqrt{d(\hat{n}+\hat{k})}\,\|x\mathbf{r}_b\|_\infty \\ &\le \sqrt{d(\hat{n}+\hat{k})}\,\|x\|_1\|\mathbf{r}_b\|_\infty \\ &\le \sqrt{d(\hat{n}+\hat{k})}\,w.\end{aligned}$$

where we used $\|\mathbf{r}_b\|_\infty \le 1$. So, we can set $B_b \ge \sqrt{d(\hat{n}+\hat{k})}\,w$, and by Lemma A.4, the acceptance probability for $\mathbf{z}_b$ is $1/\mu(\phi_b) + \varepsilon_b$ with $|\,\varepsilon_b\,| \le 2^{-100}$ and

$$\Pr[E_{\mathrm{Rej}_{z_b}}] \approx 1 - \frac{1}{\mu(\phi_b)}.$$

Event E_{BG} (Bai–Galbraith check). Recall that in the binary proof the signer computes

$$\hat{A}_{\mathsf{bin}} := \mathbf{G}(\mathbf{z}_b, \mathbf{f}, \mathbf{g})^\top - x\,2^{K_B} B^{(1)}_{\mathsf{bin}},$$

writes

$$\hat{A}_{\mathsf{bin}} = 2^{K_A}\hat{A}^{(1)}_{\mathsf{bin}} + \hat{A}^{(0)}_{\mathsf{bin}},$$

and restarts unless

$$\big\|\hat{A}^{(0)}_{\mathsf{bin}}\big\|_\infty \le 2^{K_A-1} - w2^{K_B-1}.$$

We therefore define

$$E_{\mathrm{BG}} := \left\{\big\|\hat{A}^{(0)}_{\mathsf{bin}}\big\|_\infty > 2^{K_A-1} - w2^{K_B-1}\right\}.$$

Following the same coefficient-wise heuristic used in the Dilithium [13] acceptance analysis, we model the centered coefficients of $\hat{A}_{\mathsf{bin}} \bmod 2^{K_A}$ as independently uniform over the 2^{K_A} possible residues. Then a single coefficient passes the stability check with probability

$$p_{\mathrm{BG},1} = \frac{2\big(2^{K_A-1} - w2^{K_B-1}\big)+1}{2^{K_A}} = \frac{2^{K_A} - w2^{K_B}+1}{2^{K_A}} = 1 - \frac{w2^{K_B}-1}{2^{K_A}}.$$

Since $\hat{A}_{\mathsf{bin}} \in \mathcal{R}_q^{\hat{n}}$ has $\hat{n}d$ coefficients, the overall acceptance probability is heuristically

$$p_{\mathrm{BG}} \approx \left(1 - \frac{w2^{K_B}-1}{2^{K_A}}\right)^{\hat{n}d} \approx e^{-\hat{n}d\,\frac{w2^{K_B-1}}{2^{K_A}}}.$$

Therefore,

$$\Pr[E_{\mathrm{BG}}] \approx 1 - p_{\mathrm{BG}} \approx 1 - e^{-\hat{n}d\,\frac{w2^{K_B-1}}{2^{K_A}}},$$

and the expected restart factor contributed by the Bai–Galbraith check is

$$\mu_{\mathrm{BG}} := p_{\mathrm{BG}}^{-1} \approx e^{\hat{n}d\,\frac{w2^{K_B-1}}{2^{K_A}}}.$$

Event E_∞ ($\mathbf{f}_1, \mathbf{f}_0, \mathbf{z}_b, \mathbf{g}_0, \mathbf{g}_1$ bound checks). We now distinguish between the original rejection-sampling bound for $\mathbf{z}_b$ and the tighter high-probability bounds used for $\mathbf{f}_1, \mathbf{f}_0, \mathbf{g}_0, \mathbf{g}_1$. Fix a total failure budget $\varepsilon_{\mathrm{tot}} > 0$, and set

$$\varepsilon_{\mathrm{each}} := \frac{\varepsilon_{\mathrm{tot}}}{4}.$$

We choose the bounds $B_{f_1}, B_{f_0}, B_{g_0}, B_{g_1}$ so that, in an honest execution before the corresponding explicit signer-side norm checks,

$$\Pr\big[\|\mathbf{f}_1\|_\infty > B_{f_1}\big] \le \varepsilon_{\mathrm{each}}, \qquad \Pr\big[\|\mathbf{f}_0\|_\infty > B_{f_0}\big] \le \varepsilon_{\mathrm{each}},$$
$$\Pr\big[\|\mathbf{g}_1\|_\infty > B_{g_1}\big] \le \varepsilon_{\mathrm{each}}, \qquad \Pr\big[\|\mathbf{g}_0\|_\infty > B_{g_0}\big] \le \varepsilon_{\mathrm{each}}.$$

We define the corresponding signer-side restart events

$$E^{\mathrm{rs}}_{f_1} := \big\{\|\mathbf{f}_1\|_\infty > B_{f_1}\big\}, \qquad E^{\mathrm{rs}}_{f_0} := \big\{\|\mathbf{f}_0\|_\infty > B_{f_0}\big\},$$
$$E^{\mathrm{rs}}_{g_1} := \big\{\|\mathbf{g}_1\|_\infty > B_{g_1}\big\}, \qquad E^{\mathrm{rs}}_{g_0} := \big\{\|\mathbf{g}_0\|_\infty > B_{g_0}\big\}.$$

By the union bound,

$$\Pr\big[E^{\mathrm{rs}}_{f_1} \vee E^{\mathrm{rs}}_{f_0} \vee E^{\mathrm{rs}}_{g_1} \vee E^{\mathrm{rs}}_{g_0}\big] \le 4\,\varepsilon_{\mathrm{each}} = \varepsilon_{\mathrm{tot}}.$$

The signer explicitly checks these four bounds and restarts whenever any of the events above occurs. Hence, conditioned on a transcript being output, the verifier-side bad events associated with these bounds have probability 0. Concretely, defining

$$E_{f_1} := \big\{\|\mathbf{f}_1\|_\infty > B_{f_1}\big\}, \qquad E_{f_0} := \big\{\|\mathbf{f}_0\|_\infty > B_{f_0}\big\},$$
$$E_g := \big\{\|\mathbf{g}_0\|_\infty > B_{g_0}\big\} \vee \big\{\|\mathbf{g}_1\|_\infty > B_{g_1}\big\},$$

we have

$$\Pr[E_{f_1}] = \Pr[E_{f_0}] = \Pr[E_g] = 0$$

for accepted transcripts.

Event E_{z_b}. We keep the original rejection-sampling treatment for $\mathbf{z}_b$. Recall that Vf rejects if

$$\|\mathbf{z}_b\|_\infty > 6\phi_b B_b.$$

We define

$$E_{z_b} := \big\{\|\mathbf{z}_b\|_\infty > 6\phi_b B_b\big\}.$$

This bound check is already performed during rejection sampling on the signer side, so for accepted transcripts

$$\Pr[E_{z_b}] = 0.$$

Putting everything together, we may write

$$E_\infty := E_{f_1} \vee E_{f_0} \vee E_{z_b} \vee E_g.$$

The verifier-side failure probability from these checks is 0 on accepted transcripts, while the additional signer-side restart probability contributed by the tightened $\mathbf{f}_1, \mathbf{f}_0, \mathbf{g}_0, \mathbf{g}_1$ bounds is at most ε_{tot}.

Choice of $B_{f_1}, B_{f_0}, B_{g_0}, B_{g_1}$. We now choose tighter *high-probability* bounds for $\mathbf{f}_1, \mathbf{f}_0, \mathbf{g}_0, \mathbf{g}_1$, while keeping the original rejection-sampling bound for $\mathbf{z}_b$.

Let

$$\sigma_a := \phi_a B_a, \qquad \varepsilon_{\text{each}} := \frac{\varepsilon_{\text{tot}}}{4}.$$

Bound for $\mathbf{f}_1$. For $j \in [0, \kappa-1]$ and $i \in [1, \beta-1]$, we have

$$f_{j,i} = x\,\delta_{\ell_j,i} + a_{j,i},$$

where each coefficient of $a_{j,i}$ is sampled from a discrete Gaussian with standard deviation σ_a, and the deterministic shift $x\,\delta_{\ell_j,i}$ contributes at most 1 in coefficient infinity norm. There are $M_{f_1} := \kappa(\beta-1)d$ coefficients in $\mathbf{f}_1$ in total. Define

$$\tau_{f_1} := \sqrt{2\ln\left(\frac{2M_{f_1}}{\varepsilon_{\text{each}}}\right)}.$$

Then, by a coefficient-wise Gaussian tail bound together with the union bound, it suffices to set $B_{f_1} := 1 + \tau_{f_1}\sigma_a$, so that

$$\Pr\left[\|\mathbf{f}_1\|_\infty > B_{f_1}\right] \le \varepsilon_{\text{each}}.$$

Bound for $\mathbf{f}_0$. For each $j \in [0, \kappa-1]$,

$$f_{j,0} = x\,\delta_{\ell_j,0} - \sum_{i=1}^{\beta-1} a_{j,i}.$$

Thus each coefficient of $f_{j,0}$ consists of a deterministic shift of size at most 1, plus the sum of $(\beta-1)$ independent discrete Gaussian coefficients of standard deviation σ_a. Approximating this sum by a discrete Gaussian with standard deviation $\sqrt{\beta-1}\,\sigma_a$, and using a union bound over the $M_{f_0} := \kappa d$ coefficients of $\mathbf{f}_0$, define

$$\tau_{f_0} := \sqrt{2\ln\left(\frac{2M_{f_0}}{\varepsilon_{\text{each}}}\right)}.$$

It then suffices to set $B_{f_0} := 1 + \tau_{f_0}\sqrt{\beta-1}\,\sigma_a$, so that

$$\Pr\left[\|\mathbf{f}_0\|_\infty > B_{f_0}\right] \le \varepsilon_{\text{each}}.$$

Bound for $\mathbf{g}_1$. For $i \ge 1$,

$$g_{j,i} = f_{j,i}(x - f_{j,i}) = x f_{j,i} - f_{j,i}^2.$$

Using Lemma A.9, $\|x\|_1 = w$, and the bound $\|f_{j,i}\|_\infty \le B_{f_1}$, we obtain

$$\|g_{j,i}\|_\infty \le \|x f_{j,i}\|_\infty + \|f_{j,i}^2\|_\infty \le w\,B_{f_1} + d\,B_{f_1}^2.$$

Hence it suffices to set $B_{g_1} := w\,B_{f_1} + d\,B_{f_1}^2$.

Bound for $\mathbf{g}_0$. Similarly, $g_{j,0} = f_{j,0}(x - f_{j,0}) = x f_{j,0} - f_{j,0}^2$. Using $\|f_{j,0}\|_\infty \le B_{f_0}$, we get

$$\|g_{j,0}\|_\infty \le \|x f_{j,0}\|_\infty + \|f_{j,0}^2\|_\infty \le w\,B_{f_0} + d\,B_{f_0}^2.$$

Hence it suffices to set $B_{g_0} := w\,B_{f_0} + d\,B_{f_0}^2$.

Interpretation. The bounds $B_{f_1}, B_{f_0}, B_{g_1}, B_{g_0}$ are not worst-case deterministic bounds; rather, they are chosen so that in an honest execution their violation probabilities are collectively at most ε_{tot} by the union bound. The signer explicitly checks these bounds and restarts if any of them is violated, and the verifier checks the same bounds on the final transcript. So, the extra restart contribution is approximately

$$\mu_{fg} \approx \frac{1}{1-\varepsilon_{\text{tot}}}$$

.

Event E_2. Recall that Vf rejects if any of the ℓ_2-bounds fail. We define

$$E_2 := \left\{\|\tilde{\mathbf{z}}\| > B_{\tilde{z}} \;\vee\; \|\tilde{\mathbf{r}}\| > B_{\tilde{r}} \;\vee\; \|\tilde{\mathbf{e}}\| > B_{\tilde{e}}\right\}.$$

By a union bound,

$$\Pr[E_2] \le \Pr[\|\tilde{\mathbf{z}}\| > B_{\tilde{z}}] + \Pr[\|\tilde{\mathbf{r}}\| > B_{\tilde{r}}] + \Pr[\|\tilde{\mathbf{e}}\| > B_{\tilde{e}}].$$

For $\tilde{\mathbf{z}} = \sum_{u=0}^{T-1} \mathbf{z}_u$, by Lemma A.4 each $\mathbf{z}_u$ is statistically close to $D^l_{\phi_0 B_0}$. Applying Lemma A.3 to $\mathbf{z}_u$ gives

$$\Pr\left[\|\mathbf{z}_u\| > t\phi_0 B_0\sqrt{dl}\right] \le t^{dl}\exp\left(\frac{dl}{2}(1-t^2)\right).$$

Taking $t = 1.2$, gives

$$\Pr\left[\|\mathbf{z}_u\| > 1.2\phi_0 B_0\sqrt{dl}\right] \le 1.2^{dl}\exp(-0.22l).$$

Therefore, define $B_z := 1.2\phi_0 B_0\sqrt{dl}$. Thus, by Lemma A.2, $\tilde{\mathbf{z}}$ is statistically close to the corresponding sum distribution $D^{dl}_{\phi_0 B_0\sqrt{T}}$ and so define $B_{\tilde{z}} = 1.2\phi_0 B_0\sqrt{Tdl}$.

Similarly, $\mathbf{r}_u$ is statistically close to $D^{l'}_{\phi_{0'}B_{0'}}$, so using Lemma A.3 with $t = 1.2$ gives

$$\Pr\left[\|\mathbf{r}_u\| > 1.2\phi'_0 B'_0\sqrt{dl'}\right] \le 1.2^{dl'}\exp(-0.22l').$$

So, define $B_r := 1.2\phi'_0 B'_0\sqrt{dl'}$ and $B_{\tilde{r}} := 1.2\phi'_0 B'_0\sqrt{Tdl'}$.

Lastly, for $\tilde{\mathbf{e}}$ statistically close to $D_{\phi_{\hat{0}}B_{\hat{0}}\sqrt{T}}$, where we have $\phi_{\hat{0}}B_{\hat{0}} := \max(\phi_0 B_0, \phi_{0'}B_{0'})$ using Lemma A.3 with $t = 1.2$ gives

$$\Pr\left[\|\mathbf{e}\| > 1.2\phi_{\hat{0}}B_{\hat{0}}\sqrt{dk}\right] \le 1.2^{dk}\exp(-0.22dk).$$

So, define $B_e := 1.2\phi_{\hat{0}}B_{\hat{0}}\sqrt{dk}$ and $B_{\tilde{e}} := 1.2\phi_{\hat{0}}B_{\hat{0}}\sqrt{Tdk}$.

For this choice of t, we have $\Pr[E_2] := \epsilon_{\text{agg}}$ for some negligible value.

Binding range for the augmented commitment. Since we have made use of nearly the same construction as Esgin. et al. [16], [18] in our Sign_{bin} algorithm, we can apply Lemma 1 of [16] to ensure our commitment $\mathbf{G}$ is hiding. Specifically, for hiding, we require $\text{MLWE}_{\hat{q},\hat{n},\hat{k},1}$ to be hard. However, once we analyze the compressed binary proof through the exact augmented-commitment route, the extraction argument is carried out with $\mathbf{G}^* := [\,\mathbf{G} \mid -\mathbf{I}_{\hat{n}}\,]$ and the associated map $\text{Com}_{\mathbf{G}^*}(\mathbf{u}, \mathbf{v}, \mathbf{e}; \mathbf{r}) := \mathbf{G}(\mathbf{r}, \mathbf{u}, \mathbf{v})^\top - \mathbf{e}$. This augmented map is used only as a proof device, and we do *not* need a hiding statement for it. What we need is that it be binding for all short openings that occur in the compressed extraction proof.

Accordingly, we assume that $\text{Com}_{\mathbf{G}^*}$ is γ^*_{com}-binding in the relevant norm range; namely, except with negligible probability, no

PPT adversary can produce two distinct valid short openings in this range to the same commitment value under $\mathbf{G}^*$. That is,

$$\mathsf{MSIS}^{\infty}_{\hat{q},\hat{n},\,2\kappa\beta+\hat{n},\,2\gamma^*_{\mathrm{com}}}.$$

is hard. A sufficient choice is

$$\gamma^*_{\mathrm{com}} := \max\Big(B_{g_0},\, B_{g_1},\, B_{f_1} := 6\phi_a B_a,\, B_{z_b} := 6\phi_b B_b,\, 2^{K_B-1},\\ 2^{K_A-1},\, 2^{K_A} - 2w2^{K_B-1},\, 2w\big(2^{K_A-1} - w2^{K_B-1}\big)\Big).$$

The first five terms are the original bounds used for $\mathrm{Com}_{\mathbf{G}}$, while the additional term appears in the compressed proof for $\mathrm{Com}_{\mathbf{G}^*}$: $\|\hat{A}^{(0)}_{\mathrm{bin}}\|_\infty \le 2^{K_A-1} - w2^{K_B-1},$.

Correctness of the compressed aggregated commitments. To prove verification step 20 of Vf holds for each of the u out of T honest signers, we start at the *exact* (that is, the uncompressed $\tilde{\mathbf{w}}_j$'s) verification equation and rewrite it in terms of each of the honest signer's commitments $(\mathbf{w}_{u,j})_{j\in[0,\kappa-1]}$:

$$\begin{aligned}
\tilde{\mathbf{w}}_0 :&= \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big)\tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) - \sum_{j=1}^{\kappa-1} x^j \tilde{\mathbf{w}}_j \\
\mathbf{0} &= \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1}(a_{j,i_j} + x\,\delta_{\ell_j,i_j})\Big)\tilde{\mathbf{t}}_i \\
&\quad - \Big(\mathbf{A}\sum_{u=0}^{T-1}\mathbf{z}'_u + \mathbf{B}\sum_{u=0}^{T-1}\mathbf{r}'_u + \sum_{u=0}^{T-1}(\mathbf{z}''_u + \mathbf{r}''_u)\Big) - \sum_{j=0}^{\kappa-1} x^j\tilde{\mathbf{w}}_j \\
&= \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1} x\delta_{\ell_j,i_j} + \sum_{j=0}^{\kappa-1} x^j p_{i,j}\Big)\tilde{\mathbf{t}}_i - \Big(\bar{\mathbf{A}}\sum_{u=0}^{T-1}\mathbf{z}_u + \bar{\mathbf{B}}\sum_{u=0}^{T-1}\mathbf{r}_u\Big) \\
&\quad - \sum_{j=1}^{\kappa-1} x^j \sum_{u=0}^{T-1}\mathbf{w}_{u,j} \\
&= \sum_{i=0}^{N-1}\Big(x^\kappa\delta_{\ell,i} + \sum_{j=0}^{\kappa-1} x^j p_{i,j}\Big)\tilde{\mathbf{t}}_i - \bar{\mathbf{A}}\sum_{u=0}^{T-1}\Big(x^\kappa\alpha_u\mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j\mathbf{y}_{u,j}\Big) \\
&\quad + \bar{\mathbf{B}}\sum_{u=0}^{T-1}\sum_{j=0}^{\kappa-1} x^j\mathbf{r}_{u,j} - \sum_{j=1}^{\kappa-1} x^j\sum_{u=0}^{T-1}\mathbf{w}_{u,j} \\
&= x^\kappa\tilde{\mathbf{t}}_\ell + \sum_{j=0}^{\kappa-1} x^j\sum_{i=0}^{N-1} p_{i,j}\tilde{\mathbf{t}}_i - x^\kappa\tilde{\mathbf{t}}_\ell \\
&\quad + \bar{\mathbf{A}}\sum_{u=0}^{T-1}\sum_{j=0}^{\kappa-1} x^j\mathbf{y}_{u,j} + \bar{\mathbf{B}}\sum_{u=0}^{T-1}\sum_{j=0}^{\kappa-1} x^j\mathbf{r}_{u,j} - \sum_{j=1}^{\kappa-1} x^j\sum_{u=0}^{T-1}\mathbf{w}_{u,j} \\
&= \sum_{j=0}^{\kappa-1} x^j\sum_{i=0}^{N-1} p_{i,j}\tilde{\mathbf{t}}_i + \sum_{j=0}^{\kappa-1} x^j\sum_{u=0}^{T-1}\Big(\bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j}\Big) - \sum_{j=1}^{\kappa-1} x^j\sum_{u=0}^{T-1}\mathbf{w}_{u,j} \\
&= \sum_{j=0}^{\kappa-1} x^j\Big(\sum_{i=0}^{N-1} p_{i,j}\tilde{\mathbf{t}}_i + \sum_{u=0}^{T-1}(\bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j})\Big) - \sum_{j=1}^{\kappa-1} x^j\sum_{u=0}^{T-1}\mathbf{w}_{u,j} \\
&= \sum_{j=0}^{\kappa-1} x^j\sum_{u=0}^{T-1}\Big(\sum_{i=0}^{N-1} p_{i,j}\alpha_u\mathbf{t}_{u,i} + (\bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j})\Big) - \sum_{j=1}^{\kappa-1} x^j\sum_{u=0}^{T-1}\mathbf{w}_{u,j},
\end{aligned} \tag{1}$$

where $\alpha_u := H_{\mathrm{agg}}(\mathsf{PK}, u)$ and, indeed for each signer u, their commitments are $\mathbf{w}_{u,j} := \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \sum_{i=0}^{N-1} p_{i,j}\alpha_u\mathbf{t}_{u,i}$ for all $j \in [0, \kappa-1]$.

Event E_w. For each signer-side transmitted aggregated commitment $\tilde{\mathbf{w}}_j$ with $j \in [1, \kappa-1]$, write the coefficient-wise centered decomposition

$$\tilde{\mathbf{w}}_j = 2^{K_{w,j}}\tilde{\mathbf{w}}^{(1)}_j + \tilde{\mathbf{w}}^{(0)}_j, \qquad \|\tilde{\mathbf{w}}^{(0)}_j\|_\infty \le 2^{K_{w,j}-1}.$$

For an honestly generated signature, the exact verification identity is (1)

$$\tilde{\mathbf{w}}_0 = \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big)\tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) - \sum_{j=1}^{\kappa-1} x^j\tilde{\mathbf{w}}_j.$$

Hence the verifier-side approximate reconstruction, using only the compressed $\tilde{\mathbf{w}}^{(1)}_j$'s, is

$$\widehat{\mathbf{w}}_0 := \sum_{i=0}^{N-1}\Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big)\tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) - \sum_{j=1}^{\kappa-1} x^j\, 2^{K_{w,j}}\tilde{\mathbf{w}}^{(1)}_j = \tilde{\mathbf{w}}_0 + \Delta_w,$$

where

$$\Delta_w := \sum_{j=1}^{\kappa-1} x^j\tilde{\mathbf{w}}^{(0)}_j.$$

Using $\|ab\|_\infty \le \|a\|_1\|b\|_\infty$ and the rough bound $\|x^j\|_1 \le w^j$, we obtain

$$\|\Delta_w\|_\infty \le \sum_{j=1}^{\kappa-1}\|x^j\|_1\,\|\tilde{\mathbf{w}}^{(0)}_j\|_\infty \le \sum_{j=1}^{\kappa-1} w^j\, 2^{K_{w,j}-1} =: M_w.$$

Now write the coefficient-wise centered decompositions modulo $2^{K_{w,0}}$:

$$\tilde{\mathbf{w}}_0 = 2^{K_{w,0}}\tilde{\mathbf{w}}^{(1)}_0 + \tilde{\mathbf{w}}^{(0)}_0, \qquad \widehat{\mathbf{w}}_0 = 2^{K_{w,0}}\widehat{\mathbf{w}}^{(1)}_0 + \widehat{\mathbf{w}}^{(0)}_0.$$

The relevant correctness condition is

$$\tilde{\mathbf{w}}^{(1)}_0 = \widehat{\mathbf{w}}^{(1)}_0,$$

since a low-order perturbation can still create carries across a $2^{K_{w,0}}$-boundary.

Accordingly, we define the signer-side bad event

$$E_w := \Big\{\big\|\tilde{\mathbf{w}}^{(0)}_0\big\|_\infty > 2^{K_{w,0}-1} - M_w\Big\},$$

and the signer restarts whenever E_w occurs. If E_w does not occur, then every coefficient of $\tilde{\mathbf{w}}_0$ lies at distance at least M_w from the nearest $2^{K_{w,0}}$-boundary, so adding Δ_w cannot change the decomposition quotient. Hence, for every accepted signature,

$$\tilde{\mathbf{w}}^{(1)}_0 = \widehat{\mathbf{w}}^{(1)}_0.$$

Under the standard heuristic that the centered remainders of $\tilde{\mathbf{w}}_0$ modulo $2^{K_{w,0}}$ are close to uniform and weakly independent coefficient-wise, the acceptance probability of this extra restart step is approximately

$$p_w := \Pr[\neg E_w] \approx \left(\frac{2^{K_{w,0}} - 2M_w + 1}{2^{K_{w,0}}}\right)^{kd} \approx \left(1 - \frac{M_w}{2^{K_{w,0}-1}}\right)^{kd},$$

and therefore the corresponding extra repetition factor is

$$\mu_w := \frac{1}{p_w} \approx \left(1 - \frac{M_w}{2^{K_{w,0}-1}}\right)^{-kd} \approx \exp\left(\frac{kd\,M_w}{2^{K_{w,0}-1}}\right).$$

Correctness of the compressed binary commitment. For the binary subproof, write the coefficient-wise centered decompositions

$$B_{\text{bin}} = 2^{K_B} B_{\text{bin}}^{(1)} + B_{\text{bin}}^{(0)}, \qquad A_{\text{bin}} = 2^{K_A} A_{\text{bin}}^{(1)} + A_{\text{bin}}^{(0)},$$

with $\|B_{\text{bin}}^{(0)}\|_\infty \le 2^{K_B-1}$, $\|A_{\text{bin}}^{(0)}\|_\infty \le 2^{K_A-1}$. The signer computes

$$x = H\big(\mu,\, A_{\text{bin}}^{(1)},\, B_{\text{bin}}^{(1)},\, \tilde{\mathbf{w}}_0^{(1)}, \tilde{\mathbf{w}}_1^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)},\, \mathsf{PK}\big),$$

and then forms

$$\mathbf{z}_b := \mathbf{r}_a + x\mathbf{r}_b, \qquad f_{j,i} := x\delta_{\ell_j,i} + a_{j,i}, \qquad g_{j,i} := f_{j,i}(x - f_{j,i}).$$

As in the uncompressed case, these satisfy the exact identity

$$\mathbf{G}(\mathbf{z}_b, \mathbf{f}, \mathbf{g})^\top = A_{\text{bin}} + xB_{\text{bin}}.$$

Now define the verifier-side reconstruction

$$\hat{A}_{\text{bin}} := \mathbf{G}(\mathbf{z}_b, \mathbf{f}, \mathbf{g})^\top - x\, 2^{K_B} B_{\text{bin}}^{(1)}.$$

Substituting the decomposition of B_{bin}, we obtain

$$\hat{A}_{\text{bin}} = A_{\text{bin}} + xB_{\text{bin}}^{(0)}.$$

Since every challenge x satisfies $\|x\|_1 = w$, it follows that

$$\|xB_{\text{bin}}^{(0)}\|_\infty \le \|x\|_1\, \|B_{\text{bin}}^{(0)}\|_\infty \le w\, 2^{K_B-1}.$$

Let $\hat{A}_{\text{bin}} = 2^{K_A}\hat{A}_{\text{bin}}^{(1)} + \hat{A}_{\text{bin}}^{(0)}$ be its coefficient-wise centered decomposition modulo 2^{K_A}. Assuming the check

$$\|\hat{A}_{\text{bin}}^{(0)}\|_\infty \le 2^{K_A-1} - w\, 2^{K_B-1}$$

passes, we have

$$A_{\text{bin}} = \hat{A}_{\text{bin}} - xB_{\text{bin}}^{(0)} = 2^{K_A}\hat{A}_{\text{bin}}^{(1)} + \big(\hat{A}_{\text{bin}}^{(0)} - xB_{\text{bin}}^{(0)}\big),$$

and

$$\|\hat{A}_{\text{bin}}^{(0)} - xB_{\text{bin}}^{(0)}\|_\infty \le \|\hat{A}_{\text{bin}}^{(0)}\|_\infty + \|xB_{\text{bin}}^{(0)}\|_\infty \le 2^{K_A-1}.$$

Hence the last term is still a valid centered remainder modulo 2^{K_A}, so by uniqueness of the coefficient-wise centered decomposition,

$$\hat{A}_{\text{bin}}^{(1)} = A_{\text{bin}}^{(1)}.$$

Therefore the verifier recovers exactly the same high-bit value of $A_{\text{bin}}^{(1)}$ that the signer used in the Fiat–Shamir hash. □

D Proofs of Lemmas

D.1 Proof of Lemma 5.2

Proof. The reduction simply guesses an index $i^* \in [Q_h]$ that is intended to be the adversary's eventual target H_{com} query, and succeeds whenever this guess is correct. □

D.2 Proof of Lemma 5.3

Claim 1. Fix any session inputs $(\ell, \mu, \mathsf{PK}; \rho)$ and signer index $u \in [0, T-1]$. For each $j \in [0, \kappa-1]$, define the (possibly j-dependent) Gaussian widths

$$\sigma_s := \begin{cases} \phi_0 B_0 & \text{if } j = 0, \\ \phi_s B_s & \text{if } j \in [1, \kappa-1], \end{cases} \qquad \sigma_{s'} := \begin{cases} \phi_{0'} B_{0'} & \text{if } j = 0, \\ \phi_{s'} B_{s'} & \text{if } j \in [1, \kappa-1]. \end{cases}$$

Let $\mathbf{A} \in \mathcal{R}_q^{k\times l}$ be the matrix generated during Setup and define $\bar{\mathbf{A}} := [\mathbf{A} \mid \mathbf{I}] \in \mathcal{R}_q^{k\times(l+k)}$. Let $\mathbf{B} := H_{\text{com}}(\mathsf{PK}, \mu) \in \mathcal{R}_q^{k\times l'}$ and set $\bar{\mathbf{B}} := [\mathbf{B} \mid \mathbf{I}] \in \mathcal{R}_q^{k\times(l'+k)}$.

Table 5: Parameters of LoTRS.

Parameter(s)	Description
$B_a = \sqrt{\kappa w}$	Base mask scale for $\delta_{\ell_j,i}$ masks $a_{j,i}$.
ϕ_a	Rej. sampling slack factor for $\mathbf{f}_1$.
$\mu(\phi_a) = \exp(12/\phi_a + 1/(2\phi_a^2))$	Restart factor due to $\mathbf{f}_1$ rej. sampling.
$B_b = \sqrt{d(\hat{n}+\hat{k})w}$	Base scale for $\mathbf{z}_b$.
ϕ_b	Rej. sampling slack factor for $\mathbf{z}_b$.
$\mu(\phi_b) = \exp(1/(2\phi_b^2))$	Restart factor due to $\mathbf{z}_b$ rej. sampling.
$\mu_{BG} = \exp(\hat{n}d\, \frac{w2^{K_B-1}}{2^{K_A}})$	Restart factor due to $\hat{A}_{\text{bin}}^{(0)}$ check.
$\mu_w \approx \exp\!\left(\frac{kdM_w}{2^{K_{w,0}-1}}\right)$	Restart factor due to $2^{K_{w,0}-1} - M_w$ check.
$B_{g0}, B_{g1} = O(\phi_a^2 B_a^2)$	Max. bound for $g_0 = f_0(x-f_0)$, $g_1 = (f_j(x-f_j))_{j\ge1}$.
$B_z = 1.2\phi_0 B_0 \sqrt{dl}$	Max. ℓ_2 bound for response $\|\mathbf{z}_1'\|$.
$B_{\tilde{z}} = \sqrt{T} B_z$	Max. ℓ_2 bound for agg. response $\|\tilde{\mathbf{z}}\|$.
$B_r = 1.2\phi_0' B_0' \sqrt{dl'}$	Max. ℓ_2 bound for response $\|\mathbf{r}_1'\|$.
$B_{\tilde{r}} = \sqrt{T} B_r$	Max. ℓ_2 bound for agg. response $\|\tilde{\mathbf{r}}\|$.
$B_e = 1.2\phi_{\hat{0}} B_{\hat{0}} \sqrt{dk}$	Max. ℓ_2 bound for error $\|\tilde{\mathbf{e}}\|$.
$B_{\tilde{e}} = \sqrt{T} B_e$	Max. ℓ_2 bound for agg. error $\|\mathbf{e}_1\|$.
$B_{\eta,w} = 2^{K_w-1}\kappa w^{\kappa-1}$	Max. verifier-side residual.
η, η'	Max. ℓ_∞-norm of the secrets $\mathbf{s}$, $\mathbf{u}$ resp.
$S_\eta = \{v \in \mathcal{R} : \|v\|_\infty \le \eta\}$	Key set.
$S_{\eta'} = \{v \in \mathcal{R} : \|v\|_\infty \le \eta'\}$	Dual key set.
$B_s, B_{s'}$	Max. ℓ_2-norm of the non-masking $\mathbf{y}_{u,j}, \mathbf{r}_{u,j}$.
$\phi_s, \phi_{s'}$	Factor for $\mathbf{y}_{u,j}, \mathbf{r}_{u,j}$
B_0	Max. ℓ_2-norm of $x^\kappa \alpha_u \mathbf{s}_u + \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j}$.
$B_{0'}$	Max. ℓ_2-norm of $x^\kappa \alpha_u \bar{\mathbf{u}} + \sum_{j=0}^{\kappa-1} x^j \mathbf{p}_{u,j}$.
$\phi_0 = \phi_{0'} = \phi$	Rej. sampling slack factor for $\mathbf{z}_u$, $\mathbf{r}_u$.
$\mu(\phi) = \exp(12/\phi + 1/(2\phi^2))$	Restart factor for a single signer due to $\mathbf{z}_u$ rej. sampling.
$\mu(\phi)^T$	Restart factor from the T independent signer-local rejections in Sign_2.
$B_{\text{det}} = (2w)^{\frac{\kappa(\kappa+1)}{2}}$	Bound on $\det(\mathbf{V}_x)$ required for AM-SIS hardness.
$B_\Gamma = (2w)^{\frac{\kappa(\kappa-1)}{2}}$	Bound on Γ_t elements from $\text{adj}(\mathbf{V}_x)$

For each column index $i \in [0, N-1]$ in PK, define $\alpha_u := H_{\text{agg}}(\mathsf{PK}, u)$. Let coefficients $p_{i,j}$'s denote the ring elements computed deterministically from (ρ, ℓ), and hence depending on the expanded $(a_{j,i})$ and on ℓ as specified in the construction. Define the deterministic offset vector

$$\mathbf{v}_{u,j} := \sum_{i=0}^{N-1} p_{i,j}\, \alpha_u\, \mathbf{t}_{u,i} \in \mathcal{R}_q^k.$$

Consider the following two experiments:
Experiment $\text{Trans}(j)$. Sample $\mathbf{y}_{u,j} \leftarrow D_{\sigma_s}^{l+k}$ and $\mathbf{r}_{u,j} \leftarrow D_{\sigma_{s'}}^{l'+k}$ independently, and output

$$\mathbf{w}_{u,j} := \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \mathbf{v}_{u,j} \in \mathcal{R}_q^k.$$

Experiment $\mathsf{Sim}(j)$. Sample $\mathbf{z}_{u,j} \leftarrow D_{\sigma_s}^{l+k}$ and $\mathbf{p}_{u,j} \leftarrow D_{\sigma_{s'}}^{l'+k}$ independently, and output

$$\mathbf{w}_{u,j}^{\mathsf{sim}} \;:=\; \bar{\mathbf{A}}\mathbf{z}_{u,j} \;+\; \bar{\mathbf{B}}\mathbf{p}_{u,j} \;+\; \mathbf{v}_{u,j} \;\in\; \mathcal{R}_q^k.$$

Then the distributions of $\mathbf{w}_{u,j}$ and $\mathbf{w}_{u,j}^{\mathsf{sim}}$ are identical. Consequently, the distributions of the full first-round commitment vectors $\mathsf{Com}_u = (\mathbf{w}_{u,j})_{j\in[0,\kappa-1]}$ and $\mathsf{Com}_u^{\mathsf{sim}} = (\mathbf{w}_{u,j}^{\mathsf{sim}})_{j\in[0,\kappa-1]}$ are identical as well.

Assuming the regularity conditions of Lemma A.8 hold for $\bar{\mathbf{B}}$ with width $\sigma_{s'}$ and for $\bar{\mathbf{A}}$ with width σ_s, the random variable $\mathbf{w}_{u,j}$ is statistically close to uniform over $\mathcal{R}_q^k$ (up to negligible distance).

The first statement is immediate: in $\mathsf{Trans}(j)$, the pair $(\mathbf{y}_{u,j}, \mathbf{r}_{u,j})$ is distributed identically to the pair $(\mathbf{z}_{u,j}, \mathbf{p}_{u,j})$ in $\mathsf{Sim}(j)$ by construction, and $\mathbf{w}_{u,j}$ (resp. $\mathbf{w}_{u,j}^{\mathsf{sim}}$) is a deterministic function of that pair together with the fixed values $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{v}_{u,j})$.

We additionally record the near uniformity of $\mathbf{w}_{u,j}$, since it will be used when we later analyze conditional distributions such as $\mathbf{y}_{u,j} \mid \mathbf{w}_{u,j}$ and $\mathbf{r}_{u,j} \mid \mathbf{w}_{u,j}$.

Fix any value $\mathbf{y}^* \in \mathcal{R}^{l+k}$ and consider the conditional distribution of $\mathbf{w}_{u,j}$ given $\mathbf{y}_{u,j} = \mathbf{y}^*$ in $\mathsf{Trans}(j)$. From the definition,

$$\mathbf{w}_{u,j} \;=\; \bar{\mathbf{B}}\mathbf{r}_{u,j} \;+\; \underbrace{\left(\bar{\mathbf{A}}\mathbf{y}^* + \mathbf{v}_{u,j}\right)}_{=:\,\mathbf{d}^*}$$

where $\mathbf{d}^* \in \mathcal{R}_q^k$ is a fixed offset under this conditioning. Thus, the distribution of $\mathbf{w}_{u,j} \mid (\mathbf{y}_{u,j} = \mathbf{y}^*)$ is exactly the distribution of $\bar{\mathbf{B}}\mathbf{r}_{u,j}$ shifted by a constant.

By Lemma A.8, under the stated parameter condition on $\sigma_{s'}$, with overwhelming probability over the choice of $\mathbf{B}$, the distribution of $\bar{\mathbf{B}}\mathbf{r}_{u,j} \in \mathcal{R}_q^k$, for $\mathbf{r}_{u,j} \leftarrow D_{\sigma_{s'}}^{l'+k}$) is within negligible statistical distance of uniform over $\mathcal{R}_q^k$. Since adding a fixed $\mathbf{d}^*$ is a bijection on $\mathcal{R}_q^k$, the same negligible bound holds for $\mathbf{w}_{u,j} \mid (\mathbf{y}_{u,j} = \mathbf{y}^*)$. Averaging over $\mathbf{y}_{u,j}$ does not increase statistical distance, so the unconditional distribution of $\mathbf{w}_{u,j}$ is also negligibly close to uniform.

An analogous argument applies in $\mathsf{Sim}(j)$: conditioning on $\mathbf{p}_{u,j} = \mathbf{p}^*$ makes $\mathbf{w}_{u,j}^{\mathsf{sim}}$ a fixed shift of $\bar{\mathbf{A}}\mathbf{z}_{u,j}$, and Lemma A.8 applied to $\bar{\mathbf{A}}$ with width σ_s yields near uniformity.

Finally, note that the special masking indices $j = 0$ only increase the Gaussian widths (from $\phi_s B_s$ to $\phi_0 B_0$ and from $\phi_{s'} B_{s'}$ to $\phi_{0'} B_{0'}$), so the same regularity-based conclusion continues to hold provided the (larger) widths meet Lemma A.8's lower bound. □

Remark 5. *In our setting, the index ℓ (the column of* PK *containing the true signers' public keys) is a session parameter shared among the signers and is not revealed to the verifier. Our claim is stated conditioned on a fixed value of ℓ (together with μ,* PK*, ρ), and therefore remains valid regardless of whether ℓ is publicly known. Moreover, ℓ influences $\mathbf{w}_{u,j}$ only through the deterministic offset $\mathbf{v}_{u,j}$. Hence, under the regularity assumptions ensuring that $\bar{\mathbf{B}}\mathbf{r}_{u,j}$ (or $\bar{\mathbf{A}}\mathbf{y}_{u,j}$) is statistically close to uniform over $\mathcal{R}_q^k$, the distribution of $\mathbf{w}_{u,j}$ is negligibly close to uniform for every fixed ℓ.*

Claim 2. Let the same regularity conditions as in Claim 1 hold, so that each first round $\mathbf{w}_{u,j}$ is statistically close to uniform over $\mathcal{R}_q^k$. Fix any signer index u and any challenge $x \in C$, and define the effective secret-multiplying challenge coefficient

$$c_u \;:=\; \alpha_u x^\kappa \in \mathcal{R}.$$

Assume that the shift vectors passed to Rej are bounded as required, explicitly,

$$\left\| c_u\,\mathbf{s}_u - \textstyle\sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j} \right\| \;\le\; B_0 \quad \text{and} \quad \left\| c_u\,\bar{\mathbf{u}} - \textstyle\sum_{j=1}^{\kappa-1} x^j \mathbf{p}_{u,j} \right\| \;\le\; B_{0'}.$$

Let $\sigma_0 := \phi_0 B_0$ and $\sigma_{0'} := \phi_{0'} B_{0'}$ be the corresponding Gaussian widths used in Rej.

Then, in both Trans and Sim, for any fixed first round output out_1 produced in the same session, and conditioned on the event that $\mathsf{Sign}_{\mathsf{bin}}$ outputs π (so x is fixed), the probability that the second round does not abort satisfies

$$\left| \Pr[\mathsf{out}_2 \neq \perp \mid \mathsf{out}_1] - \frac{1}{\mu(\phi_*)} \right| \;\le\; 2^{-100} + 2^{-\Omega(d)}.$$

where $\phi_* = \phi_0$ in Trans (Rej applied to $\mathbf{z}_u$), and $\phi_* = \phi_{0'}$ in Sim (Rej_1 applied to $\mathbf{r}_u$).

First, we isolate the masked Gaussian $j = 0$ variable. In Trans,

$$\mathbf{z}_u := c_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j} = \underbrace{\left(c_u\,\mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}\right)}_{:=\mathbf{v}_u} - \mathbf{y}_{u,0} = \mathbf{y} + \mathbf{v}_u$$

Define $\mathbf{v}_u := c_u\,\mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}$, so that $\mathbf{z}_u = \mathbf{v}_u - \mathbf{y}_{u,0}$, and by symmetry of Gaussians, the random variable $\mathbf{y} := -\mathbf{y}_{u,0}$ is distributed identically to $\mathbf{y}_{u,0}$.

In Sim, the analogous decomposition holds for the response component that is rejection sampled:

$$\mathbf{r}_u := c_u\,\bar{\mathbf{u}} - \sum_{j=0}^{\kappa-1} x^j \mathbf{p}_{u,j} = \underbrace{\left(c_u\,\bar{\mathbf{u}} - \sum_{j=1}^{\kappa-1} x^j \mathbf{p}_{u,j}\right)}_{:=\mathbf{v}'_u} - \mathbf{p}_{u,0} = \mathbf{p} + \mathbf{v}'_u,$$

where $\mathbf{p} := -\mathbf{p}_{u,0}$ has the same distribution as $\mathbf{p}_{u,0}$.

Next, we condition on an arbitrary fixed first round transcript out_1 from the same signing session. By Claim (1), the commitments $\mathbf{w}_{u,0}$ are statistically close to uniform over $\mathcal{R}_q^k$, which implies that the conditional distribution of the hidden masking value $\mathbf{y}_{u,0}$ in Trans (resp. $\mathbf{p}_{u,0}$ in Sim) is within statistical distance $2^{-\Omega(d)}$ of its original discrete Gaussian distribution. Concretely, by a standard Bayes' Theorem argument,

$$\mathbf{y}_{u,0} \mid \mathsf{out}_1 \approx_{2^{-\Omega(d)}} D_{\sigma_0}^{l+k}, \qquad \mathbf{p}_{u,0} \mid \mathsf{out}_1 \approx_{2^{-\Omega(d)}} D_{\sigma_{0'}}^{l'+k}.$$

Note that the dependence on the other indices $j \geq 1$ is irrelevant here, since the randomness for different j's is sampled independently and $\mathbf{w}_{u,0}$ depends only on $(\mathbf{y}_{u,0}, \mathbf{r}_{u,0})$.

By our assumption on bounds, we have $\|\mathbf{v}_u\| \le B_0$ in Trans and $\|\mathbf{v}'_u\| \le B_{0'}$ in Sim. In other words, the shifts satisfy our conditions of Lemma A.4 with $K = B_0$ (resp. $K = B_{0'}$) and $\sigma = \phi K$.

Now fix any value of the shift $\mathbf{v}_u$ that is consistent with out_1 and the other sampled variables. Since up to $2^{-\Omega(d)}$ the masking variable is distributed as $\mathbf{y} \leftarrow D_{\sigma_0}^{l+k}$ and $\mathbf{z}_u = \mathbf{y} + \mathbf{v}_u$, the Lemma A.4 yields

$$\left| \Pr[\mathsf{Rej}(\mathbf{z}_u, \mathbf{v}_u, \phi_0, B_0) = 0 \mid \mathsf{out}_1] \;-\; \frac{1}{\mu(\phi_0)} \right| \;\le\; 2^{-100} + 2^{-\Omega(d)}.$$

Since $\mathsf{out}_2 \neq \perp$ in Trans is exactly the event $\mathsf{Rej}_1(\mathbf{z}_u, \mathbf{v}_u, \phi_0, B_0) = 0$, this gives the desired bound.

The same argument applies in Sim, with $\mathbf{r}_u = \mathbf{p} + \mathbf{v}'_u$ and the call to $\mathsf{Rej}_1(\mathbf{r}_u, \mathbf{v}'_u, \phi_{0'}, B_{0'})$, yielding the corresponding bound.

□

Remark 6 (Inner rejection sampling in $\mathsf{Sign}_{\mathsf{bin}}$). *Within* Trans *and* Sim*, the procedure* $\mathsf{Sign}_{\mathsf{bin}}$ *also perform the rejection sampling step* $\mathsf{Rej}(\mathbf{f}_1, x\mathbf{b}_1, \phi_a, B_a)$*. The same Lemma A.4 analysis applies provided* $\|x\mathbf{b}_1\| \le B_a$*. Since* $\mathsf{Sign}_{\mathsf{bin}}$ *has a shared output among all honest signers, this rejection event contributes a single restart factor per signing session (not per signer). Accordingly, Claim 2 should be interpreted as bounding the local success probability of* Rej *on* $\mathbf{z}_u$ *conditioned on* $\mathsf{Sign}_{\mathsf{bin}}$ *having succeeded, and hence conditioned on a fixed challenge* x*.*

Claim 3. Assume the hypotheses of Claim 3: regularity as in Claim 1, the same shift bounds needed to invoke Lemma A.4, and conditioning on the event that $\mathsf{Sign}_{\mathsf{bin}}$ outputs a proof π so that the challenge $x \in C$ is fixed, and set $c_u := \alpha_u x^\kappa \in \mathcal{R}$ for a fixed signer u. Furthermore, fix any first round transcript out_1 produced in this same signing session.

Let $\mathsf{out}_2^{\mathsf{Trans}}$ and $\mathsf{out}_2^{\mathsf{Sim}}$ denote the (possibly aborting) second-round outputs of Trans and Sim, respectively, viewed as random variables over the common output space

$$\mathcal{O} := \{\perp\} \cup \big(\mathcal{R}^{l+k} \times \mathcal{R}^{l'+k}\big),$$

where $\perp$ denotes abort.

Assume furthermore that both procedures use the same rejection parameter ϕ, i.e. $\phi_0 = \phi_{0'} =: \phi$, so that $\mu(\phi_0) = \mu(\phi_{0'}) =: \mu(\phi)$. Then, for every fixed (x, out_1),

$$\Delta\Big(\mathsf{out}_2^{\mathsf{Trans}} \mid (x, \mathsf{out}_1),\ \mathsf{out}_2^{\mathsf{Sim}} \mid (x, \mathsf{out}_1)\Big) \ \le\ \frac{3 \cdot 2^{-100}}{2\,\mu(\phi)} \ +\ 2^{-\Omega(d)}.$$

Fix (x, out_1) throughout. Set the session-dependent right-hand side target to be

$$\mathbf{d}_u := c_u\, \mathbf{t}_{u,\ell} - \sum_{j=0}^{\kappa-1} x^j \big(\mathbf{w}_{u,j} - \mathbf{v}_{u,j}\big) \ \in \mathcal{R}_q^k, \tag{2}$$

where $\mathbf{t}_{u,\ell}$ is the public-key column corresponding to signer u and index ℓ, and $(\mathbf{w}_{u,j})_j$ and $(\mathbf{v}_{u,j})_j$ are the first round values contained in out_1, as defined in Claim 1.

Define the solution set

$$\mathcal{S} \ :=\ \big\{(\mathbf{z}, \mathbf{r}) \in \mathcal{R}^{l+k} \times \mathcal{R}^{l'+k} :\ \bar{\mathbf{A}}\mathbf{z} + \bar{\mathbf{B}}\mathbf{r} = \mathbf{d}_u\big\}.$$

In both Trans and Sim, whenever the second round does not abort and outputs $(\mathbf{z}_u, \mathbf{r}_u)$, we have $(\mathbf{z}_u, \mathbf{r}_u) \in \mathcal{S}$.

Trans: By definition of the first round commitment,

$$\mathbf{w}_{u,j} = \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \mathbf{v}_{u,j} \quad \text{in } \mathcal{R}_q^k \qquad \forall j \in [0, \kappa-1].$$

Multiply by x^j and sum:

$$\sum_{j=0}^{\kappa-1} x^j (\mathbf{w}_{u,j} - \mathbf{v}_{u,j}) = \bar{\mathbf{A}} \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j} + \bar{\mathbf{B}} \sum_{j=0}^{\kappa-1} x^j \mathbf{r}_{u,j}.$$

In the second round the responses are

$$\mathbf{z}_u = c_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j}, \qquad \mathbf{r}_u = -\sum_{j=0}^{\kappa-1} x^j \mathbf{r}_{u,j}.$$

Using $\mathbf{t}_{u,\ell} = \bar{\mathbf{A}}\mathbf{s}_u$, we get

$$\begin{aligned}
\bar{\mathbf{A}}\mathbf{z}_u + \bar{\mathbf{B}}\mathbf{r}_u &= \bar{\mathbf{A}}\Big(c_u \mathbf{s}_u - \sum_j x^j \mathbf{y}_{u,j}\Big) + \bar{\mathbf{B}}\Big(-\sum_j x^j \mathbf{r}_{u,j}\Big) \\
&= c_u \bar{\mathbf{A}}\mathbf{s}_u - \Big(\bar{\mathbf{A}} \sum_j x^j \mathbf{y}_{u,j} + \bar{\mathbf{B}} \sum_j x^j \mathbf{r}_{u,j}\Big) \\
&= c_u \mathbf{t}_{u,\ell} - \sum_{j=0}^{\kappa-1} x^j (\mathbf{w}_{u,j} - \mathbf{v}_{u,j}) = \mathbf{d}_u,
\end{aligned}$$

so $(\mathbf{z}_u, \mathbf{r}_u) \in \mathcal{S}$.

Sim: The simulator constructs commitments so that

$$\mathbf{w}_{u,j} = \bar{\mathbf{A}}\mathbf{z}_{u,j} + \bar{\mathbf{B}}\mathbf{p}_{u,j} + \mathbf{v}_{u,j} \quad \text{in } \mathcal{R}_q^k \qquad \forall j.$$

Moreover, by the dual-key setup, it has a short $\bar{\mathbf{u}}$ with $\bar{\mathbf{B}}\bar{\mathbf{u}} = \mathbf{t}_{u,\ell}$. The simulator defines the second-round responses as

$$\mathbf{z}_u := -\sum_{j=0}^{\kappa-1} x^j \mathbf{z}_{u,j}, \qquad \mathbf{r}_u := c_u \bar{\mathbf{u}} - \sum_{j=0}^{\kappa-1} x^j \mathbf{p}_{u,j}.$$

Then

$$\begin{aligned}
\bar{\mathbf{A}}\mathbf{z}_u + \bar{\mathbf{B}}\mathbf{r}_u &= \bar{\mathbf{A}}\Big(-\sum_j x^j \mathbf{z}_{u,j}\Big) + \bar{\mathbf{B}}\Big(c_u \bar{\mathbf{u}} - \sum_j x^j \mathbf{p}_{u,j}\Big) \\
&= c_u \bar{\mathbf{B}}\bar{\mathbf{u}} - \sum_j x^j \big(\bar{\mathbf{A}}\mathbf{z}_{u,j} + \bar{\mathbf{B}}\mathbf{p}_{u,j}\big) \\
&= c_u \mathbf{t}_{u,\ell} - \sum_{j=0}^{\kappa-1} x^j (\mathbf{w}_{u,j} - \mathbf{v}_{u,j}) = \mathbf{d}_u,
\end{aligned}$$

so again $(\mathbf{z}_u, \mathbf{r}_u) \in \mathcal{S}$.

Next, we define a common probability mass function (PMF) P. Let $|\mathcal{R}_q| = q^d$, so $|\mathcal{R}_q^k| = q^{dk}$ and the uniform mass on $\mathcal{R}_q^k$ is q^{-dk}. Define the function $P : \mathcal{O} \to \mathbb{R}_{\ge 0}$ by

$$P(\perp) := 1 - \frac{1}{\mu(\phi)},$$

and for $(\mathbf{z}, \mathbf{r}) \in \mathcal{R}^{l+k} \times \mathcal{R}^{l'+k}$,

$$P(\mathbf{z}, \mathbf{r}) := \begin{cases} \dfrac{D_{\sigma_0}^{l+k}(\mathbf{z}) \cdot D_{\sigma_{0'}}^{l'+k}(\mathbf{r})}{\mu(\phi) \cdot q^{-dk}} & \text{if } (\mathbf{z}, \mathbf{r}) \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$$

For any random variable X over $\mathcal{O}$, write P_X for its probability distribution. Now we show that $P_{(\mathsf{out}_2^{\mathsf{Trans}} \mid (x, \mathsf{out}_1))}$ is close to P.

Let $T(\omega) := \Pr[\mathsf{out}_2^{\mathsf{Trans}} = \omega \mid (x, \mathsf{out}_1)]$ be the conditional PMF for Trans outputs.

Abort probability. By Claim 2,

$$\left| T(\perp) - \left(1 - \frac{1}{\mu(\phi)}\right) \right| \le 2^{-100} + 2^{-\Omega(d)}. \tag{3}$$

Non-abort probability. Fix any $(\mathbf{z}^*, \mathbf{r}^*) \in \mathcal{S}$. Let Acc_T denote the acceptance event of Rej in Trans. Then by the chain rule,

$$\begin{aligned}
T(\mathbf{z}^*, \mathbf{r}^*) = \ & \Pr\big[\mathbf{z}_u = \mathbf{z}^* \wedge \mathsf{Acc}_T \mid (x, \mathsf{out}_1)\big] \cdot \\
& \Pr\big[\mathbf{r}_u = \mathbf{r}^* \mid (x, \mathsf{out}_1), \mathbf{z}_u = \mathbf{z}^*, \mathsf{Acc}_T\big].
\end{aligned} \tag{4}$$

By Lemma A.4, in Claim 2, the accepted $\mathbf{z}_u$ in Trans is within $2^{-100} + 2^{-\Omega(d)}$ statistical distance of $D_{\sigma_0}^{l+k}$, and the acceptance probability is within $2^{-100} + 2^{-\Omega(d)}$ of $1/\mu(\phi)$. Using $\Pr[\mathbf{z} = \mathbf{z}^* \wedge \mathsf{Acc}_T] =$

$\Pr[\mathrm{Acc}] \cdot \Pr[\mathbf{z} = \mathbf{z}^* \mid \mathrm{Acc}]$, and applying the triangle inequality gives the following joint ℓ_1 error:

$$\sum_{\mathbf{z}\in\mathcal{R}^{l+k}} \left|\Pr[\mathbf{z}_u = \mathbf{z} \wedge \mathrm{Acc}_T \mid (x, \mathrm{out}_1)] - \frac{1}{\mu(\phi)} D_{\sigma_0}^{l+k}(\mathbf{z})\right| \le \frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}. \tag{5}$$

Next, conditioned on $(x, \mathrm{out}_1, \mathbf{z}_u = \mathbf{z}^*)$, (2) forces $\bar{\mathbf{B}}\mathbf{r}_u = \mathbf{d}_u - \bar{\mathbf{A}}\mathbf{z}^*$. Under the regularity assumption for $\bar{\mathbf{B}}$ at width $\sigma_{0'}$, for $\mathbf{r} \leftarrow D_{\sigma_{0'}}^{l'+k}$,

$$\Pr[\bar{\mathbf{B}}\mathbf{r} = \mathbf{d}_u - \bar{\mathbf{A}}\mathbf{z}^* \mid \mathbf{r} \leftarrow D_{\sigma_{0'}}^{l'+k}] = q^{-dk} \pm 2^{-\Omega(d)}. \tag{6}$$

Thus the conditional probability of $\mathbf{r}^*$ given the constraint satisfies

$$\Pr[\mathbf{r}_u = \mathbf{r}^* \mid (x, \mathrm{out}_1), \mathbf{z}_u = \mathbf{z}^*, \mathrm{Acc}_T] = \frac{D_{\sigma_{0'}}^{l'+k}(\mathbf{r}^*)}{q^{-dk}} \pm 2^{-\Omega(d)}. \tag{7}$$

Substituting (5) and (7) into (4) and summing over all $(\mathbf{z}, \mathbf{r}) \in \mathcal{S}$ yields

$$\sum_{(\mathbf{z},\mathbf{r})\in\mathcal{S}} \left|T(\mathbf{z}, \mathbf{r}) - P(\mathbf{z}, \mathbf{r})\right| \le \frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}. \tag{8}$$

Combining (3) and (8) gives

$$\sum_{\omega\in O} \left|T(\omega) - P(\omega)\right| \le \frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}. \tag{9}$$

Similarly, we can show that $P_{(\mathrm{out}_2^{\mathsf{Sim}}|(x,\mathrm{out}_1))}$ is close to P.

Let $S(\omega) := \Pr[\mathrm{out}_2^{\mathsf{Sim}} = \omega \mid (x, \mathrm{out}_1)]$ be the conditional PMF for Sim outputs. This is symmetric to Trans, with $(\bar{\mathbf{A}}, \mathbf{z}, \sigma_0)$ and $(\bar{\mathbf{B}}, \mathbf{r}, \sigma_{0'})$ swapped, because Rej in Sim is applied to $\mathbf{r}_u$.

Abort probability. By Claim (1),

$$\left|S(\perp) - \left(1 - \frac{1}{\mu(\phi)}\right)\right| \le 2^{-100} + 2^{-\Omega(d)}. \tag{10}$$

Non-abort probabilities. Arguing exactly as in (4)–(8), but now using Lemma A.4 on the $\mathbf{r}_u$-rejection and using regularity of $\bar{\mathbf{A}}$ to approximate $\Pr[\bar{\mathbf{A}}\mathbf{z} = \cdot]$ by q^{-dk}, we obtain

$$\sum_{(\mathbf{z},\mathbf{r})\in\mathcal{S}} \left|S(\mathbf{z}, \mathbf{r}) - P(\mathbf{z}, \mathbf{r})\right| \le \frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}. \tag{11}$$

Together with (10), this yields

$$\sum_{\omega\in\Omega} \left|S(\omega) - P(\omega)\right| \le \frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}. \tag{12}$$

Finally, we conclude by the triangle inequality. Recall that for distributions P, Q over O, the statistical distance between them is

$$\Delta(P, Q) \;:=\; \frac{1}{2}\sum_{\omega\in O} \left|P(\omega) - Q(\omega)\right|.$$

Therefore,

$$\Delta\left(P_{\mathrm{out}_2^{\mathsf{Trans}}|(x,\mathrm{out}_1)}, P_{\mathrm{out}_2^{\mathsf{Sim}}|(x,\mathrm{out}_1)}\right) \le \Delta(T, P) + \Delta(P, S) = \frac{1}{2}\sum_{\omega\in O} |T(\omega) - P(\omega)| + \frac{1}{2}\sum_{\omega\in O} |S(\omega) - P(\omega)|.$$

Using (9) and (12),

$$\Delta\left(P_{\mathrm{out}_2^{\mathsf{Trans}}|(x,\mathrm{out}_1)}, P_{\mathrm{out}_2^{\mathsf{Sim}}|(x,\mathrm{out}_1)}\right) \le 2\cdot\frac{1}{2}\left(\frac{3\cdot 2^{-100}}{\mu(\phi)} + 2^{-\Omega(d)}\right) = \frac{3\cdot 2^{-100}}{2\,\mu(\phi)} + 2^{-\Omega(d)}. \tag{13}$$

□

D.3 Proof of Lemma 5.4

Proof. We argue via a standard hybrid sequence of games G_0, G_1, $\mathrm{G}_{2,0}$, …, G_{2,Q_h}, and G_3 (Fig. 12–13), as in DualMS [8].

Game G_0. Let G_0 denote the original sel-UF-CMA experiment. Suppose $\mathcal{A}$ makes Q_h queries to OKGen polynomially many times Q_h. For one of these queries, respond with the honest key pair $(\mathsf{pk}, \mathsf{sk}) := (\mathbf{t}_{\hat{u}}^*, \mathbf{s}_{\hat{u}}^*)$.

Game G_1. Modify the way $H_{\mathrm{com}}(\mathsf{PK}, \mu)$ is answered on every new query except the designated i^*-th one. For these queries, respond with $\mathbf{B} = [\mathbf{b} \mid \hat{\mathbf{B}}]$, where $\hat{\mathbf{B}} \xleftarrow{\$} R_q^{k\times(l'-1)}$, $\mathbf{v} \xleftarrow{\$} R_q^k$ and $\mathbf{b} := \mathbf{t}_{\hat{u}}^* - \mathbf{v}$. Since $\mathbf{v}$ is uniform, so is $\mathbf{b}$, hence $\mathbf{B}$ is distributed exactly as a uniform matrix in $\mathcal{R}_q^{k\times\ell'}$. Therefore,

$$\Pr[\mathrm{G}_1(\mathcal{A}) = 1] = \Pr[\mathrm{G}_0(\mathcal{A}) = 1]. \tag{14}$$

Game $\mathrm{G}_{2,0}$. Same as G_1 but with an additional empty table $\mathcal{T}_{\mathbf{u}}$ initialized. So,

$$\Pr[\mathrm{G}_{2,0}(\mathcal{A}) = 1] = \Pr[\mathrm{G}_1(\mathcal{A}) = 1]. \tag{15}$$

Games $\mathrm{G}_{2,1}, \ldots, \mathrm{G}_{2,Q_h}$. For $j \in \{1, \ldots, Q_h\}$ with $j \neq i^*$, game $\mathrm{G}_{2,j}$ differs from $\mathrm{G}_{2,j-1}$ only on the j-th new H_{com} query. Instead of sampling $\mathbf{v}$ uniformly, sample $\mathbf{u} \xleftarrow{\$} S_{\eta'}^{\ell'-1+k}$ and set $\mathbf{v} := [\hat{\mathbf{B}} \mid \mathbf{I}]\mathbf{u}$, storing $\mathbf{u}$ in $\mathcal{T}_{\mathbf{u}}[\mathsf{PK}, \mu]$. The pair $(\hat{\mathbf{B}}, \mathbf{v})$ is a standard MLWE sample in $\mathrm{G}_{2,j}$ and uniform in $\mathrm{G}_{2,j-1}$, so a distinguisher against $\mathsf{MLWE}_{q,k,l'-1,\eta'}$ separates these games. Hence, for each such j,

$$\left|\Pr[\mathrm{G}_{2,j}(\mathcal{A}) = 1] - \Pr[\mathrm{G}_{2,j-1}(\mathcal{A}) = 1]\right| \le \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l'-1,\eta'}(\mathcal{D}),$$

and by a hybrid argument over all $j \neq i^*$,

$$\left|\Pr[\mathrm{G}_{2,Q_h}(\mathcal{A}) = 1] - \Pr[\mathrm{G}_1(\mathcal{A}) = 1]\right| \le (Q_h - 1)\,\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l'-1,\eta'}(\mathcal{D}). \tag{16}$$

Game G_3 *(Honest signing to straight-line simulation).* In G_{2,Q_h}, signing uses the honest secret keys from S_{hon} to sign. This remains true for all keys in S_{hon} in G_3, except for $\mathbf{t}_{\hat{u}}$. In G_3, the signing oracle instead uses the table $\mathcal{T}_{\mathbf{u}}$ to retrieve the corresponding $\mathbf{u} = \mathcal{T}_{\mathbf{u}}[\mathsf{PK}, \mu]$ for the current pair (PK, μ) and performs the dual straight-line simulation using $\bar{\mathbf{u}} = [1, \mathbf{u}^\top]^\top$. Unless $H_{\mathrm{com}}(\mathsf{PK}, \mu)$ is the i^*-th fresh query, this is possible since $\bar{\mathbf{B}}$ and $\bar{\mathbf{u}}$ will have the relation specified in Fig. 11.

By Lemma 5.3, this implies that the per-session output distribution from G_{2,Q_h} to G_3 changes by at most $3\cdot 2^{-100}/2\mu(\phi) + 2^{-\Omega(d)}$.

In DualMS, the input to H_{com} is $(\tilde{\mathbf{t}}_\ell, \mu)$, where $\tilde{\mathbf{t}}_\ell = \mathsf{KAgg}(\mathsf{PK}_\ell)$ is the aggregation of column ℓ in the table PK, call this PK_ℓ. Hence, two distinct lists $\mathsf{PK}_\ell \neq \mathsf{PK}'_\ell$ that collide under key aggregation (i.e., $\mathsf{KAgg}(\mathsf{PK}_\ell) = \mathsf{KAgg}(\mathsf{PK}'_\ell)$) induce the same H_{com} query for the same μ, and may allow a forgery to change the underlying list while keeping the same aggregate key. In our construction, H_{com} is

$\mathsf{Trans}(\mathbf{s}_u;\rho)$

Secret: $\mathbf{s}_u \in S_\eta^{l+k}$
Public: $\bar{\mathbf{A}} \coloneqq [\mathbf{A} \mid \mathbf{I}]$, $\mathbf{A} \in \mathcal{R}_q^{k\times l}$;
$\mathbf{t} \coloneqq \bar{\mathbf{A}}\mathbf{s}_u$; $\bar{\mathbf{B}} \coloneqq [\mathbf{b}|\hat{\mathbf{B}}|\mathbf{I}]$,
$\hat{\mathbf{B}} \in \mathcal{R}_q^{k\times(l'-1)}$, $\mathbf{b} \coloneqq \mathbf{t} - [\hat{\mathbf{B}}|\mathbf{I}]\mathbf{u}$
$(a_{j,i})_{j,\,i\in[1,\beta-1]} \coloneqq \mathsf{Expand}(\rho, \mathsf{a}; D_{\phi_a B_a}^{\kappa(\beta-1)})$
for $j = 0, \ldots, \kappa-1$ **do**
  Set $a_{j,0} \coloneqq -\sum_{i=1}^{\beta-1} a_{j,i}$
endfor
$\mathbf{y}_{u,1}, \ldots, \mathbf{y}_{u,\kappa-1} \xleftarrow{\$} D_{\phi_s B_s}^{l+k}$, $\mathbf{y}_{u,0} \xleftarrow{\$} D_{\phi_0 B_0}$
$\mathbf{r}_{u,1}, \ldots, \mathbf{r}_{u,\kappa-1} \xleftarrow{\$} D_{\phi'_s B_{s'}}^{l'+k}$, $\mathbf{r}_{u,0} \xleftarrow{\$} D_{\phi_{0'} B_{0'}}$
$\left(\mathbf{w}_{u,j} \coloneqq \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \sum_{i=0}^{N-1} p_{i,j}\alpha_u \mathbf{t}_{u,i}\right)_j$
$\mathsf{out}_1 \coloneqq (\mathbf{w}_{u,j})_j$
$x \leftarrow \mathsf{Sign}_{\mathsf{bin}}(\cdot)$
$\mathbf{z}_u \coloneqq x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j}$
$\mathbf{r}_u \coloneqq -\sum_{j=0}^{\kappa-1} x^j \mathbf{r}_{u,j}$
if $\mathsf{Rej}(\mathbf{z}_u, x^\kappa \alpha_u \mathbf{s} - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}, \phi_0, B_0)$ **then** $\mathsf{out}_2 \coloneqq \perp$
else $\mathsf{out}_2 \coloneqq (\mathbf{z}_u, \mathbf{r}_u)$
return $(\mathsf{out}_1, \mathsf{out}_2)$

$\mathsf{Sim}(\bar{\mathbf{u}};\rho)$

Secret: $\bar{\mathbf{u}} \in S_{\eta'}^{l'+k}$
Public: $\bar{\mathbf{A}} \coloneqq [\mathbf{A} \mid \mathbf{I}]$, $\mathbf{A} \in \mathcal{R}_q^{k\times l}$;
$\mathbf{t} \coloneqq \bar{\mathbf{A}}\mathbf{s}_u$; $\bar{\mathbf{B}} \coloneqq [\mathbf{b}|\hat{\mathbf{B}}|\mathbf{I}]$,
$\hat{\mathbf{B}} \in \mathcal{R}_q^{k\times(l'-1)}$, $\mathbf{b} \coloneqq \mathbf{t} - [\hat{\mathbf{B}}|\mathbf{I}]\mathbf{u}$
$(a_{j,i})_{j,\,i\in[1,\beta-1]} \coloneqq \mathsf{Expand}(\rho, \mathsf{a}; D_{\phi_a B_a}^{\kappa(\beta-1)})$
for $j = 0, \ldots, \kappa-1$ **do**
  Set $a_{j,0} \coloneqq -\sum_{i=1}^{\beta-1} a_{j,i}$
endfor
$\mathbf{z}_{u,1}, \ldots, \mathbf{z}_{u,\kappa-1} \xleftarrow{\$} D_{\phi_s B_s}^{l+k}$, $\mathbf{z}_{u,0} \xleftarrow{\$} D_{\phi_0 B_0}$
$\mathbf{p}_{u,1}, \ldots, \mathbf{p}_{u,\kappa-1} \xleftarrow{\$} D_{\phi'_s B_{s'}}^{l'+k}$, $\mathbf{p}_{u,0} \xleftarrow{\$} D_{\phi_{0'} B_{0'}}$
$\left(\mathbf{w}_{u,j} \coloneqq \bar{\mathbf{A}}\mathbf{z}_{u,j} + \bar{\mathbf{B}}\mathbf{p}_{u,j} + \sum_{i=0}^{N-1} p_{i,j}\alpha_u \mathbf{t}_{u,i}\right)_j$
$\mathsf{out}_1 \coloneqq (\mathbf{w}_{u,j})_j$
$x \leftarrow \mathsf{Sign}_{\mathsf{bin}}(\cdot)$
$\mathbf{r}_u \coloneqq x^\kappa \alpha_u \bar{\mathbf{u}} - \sum_{j=0}^{\kappa-1} x^j \mathbf{p}_{u,j}$
$\mathbf{z}_u \coloneqq -\sum_{j=0}^{\kappa-1} x^j \mathbf{z}_{u,j}$
if $\mathsf{Rej}(\mathbf{r}_u, x^\kappa \alpha_u \bar{\mathbf{u}} - \sum_{j=1}^{\kappa-1} x^j \mathbf{p}_{u,j}, \phi_{0'}, B_{0'})$ **then** $\mathsf{out}_2 \coloneqq \perp$
else $\mathsf{out}_2 \coloneqq (\mathbf{z}_u, \mathbf{r}_u)$
return $(\mathsf{out}_1, \mathsf{out}_2)$

Figure 11: Procedures Trans **and** Sim**. Note that** $j \in [0, \kappa-1]$ **for** $(\cdot)j$**.**

queried on the ring PK itself, so such aggregation collisions do not arise. We therefore do not account for such cases and so,

$$\left|\Pr[\mathrm{G}_{2,Q_h}(\mathcal{A})=1] - \Pr[\mathrm{G}_3(\mathcal{A})=1]\right| \le Q_s\Big(\frac{3\cdot 2^{-100}}{2\mu(\phi)} + 2^{-\Omega(d)}\Big). \tag{17}$$

Combining (16), and (17), yields the bound

$$\left|\Pr[\mathrm{G}_{2,Q_h}(\mathcal{A})=1] - \Pr[\mathrm{G}_5(\mathcal{A})=1]\right| \le (Q_h-1)\,\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l'-1,\eta'}(\mathcal{D}) + Q_s\Big(\frac{3\cdot 2^{-100}}{2\mu(\phi)} + 2^{-\Omega(d)} + \Big). \tag{18}$$

Notice that in G_3, the embedded secret $\mathbf{s}_{\hat{u}}$ is never used to answer the signing queries; it is only used to define $\mathbf{t}^*_{\hat{u}}$. Hence, a reduction $\mathcal{B}$ in the sel-UF-KOA game can perfectly simulate G_3 for $\mathcal{A}$. It answers $\mathcal{A}$'s random oracle queries to H, H_{agg} and the query to i^*-th using its own random oracle access, and answers $\mathcal{A}$'s signing queries by locally running the G_3 signing oracle algorithms (honest shares for keys in S_{hon} and simulated shares for $\mathbf{s}_{\hat{u}}$). Finally, it outputs the same i^*, $\tilde{\sigma}^*$ and PK^* as $\mathcal{A}$.

There is one subtlety to account for: $\mathcal{B}$ sees exactly a random function H in its unforgeability game, however, $\mathcal{A}$ sees a programmed oracle H', that "looks random". We need to make sure that replacing the truly random H by H', does not change $\mathcal{B}$'s advantage by a non-negligible amount. Conceptually, any oracle (including H') is implemented in the lazy sampling manner, i.e. if y has not been queried before, assign $H'(y)$ a uniformly random value from the challenge space. So, from A's perspective, H' appears identical to a random oracle, unless $\mathcal{A}$ queries some point y before, and later $\mathcal{B}$ attempts to reprogram $H'(y)$ to a different value. In this case, overwriting H' would no longer behave like an honest random oracle. The probability of such a collision event can easily be bounded by $Q_s\frac{Q_h}{|C|}$, where Q_s is the maximum number of signing sessions $\mathcal{A}$ launches. Consequently,

$$\mathsf{Adv}^{\text{sel-UF-CMA}}_{\mathsf{LoTRS}}(\mathcal{A}) \le \mathsf{Adv}^{\text{sel-UF-KOA}}_{\mathsf{LoTRS}}(\mathcal{B}) + (Q_h-1)\,\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l'-1,\eta'}(\mathcal{D}) + Q_s\Big(\frac{3\cdot 2^{-100}}{2\mu(\phi)} + 2^{-\Omega(N)} + \frac{Q_h}{|C|}\Big) \tag{19}$$

□

D.4 Proof of Lemma 5.5

Proof. We use the generic binary proof extraction argument from Theorem 1 of Esgin et al. [16], but applied to the compressed binary-proof first message. Let

$$\mathbf{G}^* := [\ \mathbf{G} \mid -\mathbf{I}_{\hat{n}}\ ]$$

and define the augmented commitment map

$$\mathsf{Com}_{\mathbf{G}^*}(\mathbf{u}, \mathbf{v}, \mathbf{e}; \mathbf{r}) := \mathbf{G}(\mathbf{r}, \mathbf{u}, \mathbf{v})^\top - \mathbf{e}.$$

We assume that $\mathsf{Com}_{\mathbf{G}^*}$ is γ^*_{com}-binding for all openings in the norm range that occur in this proof. Concretely, this means that, except with negligible probability, no PPT adversary can produce two distinct valid short openings in this range to the same commitment value under $\mathbf{G}^*$. The relevant range must cover:

- the honest compressed openings

  $$\big(\mathbf{b},\ \mathbf{a} \circ (\mathbf{1} - 2\mathbf{b}),\ B^{(0)}_{\mathsf{bin}};\ \mathbf{r}_b\big), \qquad \big(\mathbf{a},\ -\mathbf{a}^{\circ 2},\ A^{(0)}_{\mathsf{bin}};\ \mathbf{r}_a\big),$$

  where

  $$\|B^{(0)}_{\mathsf{bin}}\|_\infty \le 2^{K_B - 1}, \qquad \|A^{(0)}_{\mathsf{bin}}\|_\infty \le 2^{K_A - 1},$$

- and the extracted linear combinations

  $$(\hat{\mathbf{b}}, \hat{\mathbf{b}}_g, \hat{\mathbf{b}}_e; \hat{\mathbf{r}}_b), \qquad (\hat{\mathbf{a}}, \hat{\mathbf{a}}_g, \hat{\mathbf{a}}_e; \hat{\mathbf{r}}_a),$$

  defined below.

In particular, since acceptance enforces

$$\|\hat{A}^{(0)}_{\mathsf{bin}}\|_\infty \le 2^{K_A - 1} - w2^{K_B - 1},$$

the extracted remainder coordinates satisfy

$$\|\hat{\mathbf{b}}_e\|_\infty = \|\hat{A}^{(0)}_{\mathsf{bin}} - \hat{A}^{(0)\prime}_{\mathsf{bin}}\|_\infty \le 2^{K_A} - 2w2^{K_B - 1},$$

and

$$\|\hat{\mathbf{a}}_e\|_\infty = \|x\hat{A}^{(0)\prime}_{\mathsf{bin}} - x'\hat{A}^{(0)}_{\mathsf{bin}}\|_\infty \le 2w\big(2^{K_A - 1} - w2^{K_B - 1}\big).$$

Let $\mathsf{Com}_{\mathbf{G}}$ be the (additively homomorphic) HMC commitment used in $\mathsf{Sign}_{\mathsf{bin}}$. Let $\mathbf{a}, \mathbf{b}$ be the prover's masked witness vectors where $\mathbf{b}$ is intended to be binary (component-wise). The prover's exact first-message values are

$$B_{\mathsf{bin}} := \mathsf{Com}_{\mathbf{G}}(\mathbf{b}, \mathbf{a} \circ (\mathbf{1} - 2\mathbf{b});\ \mathbf{r}_b), \qquad A_{\mathsf{bin}} := \mathsf{Com}_{\mathbf{G}}(\mathbf{a}, -\mathbf{a}^{\circ 2};\ \mathbf{r}_a).$$

where "$\circ$" denotes component-wise Hadamard multiplication.

Write their coefficient-wise centered decompositions as

$$B_{\mathsf{bin}} = 2^{K_B} B^{(1)}_{\mathsf{bin}} + B^{(0)}_{\mathsf{bin}}, \qquad A_{\mathsf{bin}} = 2^{K_A} A^{(1)}_{\mathsf{bin}} + A^{(0)}_{\mathsf{bin}}.$$

Set

$$\bar{B}_{\mathsf{bin}} := 2^{K_B} B^{(1)}_{\mathsf{bin}}, \qquad \bar{A}_{\mathsf{bin}} := 2^{K_A} A^{(1)}_{\mathsf{bin}}.$$

Then

$$\bar{B}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}\big(\mathbf{b}, \mathbf{a} \circ (\mathbf{1} - 2\mathbf{b}), B^{(0)}_{\mathsf{bin}}; \mathbf{r}_b\big),$$

and

$$\bar{A}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}\big(\mathbf{a}, -\mathbf{a}^{\circ 2}, A^{(0)}_{\mathsf{bin}}; \mathbf{r}_a\big).$$

Upon challenge $x \in C$, the prover responds with

$$\mathbf{f} := x\mathbf{b} + \mathbf{a}, \qquad \mathbf{z}_b := x\mathbf{r}_b + \mathbf{r}_a,$$

and the verifier recomputes

$$\mathbf{g} := \mathbf{f} \circ (x\mathbf{1} - \mathbf{f}).$$

It also computes

$$\hat{A}_{\mathsf{bin}} := \mathsf{Com}_{\mathbf{G}}(\mathbf{f}, \mathbf{g}; \mathbf{z}_b) - x\,\bar{B}_{\mathsf{bin}},$$

and writes

$$\hat{A}_{\mathsf{bin}} = \bar{A}_{\mathsf{bin}} + \hat{A}^{(0)}_{\mathsf{bin}}$$

where $\hat{A}^{(0)}_{\mathsf{bin}}$ is the centered remainder modulo 2^{K_A}. Since three transcripts in the fork answer the same random-oracle query, they use the same high-bit value $A^{(1)}_{\mathsf{bin}}$, and hence the same $\bar{A}_{\mathsf{bin}}$. Therefore, acceptance is equivalent to the exact equation

$$x\,\bar{B}_{\mathsf{bin}} + \bar{A}_{\mathsf{bin}} \stackrel{?}{=} \mathsf{Com}_{\mathbf{G}^*}(\mathbf{f}, \mathbf{g}, \hat{A}^{(0)}_{\mathsf{bin}};\ \mathbf{z}_b). \tag{20}$$

Intuitively, since the left-hand side is linear in x, acceptance for 3 distinct challenges forces the x^2-coefficient of $\mathbf{g} = \mathbf{f} \circ (x\mathbf{1} - \mathbf{f})$ to vanish, which is exactly the "$\mathbf{b}(1 - \mathbf{b}) = 0$" condition (up to the usual ring subtleties).

Extraction from three accepting transcripts. Assume we have *three* accepting transcripts for the same public first-round data

$$(\bar{A}_{\mathsf{bin}}, \bar{B}_{\mathsf{bin}}, \tilde{\mathbf{w}}^{(1)}_0, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1})$$

but for three distinct challenges $x, x', x'' \in C$:

$$(x, \mathbf{f}, \mathbf{z}_b), \quad (x', \mathbf{f}', \mathbf{z}'_b), \quad (x'', \mathbf{f}'', \mathbf{z}''_b).$$

Let $\mathbf{g}, \mathbf{g}', \mathbf{g}''$ be recomputed from these transcripts by $\mathbf{g} = \mathbf{f} \circ (x\mathbf{1} - \mathbf{f})$ and similarly for $(x', \mathbf{f}')$, $(x'', \mathbf{f}'')$. Let $\hat{A}^{(0)}_{\mathsf{bin}}, \hat{A}^{(0)\prime}_{\mathsf{bin}}, \hat{A}^{(0)\prime\prime}_{\mathsf{bin}}$ be the corresponding verifier-side centered remainders.

By acceptance, we have

$$x\,\bar{B}_{\mathsf{bin}} + \bar{A}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}(\mathbf{f}, \mathbf{g}, \hat{A}^{(0)}_{\mathsf{bin}};\ \mathbf{z}_b), \tag{21}$$

$$x'\,\bar{B}_{\mathsf{bin}} + \bar{A}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}(\mathbf{f}', \mathbf{g}', \hat{A}^{(0)\prime}_{\mathsf{bin}};\ \mathbf{z}'_b), \tag{22}$$

$$x''\,\bar{B}_{\mathsf{bin}} + \bar{A}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}(\mathbf{f}'', \mathbf{g}'', \hat{A}^{(0)\prime\prime}_{\mathsf{bin}};\ \mathbf{z}''_b). \tag{23}$$

Subtracting (22) from (21) gives

$$(x - x')\,\bar{B}_{\mathsf{bin}} = \mathsf{Com}_{\mathbf{G}^*}(\mathbf{f} - \mathbf{f}',\ \mathbf{g} - \mathbf{g}',\ \hat{A}^{(0)}_{\mathsf{bin}} - \hat{A}^{(0)\prime}_{\mathsf{bin}};\ \mathbf{z}_b - \mathbf{z}'_b).$$

Let $y := x - x'$ and define the extracted opening of $y\bar{B}_{\mathsf{bin}}$ by

$$\begin{aligned} y\,\bar{B}_{\mathsf{bin}} &= \mathsf{Com}_{\mathbf{G}^*}(\hat{\mathbf{b}}, \hat{\mathbf{b}}_g, \hat{\mathbf{b}}_e;\ \hat{\mathbf{r}}_b), \quad \hat{\mathbf{b}} := \mathbf{f} - \mathbf{f}', \\ \hat{\mathbf{b}}_g &:= \mathbf{g} - \mathbf{g}', \quad \hat{\mathbf{b}}_e := \hat{A}^{(0)}_{\mathsf{bin}} - \hat{A}^{(0)\prime}_{\mathsf{bin}},\ \hat{\mathbf{r}}_b := \mathbf{z}_b - \mathbf{z}'_b. \end{aligned} \tag{24}$$

Multiplying (21) by y and substituting (24) yields an opening of $y\bar{A}_{\mathsf{bin}}$:

$$\begin{aligned} y\,\bar{A}_{\mathsf{bin}} &= \mathsf{Com}_{\mathbf{G}^*}(y\mathbf{f}, y\mathbf{g}, y\hat{A}^{(0)}_{\mathsf{bin}};\ y\mathbf{z}_b) - x \cdot \mathsf{Com}_{\mathbf{G}^*}(\hat{\mathbf{b}}, \hat{\mathbf{b}}_g, \hat{\mathbf{b}}_e;\ \hat{\mathbf{r}}_b) \\ &= \mathsf{Com}_{\mathbf{G}^*}(y\mathbf{f} - x\hat{\mathbf{b}},\ y\mathbf{g} - x\hat{\mathbf{b}}_g,\ y\hat{A}^{(0)}_{\mathsf{bin}} - x\hat{\mathbf{b}}_e;\ y\mathbf{z}_b - x\hat{\mathbf{r}}_b) \\ &=: \mathsf{Com}_{\mathbf{G}^*}(\hat{\mathbf{a}}, \hat{\mathbf{a}}_g, \hat{\mathbf{a}}_e;\ \hat{\mathbf{r}}_a), \end{aligned} \tag{25}$$

where explicitly

$$\begin{aligned} \hat{\mathbf{a}} &:= y\mathbf{f} - x\hat{\mathbf{b}} = x\mathbf{f}' - x'\mathbf{f}, \\ \hat{\mathbf{a}}_g &:= y\mathbf{g} - x\hat{\mathbf{b}}_g = x\mathbf{g}' - x'\mathbf{g}, \\ \hat{\mathbf{a}}_e &:= y\hat{A}^{(0)}_{\mathsf{bin}} - x\hat{\mathbf{b}}_e = x\hat{A}^{(0)\prime}_{\mathsf{bin}} - x'\hat{A}^{(0)}_{\mathsf{bin}}, \\ \hat{\mathbf{r}}_a &:= y\mathbf{z}_b - x\hat{\mathbf{r}}_b = x\mathbf{z}'_b - x'\mathbf{z}_b. \end{aligned}$$

From the definitions of $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, we record the key linear relation

$$y\,\mathbf{f} = x\,\hat{\mathbf{b}} + \hat{\mathbf{a}}. \tag{26}$$

Likewise, from the definitions of $\hat{\mathbf{a}}_g$ and $\hat{\mathbf{b}}_g$, we have

$$y\,\mathbf{g} = x\,\hat{\mathbf{b}}_g + \hat{\mathbf{a}}_g. \tag{27}$$

$\mathrm{G}_0(\mathcal{A})$–$\mathrm{G}_3(\mathcal{A})$

$\mathbf{A} \coloneqq \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k\times l})$
$\mathbf{s}_{\hat{u}} \xleftarrow{\$} S_\eta^{l+k},\ \mathbf{t}^*_{\hat{u}} \coloneqq \bar{\mathbf{A}}\mathbf{s}_{\hat{u}}$
$P \coloneqq \emptyset,\ P_{\mathrm{corr}} \coloneqq \emptyset, P^{\mu}_{\mathrm{sig}} \coloneqq \emptyset\ \forall$ messages μ
$\mathcal{T}_{\mathbf{u}} \coloneqq \emptyset$ / $\mathrm{G}_{2,0}$–G_3
$\mathsf{ctr} \coloneqq 0;\ \mathcal{S} \coloneqq \emptyset;\ \mathcal{Q} \coloneqq \emptyset$
$\mathsf{O} \coloneqq \{\mathsf{OKGen}, \mathsf{OCorr}, \mathsf{OReg},$
  $\mathsf{OSign}_1, \mathsf{OSign}_2, H_{\mathrm{agg}}, H_{\mathrm{com}}, H\}$
$(\mathsf{st}, i^*) \leftarrow \mathcal{A}_0(\mathsf{pp})$ / selective
$(\tilde{\sigma}^*, \mathsf{PK}^*) \leftarrow \mathcal{A}_1^{\mathsf{O}}(\mathsf{st})$
if PK^* is a $T \times N$ table
  $\wedge$ PK^* has distinct public keys
  $\wedge$ **for every column** $U \in \mathcal{F}_\Psi(\mathsf{PK}^*)$,
    $|U \cap (P_{\mathrm{corr}} \cup P^{\mu^*}_{\mathrm{sig}})| < T$
  $\wedge$ $(\mu^*, U, \mathsf{PK}^*) \notin \mathcal{Q}$ **then**
    return $\mathsf{Vf}(\mu^*, \tilde{\sigma}^*, \mathsf{PK}^*)$
return 0

$H_{\mathrm{com}}(\mathsf{PK}, \mu)$

if $\mathcal{T}_{\mathrm{com}}[\mathsf{PK}, \mu] \neq \perp$ **then**
  return $\mathcal{T}_{\mathrm{com}}[\mathsf{PK}, \mu]$
$\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{k\times l'}$ / G_0
if $|\mathcal{T}_{\mathrm{com}}| = i^* - 1$ **then** / G_1–G_3
  $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{k\times l'}$ / G_1–G_3
else
  $\hat{\mathbf{B}} \xleftarrow{\$} \mathcal{R}_q^{k\times(l'-1)}$ / G_1–G_3
  $\mathbf{v} \xleftarrow{\$} \mathcal{R}_q^{k}$ / G_1
  if $|\mathcal{T}_{\mathrm{com}}| < j$ **then** / $\mathrm{G}_{2,j}$–G_3
    $\mathbf{u} \xleftarrow{\$} S_{\eta'}^{l'-1+k}$ / $\mathrm{G}_{2,j}$–G_3
    $\mathcal{T}_{\mathbf{u}}[\mathsf{PK}, \mu] \coloneqq \mathbf{u}$ / $\mathrm{G}_{2,j}$–G_3
    $\mathbf{v} \coloneqq [\hat{\mathbf{B}} \mid \mathbf{I}]\,\mathbf{u}$ / $\mathrm{G}_{2,j}$–G_3
  else
    $\mathbf{v} \xleftarrow{\$} \mathcal{R}_q^{k}$ / $\mathrm{G}_{2,j}$–G_3
  $\mathbf{b} \coloneqq \mathbf{t}^*_{\hat{u}} - \mathbf{v}$ / G_1–G_3
  $\mathbf{B} \coloneqq [\mathbf{b} \mid \hat{\mathbf{B}}]$ / G_1–G_3
$\mathcal{T}_{\mathrm{com}}[\mathsf{PK}, \mu] \coloneqq \mathbf{B}$
if $|\mathcal{T}_{\mathrm{com}}| = i^*$ **then**
  $(\mathsf{PK}^*, \mu^*) \coloneqq (\mathsf{PK}, \mu)$ / selective
return $\mathcal{T}_{\mathrm{com}}[\mathsf{PK}, \mu]$

$\mathsf{OSign}_1(\mu, \ell, \mathsf{PK})$

Partition PK_ℓ into $S_{\mathsf{hon}} \sqcup S_{\mathrm{corr}}$
if $S_{\mathsf{hon}} = \emptyset$ **then return** $\perp$
$\mathcal{Q} \coloneqq \mathcal{Q} \cup \{(\mu, \mathsf{PK})\}$
$P^{\mu}_{\mathrm{sig}} \coloneqq P^{\mu}_{\mathrm{sig}} \cup \mathsf{PK}_\ell$
$\mathbf{B} \coloneqq H_{\mathrm{com}}(\mathsf{PK}, \mu)$
$\bar{\mathbf{B}} \coloneqq [\mathbf{B} \mid \mathbf{I}]$
for $\mathbf{t}_{u,\ell} \in S_{\mathsf{hon}}$ **do**
  $\mathsf{ctr} \coloneqq \mathsf{ctr} + 1$
  $\mathsf{sid} \coloneqq \mathsf{ctr};\ \mathcal{S} \coloneqq \mathcal{S} \cup \{\mathsf{sid}\}$
$(a_{j,i})_{j,\,i\in[1,\beta-1]} \coloneqq \mathsf{Expand}(\rho, \mathsf{a}|\mathsf{ctr}; D_{\phi_a B_a}^{\kappa(\beta-1)})$
Set $a_{j,0} \coloneqq -\sum_{i=1}^{\beta-1} a_{j,i} \quad \forall j$
  $\alpha_u \coloneqq H_{\mathrm{agg}}(\mathsf{PK}, u)$
  if $\mathbf{t}_{u,\ell} = \mathbf{t}^*_{\hat{u}}$ **then** / G_3
    $\mathbf{p}_{u,0} \xleftarrow{\$} D_{\phi_{0'} B_{0'}}$
    $(\mathbf{p}_{u,j})_{j\geq 1} \xleftarrow{\$} D_{s'}^{l'+k}$
    $\mathbf{z}_{u,0} \xleftarrow{\$} D_{\phi_y B_y}$
    $(\mathbf{z}_{u,j})_{j\geq 1} \xleftarrow{\$} D_s^{l+k}$
    $\left(\mathbf{w}_{u,j} \coloneqq \bar{\mathbf{A}}\mathbf{z}_u + \bar{\mathbf{B}}\mathbf{p}_{u,j} + \sum_{i=0}^{N-1} p_{i,j}\alpha_u \mathbf{t}_{u,i}\right)_j$
    $\mathsf{st}_{\mathsf{sid}} \coloneqq \left(\ell, \mu, \mathsf{PK}, \mathbf{t}_{u,\ell}, \alpha_u, (\mathbf{z}_{u,j})_j, (\mathbf{p}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\right)$
  else / G_0–$\mathrm{G}_{2,\mathcal{Q}_h}$
    $\mathbf{y}_{u,0} \xleftarrow{\$} D_{\phi_0 B_0}$
    $(\mathbf{y}_{u,j})_{j\geq 1} \xleftarrow{\$} D_{\phi_s B_s}^{l+k}$
    $\mathbf{r}_{u,0} \xleftarrow{\$} D_{\phi_{0'} B_{0'}}$
    $(\mathbf{r}_{u,j})_{j\geq 1} \xleftarrow{\$} D_{\phi'_s B_{s'}}^{l'+k}$
    $\left(\mathbf{w}_{u,j} \coloneqq \bar{\mathbf{A}}\mathbf{y}_{u,j} + \bar{\mathbf{B}}\mathbf{r}_{u,j} + \sum_{i=0}^{N-1} p_{i,j}\alpha_u \mathbf{t}_{u,i}\right)_j$
    $\mathsf{st}_{\mathsf{sid}} \coloneqq \left(\ell, \mu, \mathsf{PK}, \mathbf{t}_{u,\ell}, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\right)$
endfor
return $\left((\mathbf{w}_{u,j})_{j\in[0,\kappa-1]}\right)_{u\in|S_{\mathsf{hon}}|}$

Figure 12: Hybrid games G_0–G_3. Note that $j \in [0, \kappa - 1]$ for $(\cdot)j$.

By the γ^*_{com}-binding property of $\mathrm{Com}_{\mathrm{G}^*}$, except with negligible probability a PPT prover cannot provide two different valid short openings of the *same* committed values $y\bar{A}_{\mathsf{bin}}$ and $y\bar{B}_{\mathsf{bin}}$ across different rewinds. Hence, except with negligible probability, the same tuples $(\hat{\mathbf{b}}, \hat{\mathbf{b}}_g, \hat{\mathbf{b}}_e)$ and $(\hat{\mathbf{a}}, \hat{\mathbf{a}}_g, \hat{\mathbf{a}}_e)$ witness all three accepting transcripts, which implies that in addition to (26)–(27) we also have

$\mathrm{OSign}_2(\mathsf{sid}, \{\mathbf{w}_1, \ldots, \mathbf{w}_{T-1}\})$

1 : **if** $\mathsf{sid} \notin \mathcal{S}$ **then return** $\perp$
2 : $\mathcal{S} \coloneqq \mathcal{S} \setminus \{\mathsf{sid}\}$
3 : Extract $\mathbf{t}_{u,\ell}$ from $\mathsf{st}_{\mathsf{sid}}$
4 : **if** $\mathbf{t}_{u,\ell} = \mathbf{t}_{\hat{u}}$ **then** / G_3
5 : $\quad \left(\ell, \mu, \mathsf{PK}, \mathbf{t}_{u,\ell}, \alpha_u, (\mathbf{z}_{u,j})_j, (\mathbf{p}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\right) \coloneqq \mathsf{st}_{\mathsf{sid}}$
6 : **else** / G_0–G_{2,Q_h}
7 : $\quad \left(\ell, \mu, \mathsf{PK}, \mathbf{t}_{u,\ell}, \alpha_u, (\mathbf{y}_{u,j})_j, (\mathbf{r}_{u,j})_j, (\mathbf{w}_{u,j})_j; \rho\right) \coloneqq \mathsf{st}_{\mathsf{sid}}$
8 : $\left(\tilde{\mathbf{w}}_j \coloneqq \sum_{u'=0}^{T-1} \mathbf{w}_{u',j}\right)_j$
9 : **for** $j = 0, \ldots, \kappa - 1$ **do**
10 : $\quad$ Decompose $\tilde{\mathbf{w}}_j \coloneqq 2^{K_w} \tilde{\mathbf{w}}_j^{(1)} + \tilde{\mathbf{w}}_j^{(0)}$
11 : **endfor**
12 : $(B_{\mathsf{bin}}^{(1)}, (\tilde{\mathbf{w}}_1^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}), x, \mathbf{f}_1, \mathbf{z}_b) \coloneqq \mathsf{Sign}_{\mathsf{bin}}(\ell, \mu, (\tilde{\mathbf{w}}_j), \mathsf{PK}; \rho, \rho_G)$
13 : $\mathbf{z}_u \coloneqq x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=0}^{\kappa-1} x^j \mathbf{y}_{u,j} \in \mathcal{R}^{l+k}$ / G_0–G_{2,Q_h}
14 : $\mathbf{u} \coloneqq \mathcal{T}_{\mathbf{u}}[\mathsf{PK}, \mu]$ / G_3
15 : $\bar{\mathbf{u}} \coloneqq [1, \mathbf{u}^\top]^\top$ / G_3
16 : $\mathbf{r}_u \coloneqq \alpha_u x^\kappa \bar{\mathbf{u}} - \sum_{j=0}^{\kappa-1} x^j \mathbf{p}_{u,j}$ / G_3
17 : **if** $\mathsf{Rej}\left(\mathbf{z}_u, x^\kappa \alpha_u \mathbf{s}_u - \sum_{j=1}^{\kappa-1} x^j \mathbf{y}_{u,j}, \phi_0, B_0\right)$ / G_0–G_{2,Q_h}
18 : $\quad$ **then Restart**
19 : $\mathbf{r}_u \coloneqq - \sum_{j=0}^{\kappa-1} x^j \mathbf{r}_{u,j}$ / G_0–G_{2,Q_h}
20 : **if** $\mathsf{Rej}\left(\mathbf{r}_u, x^\kappa \alpha_u \bar{\mathbf{u}} - \sum_{j=1}^{\kappa-1} x^j \mathbf{p}_{u,j}, \phi_{0'}, B_{0'}\right)$ / G_3
21 : $\quad$ **then Restart**
22 : $\mathbf{z}_u \coloneqq - \sum_{j=0}^{\kappa-1} x^j \mathbf{z}_{u,j}$ / G_3
23 : **return** $\sigma_u \coloneqq (\pi_u, \mathbf{z}_u, \mathbf{r}_u)$

Figure 13: Hybrid games G_0–G_3. Note that $j \in [0, \kappa - 1]$ for $(\cdot)j$.

the analogous equalities for x' and x'':

$$y\,\mathbf{f}' = x'\hat{\mathbf{b}} + \hat{\mathbf{a}},\ y\,\mathbf{f}'' = x''\hat{\mathbf{b}} + \hat{\mathbf{a}},\ y\,\mathbf{g}' = x'\hat{\mathbf{b}}_g + \hat{\mathbf{a}}_g,\ y\,\mathbf{g}'' = x''\hat{\mathbf{b}}_g + \hat{\mathbf{a}}_g. \tag{28}$$

The extra coordinates $\hat{\mathbf{b}}_e$ and $\hat{\mathbf{a}}_e$ are only needed to make the extraction exact for the compressed first message. They do not enter the quadratic argument below.

Deriving the quadratic constraint $\hat{\mathbf{b}} \circ (y\mathbf{1} - \hat{\mathbf{b}}) = \mathbf{0}$. Recall that the verifier computes $\mathbf{g} = \mathbf{f} \circ (x\mathbf{1} - \mathbf{f})$ component-wise. Using (26) and the identity $y(x\mathbf{1} - \mathbf{f}) = (yx)\mathbf{1} - y\mathbf{f}$, we get

$$y(x\mathbf{1} - \mathbf{f}) = (yx)\mathbf{1} - (x\hat{\mathbf{b}} + \hat{\mathbf{a}}) = x\,(y\mathbf{1} - \hat{\mathbf{b}}) - \hat{\mathbf{a}}.$$

Now multiply (27) by y and substitute $\mathbf{g} = \mathbf{f} \circ (x\mathbf{1} - \mathbf{f})$:

$$y \cdot (x\hat{\mathbf{b}}_g + \hat{\mathbf{a}}_g) = y^2\,\mathbf{g} \tag{29}$$

$$\begin{aligned} &= (y\mathbf{f}) \circ (y(x\mathbf{1} - \mathbf{f})) \\ &= (x\hat{\mathbf{b}} + \hat{\mathbf{a}}) \circ \left(x(y\mathbf{1} - \hat{\mathbf{b}}) - \hat{\mathbf{a}}\right) \\ &= x^2\,\hat{\mathbf{b}} \circ (y\mathbf{1} - \hat{\mathbf{b}}) \ +\ x\left(\hat{\mathbf{a}} \circ (y\mathbf{1} - 2\hat{\mathbf{b}})\right) \ -\ \hat{\mathbf{a}}^{\circ 2}. \end{aligned} \tag{30}$$

Rearranging gives the degree-2 polynomial identity in x:

$$x^2 \cdot \hat{\mathbf{b}} \circ (y\mathbf{1} - \hat{\mathbf{b}}) \ +\ x \cdot \left(\hat{\mathbf{a}} \circ (y\mathbf{1} - 2\hat{\mathbf{b}}) - y\hat{\mathbf{b}}_g\right) \ -\ \hat{\mathbf{a}}^{\circ 2} - y\hat{\mathbf{a}}_g \ =\ \mathbf{0} \quad \text{in } \mathcal{R}_{\hat{q}}^{\kappa\beta}. \tag{31}$$

The same derivation applied to the accepting transcripts for x' and x'' yields two more copies of (31) with x replaced by x' and x''.

Equivalently, with the Vandermonde matrix $\mathbf{V}$,

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \begin{pmatrix} -\hat{\mathbf{a}}^{\circ 2} - y\hat{\mathbf{a}}_g \\ \hat{\mathbf{a}} \circ (y\mathbf{1} - 2\hat{\mathbf{b}}) - y\hat{\mathbf{b}}_g \\ \hat{\mathbf{b}} \circ (y\mathbf{1} - \hat{\mathbf{b}}) \end{pmatrix}.$$

By Lemma A.1, since the difference of any distinct $x, x', x'' \in \mathcal{R}_{\hat{q}}$ is invertible over the ring, $\mathbf{V}^{-1}$ exists, and we get

$$\hat{\mathbf{b}} \circ (y\mathbf{1} - \hat{\mathbf{b}}) = \mathbf{0}.$$

Since $y = x - x' \neq 0$, this forces each component $\hat{b}$ of $\hat{\mathbf{b}}$ to satisfy $\hat{b} \in \{0, y\}$, and therefore (writing $\hat{b} = y \cdot b$) we obtain $b \in \{0, 1\}$ component-wise. Finally, by (26), $\mathbf{f} = x\mathbf{b} + \mathbf{a}$ is indeed an encoding of the binary vector $\mathbf{b}$ (masked by $\mathbf{a}$), exactly as required.

From binarity to the selector polynomial. Assume we have established that for each $j \in [0, \kappa - 1]$ and $i \in [0, \beta - 1]$ there exist ring elements $a_{j,i} \in \mathcal{R}$ and bits $b_{j,i} \in \{0, 1\} \subseteq \mathcal{R}$ such that the verifier-side reconstructed values satisfy

$$f_{j,i} \;=\; a_{j,i} + x\, b_{j,i}. \tag{32}$$

Moreover, since the verifier reconstructs $f_{j,0}$ by $f_{j,0} := x - \sum_{i=1}^{\beta-1} f_{j,i}$ and we have $a_{j,0} := -\sum_{i=1}^{\beta-1} a_{j,i}$, it follows that

$$\sum_{i=0}^{\beta-1} b_{j,i} \;=\; 1 \qquad \text{for every } j \in [0, \kappa - 1]. \tag{33}$$

Together with $b_{j,i} \in \{0, 1\}$, this implies that in each block j there is a unique $\ell_j \in [0, \beta - 1]$ such that $b_{j,\ell_j} = 1$ and $b_{j,i} = 0$ for $i \neq \ell_j$. Define

$$\ell := \sum_{j=0}^{\kappa-1} \ell_j \beta^j \in [0, \beta^\kappa - 1] = [0, N - 1].$$

Degree-κ expansion. For each $i \in [0, N - 1]$, write its base-β digits as $i = \sum_{j=0}^{\kappa-1} i_j \beta^j$ with $i_j \in [0, \beta - 1]$, and define the product

$$P_i(x) \;:=\; \prod_{j=0}^{\kappa-1} f_{j,i_j}.$$

Using (32), we can view $p_i(\cdot)$ as the evaluation at x of the degree-κ polynomial

$$P_i(X) \;:=\; \prod_{j=0}^{\kappa-1} \big(a_{j,i_j} + X\, b_{j,i_j}\big) \in \mathcal{R}[X].$$

Expanding the product, we obtain

$$P_i(X) \;=\; X^\kappa \prod_{j=0}^{\kappa-1} b_{j,i_j} \;+\; \sum_{m=0}^{\kappa-1} E_{i,m}\, X^m \tag{34}$$

for some coefficients $E_{i,0}, \ldots, E_{i,\kappa-1} \in \mathcal{R}$ that depend only on (a_{j,i_j}) and (b_{j,i_j}) (and hence are independent of the challenge X). By the block-one-hot property of b, the leading coefficient satisfies

$$\prod_{j=0}^{\kappa-1} b_{j,i_j} \;=\; \begin{cases} 1 & \text{if } i_j = \ell_j \text{ for all } j \;\; (\text{i.e. } i = \ell), \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, (34) becomes the selector form

$$P_i(X) \;=\; \delta_{i,\ell}\, X^\kappa \;+\; \sum_{m=0}^{\kappa-1} E_{i,m}\, X^m, \tag{35}$$

where $\delta_{i,\ell} \in \{0, 1\}$ is the Kronecker delta.

Summing against the aggregated public keys. Let $(\tilde{\mathbf{t}}_i)_{i\in[0,N-1]}$ be the vectors output by KAgg(PK) (with $\tilde{\mathbf{t}}_i \in \mathcal{R}_q^k$). Define vector coefficients

$$\mathbf{C}_j \;:=\; \sum_{i=0}^{N-1} E_{i,j}\, \tilde{\mathbf{t}}_i \in \mathcal{R}_q^k \qquad \text{for } j = 0, \ldots, \kappa - 1.$$

Evaluating (35) at $X = x$ and summing over i yields

$$\sum_{i=0}^{N-1} \Big(\prod_{j=0}^{\kappa-1} f_{j,i_j}\Big) \tilde{\mathbf{t}}_i = \sum_{i=0}^{N-1} \left(\delta_{i,\ell} x^\kappa + \sum_{m=0}^{\kappa-1} E_{i,m} x^m\right) \tilde{\mathbf{t}}_i = x^\kappa \tilde{\mathbf{t}}_\ell + \sum_{m=0}^{\kappa-1} \mathbf{C}_m x^m,$$

which is exactly the desired identity.

Effect of compressing the aggregated commitments. Note that the compression of the $\tilde{\mathbf{w}}_j$'s does not alter the binary extraction itself: the exact selector-polynomial identity proved above depends only on the extracted binary structure and on the aggregated public keys $(\tilde{\mathbf{t}}_i)_{i\in[0,N-1]}$. The compression of the public commitments $\tilde{\mathbf{w}}_j$ affects only the later verifier equation, where each $\tilde{\mathbf{w}}_j$ is replaced by $2^{K_w}\tilde{\mathbf{w}}_j^{(1)}$ plus a bounded dropped-bit part $\tilde{\mathbf{w}}_j^{(0)}$. Accordingly, the selector identity above is unchanged, and the resulting compression discrepancy is handled separately in correctness and in the main unforgeability proof through the explicit residual term $\boldsymbol{\eta}_w := \sum_{j=1}^{\kappa-1} x^j \tilde{\mathbf{w}}_j^{(0)}$. □

Lemma D.1 (Adapted from [18], Corollary 3). *Suppose $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, $b \in R_q$, and Δ is invertible in R_q. Assume that b is binary in the canonical embedding, and that $\|\Delta b\|_\infty \leq 2B$. If*

$$q > d\big(\|\Delta\|_\infty + 2B\big)^2,$$

then b is uniquely determined and lies in $\{0, 1\}$ in the canonical embedding.

D.5 Proof of Lemma 5.6

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary in the sel-UF-CMA experiment that attempts to forge a LoTRS signature, and let $\mathcal{B}$ be an adversary against an MSIS instance $\mathsf{MSIS}_{q,k,1+l+l',\beta_{\mathsf{SIS}}}$ which takes as inputs $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{t}^*_{\hat{u}}$, uniformly distributed over $\mathcal{R}_q^{k\times l}, \mathcal{R}_q^{k\times l'}$, and $\mathcal{R}_q^k$, respectively. $\mathcal{B}$ plays the role of the challenger to $\mathcal{A}$ by running $\mathcal{A}$ on input $\bar{\mathbf{A}}$ and simulates all oracles with standard bookkeeping tables $\mathcal{T}_{\mathsf{com}}, \mathcal{T}_{\mathsf{agg}}, \mathcal{T}_{\mathsf{sig}}$. In particular,

- $\mathcal{B}$ answers the i^*-th query of $H_{\mathsf{com}}(\mathsf{PK}^*, \mu^*)$ with $\mathbf{B}$ and all others with a uniform element in $\mathcal{R}_q^{l'+k}$, and records it in $\mathcal{T}_{\mathsf{com}}$.
- $\mathcal{B}$ answers a fresh $H_{\mathsf{agg}}(\mathsf{PK}, u)$ query with a uniform $\alpha_u \xleftarrow{\$} C$ and records it in $\mathcal{T}_{\mathsf{agg}}$.
- $\mathcal{B}$ answers the Fiat-Shamir oracle $H(\mu, A_{\mathsf{bin}}^{(1)}, B_{\mathsf{bin}}^{(1)}, \tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK})$ with a uniform $x \xleftarrow{\$} C$, and records it in $\mathcal{T}_{\mathsf{sig}}$.

Additionally, $\mathcal{A}$ makes key generating queries polynomially many times Q_k to OKGen. $\mathcal{B}$ randomly chooses one of these queries, say the K-th one, and sets $\mathsf{pk}_K := \mathbf{t}^*_{\hat{u}}$. If $\mathcal{A}$ requests to corrupt key pk_K, $\mathcal{B}$ aborts. Suppose $\mathcal{A}$ outputs a valid forgery

$$\Big(\mathsf{PK}^*, \tilde{\sigma}^* := \big((B_{\mathsf{bin}}^{(1)}, (\tilde{\mathbf{w}}_1^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}), x, \mathbf{f}_1, \mathbf{z}_b), \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}\big)\Big).$$

If PK^* does not contain pk_K, then $\mathcal{B}$ aborts. Otherwise for pk_K in row u and column ℓ^*, $\mathcal{B}$ defines coefficients $\alpha_u := \mathcal{T}_{\mathsf{agg}}[\mathsf{PK}^*, u]$ and

$x^{(0)} := \mathcal{T}_{\text{sig}}[\mu^*, B^{(1)}_{\text{bin}}, A^{(1)}_{\text{bin}}, \tilde{\mathbf{w}}^{(1)}_0, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}, \mathsf{PK}^*]$, and then rewinds $\mathcal{A}$ multiple times on this particular execution to obtain a tree of accepting transcripts. More specifically, $\mathcal{B}$ regards $\mathcal{A}$ as a rewindable black-box and computes a $(2, \kappa+1)$-tree of accepting transcripts according to the Tree-Finder algorithm in Lemma 3.2 [11]. Using this, there will be two levels of branching with:

1-*rewind.* $\mathcal{B}$ rewinds until just before the query to $\mathcal{T}_{\text{agg}}[\mathsf{PK}^*, u]$, where the execution is identical until the point that this query is made. So, naturally PK^* and u will be the same across all $2(\kappa+1)$ transcripts. Therefore $\mathbf{B}$ obtains:

$$\alpha_u, \qquad \alpha'_u \tag{36}$$

κ-*rewinds.* Repeatedly rewinds $\mathcal{A}$ until just before the query to H and assigns a new randomly chosen x to assign $\mathcal{T}_{\text{sig}}$. So, $\mathcal{B}$ obtains $(\kappa+1)$ accepting transcripts (per α_u α'_u branch) that share the same first round data (per α_u α'_u branch) to obtain:

$$\alpha_u, \quad \left(x^{(t)}, \mathbf{f}^{(t)}_1, \mathbf{z}^{(t)}_b, \tilde{\mathbf{z}}^{(t)}, \tilde{\mathbf{r}}^{(t)}, \tilde{\mathbf{e}}^{(t)}\right)^{\kappa}_{t=0}, \quad \left(\mathsf{PK}^*, A^{(1)}_{\text{bin}}, B^{(1)}_{\text{bin}}, \tilde{\mathbf{w}}^{(1)}_0, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}\right) \tag{37}$$

$$\alpha'_u, \quad \left(x'^{(t)}, \mathbf{f}'^{(t)}_1, \mathbf{z}'^{(t)}_b, \tilde{\mathbf{z}}'^{(t)}, \tilde{\mathbf{r}}'^{(t)}, \tilde{\mathbf{e}}'^{(t)}\right)^{\kappa}_{t=0}, \quad \left(\mathsf{PK}^*, A'^{(1)}_{\text{bin}}, B'^{(1)}_{\text{bin}}, \tilde{\mathbf{w}}'^{(1)}_0, \ldots, \tilde{\mathbf{w}}'^{(1)}_{\kappa-1}\right) \tag{38}$$

By [11], $\mathcal{B}$ succeeds in obtaining these transcripts with probability at least $\frac{1}{4}$ in an expected time of at most:

$$O\left(\frac{\kappa+1}{\epsilon_{\mathcal{A}} - \epsilon_{\mathcal{D}}}\right) \tag{39}$$

where $\epsilon_{\mathcal{A}} := \mathsf{Adv}^{\mathsf{sel\text{-}UF\text{-}KOA}}_{\mathsf{LoTRS}}(\mathcal{A})$ is the probability that $\mathcal{A}$ outputs a valid forgery, and $\epsilon_{\mathcal{D}} := \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\eta}(\mathcal{D})$.

Extraction on α_u branch. Let us first focus on the α_u branch. Since these are $\kappa+1$ accepting transcripts, the selector-polynomial lemma remains exact. The compression of the aggregated commitments is handled via a proof-local bounded remainder term. For each transcript t, we have

$$\sum_{i=0}^{N-1}\left(\prod_{j=0}^{\kappa-1} f^{(t)}_{j,i_j}\right)\tilde{\mathbf{t}}_i = (x^{(t)})^{\kappa}\tilde{\mathbf{t}}_{\ell^\dagger} + \sum_{j=0}^{\kappa-1}(x^{(t)})^j \mathbf{C}_j.$$

Define the verifier-side approximate reconstruction

$$\widehat{\mathbf{w}}^{(t)}_0 := \sum_{i=0}^{N-1}\left(\prod_{j=0}^{\kappa-1} f^{(t)}_{j,i_j}\right)\tilde{\mathbf{t}}_i - (\mathbf{A}\tilde{\mathbf{z}}^{(t)} + \mathbf{B}\tilde{\mathbf{r}}^{(t)} + \tilde{\mathbf{e}}^{(t)}) - \sum_{j=1}^{\kappa-1}(x^{(t)})^j\, 2^{K_{w,j}}\tilde{\mathbf{w}}^{(1)}_j.$$

Since the transcript is accepting, the quotient part of $\widehat{\mathbf{w}}^{(t)}_0$ modulo $2^{K_{w,0}}$ equals the transmitted $\tilde{\mathbf{w}}^{(1)}_0$. Hence there exists a centered remainder $\boldsymbol{\eta}^{(t)}_w$ such that

$$\widehat{\mathbf{w}}^{(t)}_0 = 2^{K_{w,0}}\tilde{\mathbf{w}}^{(1)}_0 + \boldsymbol{\eta}^{(t)}_w, \qquad \|\boldsymbol{\eta}^{(t)}_w\|_\infty \le 2^{K_{w,0}-1}.$$

For convenience, define the proof-side bound $B_{\eta,w} := 2^{K_{w,0}-1}$. Our goal now, is to turn verification into a degree-κ polynomial identity. By Lemma 5.5, the verification for each accepting transcript can be written as

$$\mathbf{A}\tilde{\mathbf{z}}^{(t)} + \mathbf{B}\tilde{\mathbf{r}}^{(t)} + \tilde{\mathbf{e}}^{(t)} + \boldsymbol{\eta}^{(t)}_w = (x^{(t)})^{\kappa}\tilde{\mathbf{t}}_{\ell^\dagger} + \sum_{j=0}^{\kappa-1}(x^{(t)})^j \mathbf{C}'_j, \quad t = 0, \ldots, \kappa, \tag{40}$$

for some $\ell^\dagger \in \{0, \ldots, N-1\}$, fixed vectors $\mathbf{C}'_0, \ldots, \mathbf{C}'_{\kappa-1}$ where $\mathbf{C}'_0 := \mathbf{C}_0 - 2^{K_{w,0}}\tilde{\mathbf{w}}^{(1)}_0, \mathbf{C}'_j := \mathbf{C}_j - 2^{K_{w,j}}\tilde{\mathbf{w}}^{(1)}_j$ for $j = 1, \ldots, \kappa-1$ and $\|\boldsymbol{\eta}^{(t)}_w\|_\infty \le B_{\eta,w}$. Equivalently,

$$\begin{aligned}\begin{pmatrix} 1 & x^{(0)} & (x^{(0)})^2 & \cdots & (x^{(0)})^{\kappa} \\ 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{\kappa} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(\kappa)} & (x^{(\kappa)})^2 & \cdots & (x^{(\kappa)})^{\kappa} \end{pmatrix}\begin{pmatrix} \mathbf{C}'_0 \\ \mathbf{C}'_1 \\ \vdots \\ \tilde{\mathbf{t}}_{\ell^\dagger} \end{pmatrix} \\ = \begin{pmatrix} \mathbf{A}\tilde{\mathbf{z}}^{(0)} + \mathbf{B}\tilde{\mathbf{r}}^{(0)} + \tilde{\mathbf{e}}^{(0)} + \boldsymbol{\eta}^{(0)}_w \\ \mathbf{A}\tilde{\mathbf{z}}^{(1)} + \mathbf{B}\tilde{\mathbf{r}}^{(1)} + \tilde{\mathbf{e}}^{(1)} + \boldsymbol{\eta}^{(1)}_w \\ \vdots \\ \mathbf{A}\tilde{\mathbf{z}}^{(\kappa)} + \mathbf{B}\tilde{\mathbf{r}}^{(\kappa)} + \tilde{\mathbf{e}}^{(\kappa)} + \boldsymbol{\eta}^{(\kappa)}_w \end{pmatrix}\end{aligned} \tag{41}$$

We now perform Vandermonde/adjugate extraction. Let $\mathbf{V}_x \in R_q^{(\kappa+1)\times(\kappa+1)}$ be the Vandermonde matrix on the distinct points $x^{(0)}, \ldots, x^{(\kappa)}$:

$$\mathbf{V}_x := \begin{pmatrix} 1 & x^{(0)} & (x^{(0)})^2 & \cdots & (x^{(0)})^{\kappa} \\ 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{\kappa} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(\kappa)} & (x^{(\kappa)})^2 & \cdots & (x^{(\kappa)})^{\kappa} \end{pmatrix}.$$

Let $(\Gamma_0, \ldots, \Gamma_\kappa)$ be the last row of $\mathrm{adj}(\mathbf{V}_x)$. By $\mathrm{adj}(\mathbf{V}_x)\mathbf{V}_x = \det(\mathbf{V}_x)\mathbf{I}$, these coefficients satisfy

$$\sum_{t=0}^{\kappa}\Gamma_t\,(x^{(t)})^j = 0 \ \text{ for all } j < \kappa, \quad \text{and} \quad \sum_{t=0}^{\kappa}\Gamma_t\,(x^{(t)})^{\kappa} = \det(\mathbf{V}_x).$$

Now multiply (40) by Γ_t and sum over t:

$$\mathbf{A}\hat{\mathbf{z}} + \mathbf{B}\hat{\mathbf{r}} + \hat{\mathbf{e}} + \hat{\boldsymbol{\eta}}_w = \sum_{t=0}^{\kappa}\Gamma_t\sum_{j=0}^{\kappa}(x^{(t)})^j \mathbf{C}'_j = \det(\mathbf{V}_x)\,\tilde{\mathbf{t}}_{\ell^\dagger},$$

where we have defined the extracted responses as

$$\hat{\mathbf{z}} := \sum_{t=0}^{\kappa}\Gamma_t\,\tilde{\mathbf{z}}^{(t)}, \qquad \hat{\mathbf{r}} := \sum_{t=0}^{\kappa}\Gamma_t\,\tilde{\mathbf{r}}^{(t)}, \qquad \hat{\mathbf{e}} := \sum_{t=0}^{\kappa}\Gamma_t\,\tilde{\mathbf{e}}^{(t)},$$

and

$$\hat{\boldsymbol{\eta}}_w := \sum_{t=0}^{\kappa}\Gamma_t\,\boldsymbol{\eta}^{(t)}_w.$$

Thus we have extracted the degree-κ coefficient (up to the relaxation factor $\det(\mathbf{V}_x)$):

$$\mathbf{A}\hat{\mathbf{z}} + \mathbf{B}\hat{\mathbf{r}} + \hat{\mathbf{e}} + \hat{\boldsymbol{\eta}}_w = \det(\mathbf{V}_x)\,\tilde{\mathbf{t}}_{\ell^\dagger}. \tag{42}$$

Extraction on α'_u branch. We can follow the exact same procedure for the α'_u branch to obtain

$$\mathbf{A}\hat{\mathbf{z}}' + \mathbf{B}\hat{\mathbf{r}}' + \hat{\mathbf{e}}' + \hat{\boldsymbol{\eta}}'_w = \det(\mathbf{V}'_x)\,\tilde{\mathbf{t}}_{\ell^\dagger}. \tag{43}$$

Note that because PK^* and u is fixed across both branches α_u and α'_u, we will indeed have the same aggregated key index $\ell^\dagger$ (and hence $\tilde{\mathbf{t}}_{\ell^\dagger}, \tilde{\mathbf{t}}'_{\ell^\dagger}$) extracted across both branches. By the call to $\mathsf{KAgg}(\mathsf{PK}^*)$ in Vf (Fig. 7), we know that

$$\tilde{\mathbf{t}}_{\ell^\dagger} = \sum_{u=0}^{T-1}\alpha^{\dagger}_u \mathbf{t}_{u,\ell^\dagger}, \quad \tilde{\mathbf{t}}'_{\ell^\dagger} = \sum_{u=0}^{T-1}\alpha'^{\dagger}_u \mathbf{t}_{u,\ell^\dagger}.$$

Then, by the formal model, (at least) one of the $\mathbf{t}_{u,\ell^\dagger}$'s must have been generated using OKGen, denote this key as $\mathsf{pk}_{K^\dagger}$. Because otherwise, the adversary would have control of at least T public

keys. Let $K^\dagger \in [Q_k]$ be the index of the OKGen query that returned $\mathsf{pk}_{K^\dagger}$. If $K^\dagger = K$ (which happens with probability $1/Q_k$), we get the following expression for each branch:

$$\mathbf{A}\hat{\mathbf{z}} + \mathbf{B}\hat{\mathbf{r}} + \hat{\mathbf{e}} + \hat{\boldsymbol{\eta}}_w = \det(\mathbf{V}_x)\Big(\alpha_u\, \mathbf{t}^*_{\hat{u}} + \sum_{\mathbf{t}_i \in \mathsf{PK}^*_{\ell^*} \mathbf{t}_i \neq \mathbf{t}^*_{\hat{u}}} \alpha_i\, \mathbf{t}_i\Big), \tag{44}$$

$$\mathbf{A}\hat{\mathbf{z}}' + \mathbf{B}\hat{\mathbf{r}}' + \hat{\mathbf{e}}' + \hat{\boldsymbol{\eta}}'_w = \det(\mathbf{V}'_x)\Big(\alpha'_u\, \mathbf{t}^*_{\hat{u}} + \sum_{\mathbf{t}_i \in \mathsf{PK}^*_{\ell^*} \mathbf{t}_i \neq \mathbf{t}^*_{\hat{u}}} \alpha_i\, \mathbf{t}_i\Big). \tag{45}$$

Subtracting (45) from (44) yields

$$\mathbf{A}(\hat{\mathbf{z}}-\hat{\mathbf{z}}')+\mathbf{B}(\hat{\mathbf{r}}-\hat{\mathbf{r}}')+(\hat{\mathbf{e}}-\hat{\mathbf{e}}')+(\hat{\boldsymbol{\eta}}_w-\hat{\boldsymbol{\eta}}'_w) = \big(\det(\mathbf{V}_x)\alpha_u - \det(\mathbf{V}'_x)\alpha'_u\big)\mathbf{t}^*_{\hat{u}}. \tag{46}$$

Rearrange (46) as

$$\big(\mathbf{t}^*_{\hat{u}} \mid \mathbf{A} \mid \mathbf{B} \mid \mathbf{I} \mid \mathbf{I}\big)\begin{pmatrix} \det(\mathbf{V}_x)\alpha_u - \det(\mathbf{V}'_x)\alpha'_u \\ \hat{\mathbf{z}} - \hat{\mathbf{z}}' \\ \hat{\mathbf{r}} - \hat{\mathbf{r}}' \\ \hat{\mathbf{e}} - \hat{\mathbf{e}}' \\ \hat{\boldsymbol{\eta}}_w - \hat{\boldsymbol{\eta}}'_w \end{pmatrix} = 0 \tag{47}$$

Therefore, $\mathcal{B}$ outputs the non-zero vector

$$\mathbf{v} := \begin{pmatrix} \det(\mathbf{V}_x)\alpha_u - \det(\mathbf{V}'_x)\alpha'_u \\ \hat{\mathbf{z}} - \hat{\mathbf{z}}' \\ \hat{\mathbf{r}} - \hat{\mathbf{r}}' \\ \hat{\mathbf{e}} - \hat{\mathbf{e}}' \\ \hat{\boldsymbol{\eta}}_w - \hat{\boldsymbol{\eta}}'_w \end{pmatrix} \in \mathcal{R}^{1+l+l'+2k}.$$

Non-zeroness holds with overwhelming probability since $\alpha_u \neq \alpha'_u$ and challenges $x^{(0)}, \ldots, x^{(2\kappa)}$ are distinct. We now bound the extracted blocks in coefficientwise ℓ_∞ norm.

Bounding the extracted MSIS vector in ℓ_∞. We bound each block of the right-hand vector of (47) in terms of the verifier bounds $B_{\tilde{z}}, B_{\tilde{r}}, B_{\tilde{e}}$ and the challenge-domain bounds.

Bounds on Vandermonde/adjugate coefficients. Assuming the Fiat-Shamir challenge set $C \subseteq \mathcal{R}$ satisfies $\|x\|_1 \leq w$ for all $x \in C$, we get for the $(\kappa+1)$-forking challenges $x^{(0)}, \ldots, x^{(\kappa)}$, define

$$M_{\det} := \binom{\kappa+1}{2} = \frac{\kappa(\kappa+1)}{2}, \qquad M_\Gamma := \binom{\kappa}{2} = \frac{\kappa(\kappa-1)}{2}.$$

Then, we have $\|x^{(j)} - x^{(i)}\|_1 \leq 2w$ for all $i \neq j$. By the Vandermonde determinant identity, $\det(\mathbf{V}'_x)$, $\det(\mathbf{V}_x) = \prod_{0 \leq i < j \leq \kappa}(x^{(j)} - x^{(i)})$, and by the definition of Γ_t as a product of M_Γ such differences, we obtain

$$\|\det(\mathbf{V}'_x)\|_1, \|\det(\mathbf{V}_x)\|_1 \leq (2w)^{M_{\det}},$$
$$\|\Gamma_t\|_1 \leq (2w)^{M_\Gamma} \quad \text{for every } t \in \{0, \ldots, \kappa\}.$$

For convenience, define the scalars

$$B_{\det} := (2w)^{M_{\det}}, \qquad B_\Gamma := (2w)^{M_\Gamma}.$$

Bounds on $\hat{\mathbf{z}}, \hat{\mathbf{r}}, \hat{\mathbf{e}}$. Each accepting transcript satisfies the verifier norm checks

$$\|\tilde{\mathbf{z}}^{(t)}\| \leq B_{\tilde{z}}, \quad \|\tilde{\mathbf{r}}^{(t)}\| \leq B_{\tilde{r}}, \quad \|\tilde{\mathbf{e}}^{(t)}\| \leq B_{\tilde{e}} \quad \text{for all } t \in \{0, \ldots, \kappa\}.$$

Recall $\hat{\mathbf{z}} = \sum_{t=0}^{\kappa} \Gamma_t \tilde{\mathbf{z}}^{(t)}$ and similarly for $\hat{\mathbf{r}}, \hat{\mathbf{e}}$. Since $\|\mathbf{v}\|_\infty \leq \|\mathbf{v}\|$ for every vector $\mathbf{v}$, we get

$$\begin{aligned}
\|\hat{\mathbf{z}}\|_\infty &\leq \sum_{t=0}^{\kappa} \|\Gamma_t \tilde{\mathbf{z}}^{(t)}\|_\infty \leq \sum_{t=0}^{\kappa} \|\Gamma_t\|_1 \, \|\tilde{\mathbf{z}}^{(t)}\|_\infty \\
&\leq \sum_{t=0}^{\kappa} \|\Gamma_t\|_1 \, \|\tilde{\mathbf{z}}^{(t)}\| \leq (\kappa+1) B_\Gamma B_{\tilde{z}}, \\
\|\hat{\mathbf{r}}\|_\infty &\leq (\kappa+1) B_\Gamma B_{\tilde{r}}, \\
\|\hat{\mathbf{e}}\|_\infty &\leq (\kappa+1) B_\Gamma B_{\tilde{e}}, \\
\|\hat{\boldsymbol{\eta}}_w\|_\infty &\leq \sum_{t=0}^{\kappa} \|\Gamma_t \boldsymbol{\eta}_w^{(t)}\|_\infty \leq \sum_{t=0}^{\kappa} \|\Gamma_t\|_1 \, \|\boldsymbol{\eta}_w^{(t)}\|_\infty \\
&\leq (\kappa+1) B_\Gamma B_{\eta,w}.
\end{aligned}$$

By the same reasoning for the primed quantities,

$$\|\hat{\mathbf{z}}'\|_\infty \leq (\kappa+1) B_\Gamma B_{\tilde{z}}, \quad \|\hat{\mathbf{r}}'\|_\infty \leq (\kappa+1) B_\Gamma B_{\tilde{r}},$$
$$\|\hat{\mathbf{e}}'\|_\infty \leq (\kappa+1) B_\Gamma B_{\tilde{e}}, \quad \|\hat{\boldsymbol{\eta}}'_w\|_\infty \leq (\kappa+1) B_\Gamma B_{\eta,w}.$$

Hence, by the triangle inequality,

$$\|\hat{\mathbf{z}} - \hat{\mathbf{z}}'\|_\infty \leq 2(\kappa+1) B_\Gamma B_{\tilde{z}}, \tag{48}$$
$$\|\hat{\mathbf{r}} - \hat{\mathbf{r}}'\|_\infty \leq 2(\kappa+1) B_\Gamma B_{\tilde{r}}, \tag{49}$$
$$\|\hat{\mathbf{e}} - \hat{\mathbf{e}}'\|_\infty \leq 2(\kappa+1) B_\Gamma B_{\tilde{e}}, \tag{50}$$
$$\|\hat{\boldsymbol{\eta}}_w - \hat{\boldsymbol{\eta}}'_w\|_\infty \leq 2(\kappa+1) B_\Gamma B_{\eta,w}. \tag{51}$$

Bound on $\det(\mathbf{V}_x)\alpha_u - \det(\mathbf{V}'_x)\alpha'_u$. Since $\alpha_u, \alpha'_u \in C$, we have $\|\alpha_u\|_\infty, \|\alpha'_u\|_\infty \leq 1$. Hence, by the coefficientwise bound $\|ab\|_\infty \leq \|a\|_1 \|b\|_\infty$ and the triangle inequality,

$$\begin{aligned}
&\|\det(\mathbf{V}_x)\alpha_u - \det(\mathbf{V}'_x)\alpha'_u\|_\infty \\
&\leq \|\det(\mathbf{V}_x)\alpha_u\|_\infty + \|\det(\mathbf{V}'_x)\alpha'_u\|_\infty \\
&\leq \|\det(\mathbf{V}_x)\|_1 \|\alpha_u\|_\infty + \|\det(\mathbf{V}'_x)\|_1 \|\alpha'_u\|_\infty \\
&\leq 2B_{\det}.
\end{aligned} \tag{52}$$

MSIS^∞ *bound.* Define

$$B_{\max} := \max\{B_\alpha, B_{\hat{z}}, B_{\hat{r}}, B_{\hat{e}}, B_{\hat{\eta}}\}.$$

Then the extracted non-zero vector $\mathbf{v} \in \mathcal{R}^{1+l+l'+2k}$ satisfies $\|\mathbf{v}\|_\infty \leq B_{\max}$. Thus, $\mathcal{B}$ successfully obtains a solution to

$$\mathsf{MSIS}^\infty_{q,k,1+l+l'+2k,B_{\max}}$$

with non-negligible probability.

D.6 Proof of Theorem 5.7

Proof. We proceed via a sequence of hybrid games. Fix an adversary $\mathcal{A}$ and any valid anonymity challenge $(\mu^*, U_0, U_1, (R^*, \Psi^*))$ produced by $\mathcal{A}$ in the experiment, where U_0 and U_1 are two candidate signer sets of size T in R^* containing honest signers. Let S_{U_0} and S_{U_1} denote the corresponding secret-key sets.

Game G_0. This is exactly $\mathrm{Exp}^{\text{anon-0}}_{\mathsf{LoTRS},\mathcal{A}}$. The challenger answers all signing oracle queries honestly, and the challenge signature

$$\sigma^* := \big(\pi := (B^{(1)}_{\text{bin}}, (\tilde{\mathbf{w}}^{(1)}_1, \ldots, \tilde{\mathbf{w}}^{(1)}_{\kappa-1}), x, \mathbf{f}_1, \mathbf{z}_b), \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}\big)$$

is also produced honestly using S_{U_0}.

$\mathsf{OSign}_{\mathsf{bin}}(\ell, \mu, (\tilde{\mathbf{w}}^{(1)})_{j\in[1,\kappa-1]}, \mathsf{PK}, \tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}; \rho)$

$\mathbf{A} \coloneqq \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{A}; \mathcal{R}_q^{k\times l})$

$\mathbf{G} \coloneqq \mathsf{Expand}(\rho_{\mathsf{pp}}, \mathsf{G}; \mathcal{R}_{\hat{q}}^{\hat{n}\times(\hat{n}+\hat{k}+2\kappa\beta)})$

$\mathbf{z}_b \xleftarrow{\$} D_{\phi_b \mathcal{B}_b}^{\hat{n}+\hat{k}}$ / G_1–G_5

$(f_{j,i}) \leftarrow D_{\phi_a B_a}$ / G_1–G_5

for $j = 0, \ldots, \kappa - 1$ **do**

$\quad f_{j,0} \coloneqq x - \sum_{i=1}^{\beta-1} f_{j,i}$

endfor

$\mathbf{f} \coloneqq (f_{j,0}, \ldots, f_{j,\beta-1})_{j\in[0,\kappa-1]}$

Define $g_{j,i} \coloneqq f_{j,i}(x - f_{j,i})$

$\mathbf{g}_0 \coloneqq (g_{0,0}, \ldots, g_{\kappa-1,0})$

$\mathbf{g}_1 \coloneqq (g_{j,i})_{j\in[0,\kappa-1],\, i\in[1,\beta-1]}$

Decompose ℓ in base β as $\ell = (\ell_j)$ with $\ell_j \in [0, \beta-1]$ / G_1–G_3

$\mathbf{b} \coloneqq (\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})$ / G_1–G_3

$(a_{j,i})_{j,\, i\in[1,\beta-1]} \coloneqq \mathsf{Expand}(\rho, \mathsf{a}; D_{\phi_a B_a}^{\kappa(\beta-1)})$ / G_1–G_3

for $j = 0, \ldots, \kappa - 1$ **do** / G_1–G_3

$\quad a_{j,0} \coloneqq -\sum_{i=1}^{\beta-1} a_{j,i}$

endfor

$\mathbf{a} \coloneqq (a_{j,0}, \ldots, a_{j,\beta-1})_{j\in[0,\kappa-1]}$ / G_1–G_3

$\mathbf{c} \coloneqq \mathbf{a} \circ (\mathbf{1} - 2\mathbf{b}),\ \mathbf{d} \coloneqq (-(a_{j,i})^2)_{j,i}$ / G_1–G_3

$\mathbf{r}_b \coloneqq \mathsf{Expand}(\rho, \mathsf{rb}; S_1^{\hat{n}+\hat{k}})$ / G_1

$B_{\mathsf{bin}} \coloneqq \mathbf{G}(\mathbf{r}_b, \mathbf{b}, \mathbf{c})^\top$ / G_1

$B_{\mathsf{bin}} \xleftarrow{\$} \mathcal{R}_{\hat{q}}^{\hat{n}}$ / G_2

Decompose $B_{\mathsf{bin}} \coloneqq 2^{K_B} B_{\mathsf{bin}}^{(1)} + B_{\mathsf{bin}}^{(0)}$

$A_{\mathsf{bin}} \coloneqq \mathbf{G}(\mathbf{z}_b, \mathbf{f}, \mathbf{g})^\top - x2^{K_B} B_{\mathsf{bin}}^{(1)} \in \mathcal{R}_{\hat{q}}^{\hat{n}}$ / G_1–G_5

Decompose $A_{\mathsf{bin}} \coloneqq 2^{K_A} A_{\mathsf{bin}}^{(1)} + A_{\mathsf{bin}}^{(0)}$

$H(\mu, A_{\mathsf{bin}}^{(1)}, B_{\mathsf{bin}}^{(1)}, \tilde{\mathbf{w}}_0^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}, \mathsf{PK}) \coloneqq x$ / program H with x, G_1–G_5

return $\pi \coloneqq (B_{\mathsf{bin}}^{(1)},\ (\tilde{\mathbf{w}}_1^{(1)}, \ldots, \tilde{\mathbf{w}}_{\kappa-1}^{(1)}),\ x,\ \mathbf{f}_1,\ \mathbf{z}_b)$

Figure 14: Partial hybrids for anonymity game.

Simulation. In the following games, we will simulate σ^*. Note that simulation of $\tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}$ is easier to picture compared to that of π. Therefore we only show how the $\mathsf{Sign}_{\mathsf{bin}}$ procedure changes via $\mathsf{OSign}_{\mathsf{bin}}$ in Fig. 14.

Game G_1. Now, we start simulating responses for some parts of σ^*. First, replace $\mathbf{z}_b$ by random samples from $D_{\phi_b B_b}$ and $\mathbf{f}_1$ by random samples in $D_{\phi_a B_a}$. Since $\mathcal{A}$ launches Q_s signing sessions, by rejection sampling Lemma A.4, G_1 is indistinguishable from G_0, so

$$\left|\Pr[G_1(\mathcal{A}) = 1] - \Pr[G_0(\mathcal{A}) = 1]\right| \le Q_s\left(\frac{2^{-100}}{\mu(\phi_b)} + \frac{2^{-100}}{\mu(\phi_a)}\right). \quad (53)$$

Game G_2. In G_2, replace B_{bin} by a uniformly random element in $\mathcal{R}_{\hat{q}}^{\hat{n}}$. Then, for uniformly generated $\mathbf{G}$, the pair $(\mathbf{G},\ \mathbf{G}\begin{pmatrix}\mathbf{r}_b\\ \mathbf{b}\\ \mathbf{c}\end{pmatrix})$ is computationally indistinguishable from $(\mathbf{G},\ B_{\mathsf{bin}})$, under the $\mathsf{MLWE}_{\hat{q},\hat{n},\hat{k},1}$ assumption.

$$\left|\Pr[G_2(\mathcal{A}) = 1] - \Pr[G_1(\mathcal{A}) = 1]\right| \le Q_s \cdot \mathsf{Adv}^{\mathsf{MLWE}}_{\hat{q},\hat{n},\hat{k},1}(\mathcal{D}'). \quad (54)$$

Game G_3. In this hybrid, for each signer $u \in U_0$, sample $\mathbf{z}_u \xleftarrow{\$} D_{\phi_0 B_0}$. By rejection sampling, the statistical distance between the randomly sampled $\mathbf{z}_u$ and a real one is at most 2^{-100}. Given that

there are T signers, in at most Q_s sessions, we get

$$\left|\Pr[\mathrm{G}_3(\mathcal{A})=1]-\Pr[\mathrm{G}_2(\mathcal{A})=1]\right| \le Q_s \cdot T\frac{2^{-100}}{\mu(\phi_0)}. \tag{55}$$

Game G_4. Here we replace $\tilde{\mathbf{w}}_1,\dots,\tilde{\mathbf{w}}_{\kappa-1}$ with simulated values: for each signer $u \in U_0$, by Lemma A.8, each $\mathbf{w}_{u,j}$ is within a negligible statistical distance ϵ from uniform (see Claim (1), Lemma 5.3). Therefore, since we have $\kappa-1$ randomly chosen samples for each signer and T signers

$$\left|\Pr[\mathrm{G}_4(\mathcal{A})=1]-\Pr[\mathrm{G}_3(\mathcal{A})=1]\right| \le Q_s \cdot T \cdot (\kappa-1)\epsilon. \tag{56}$$

At this point, the entire σ^* is simulated and is independent of any secret keys and the signer set U_0.

Game G_5. In this game, the challenger now switches to the signer set U_1. Notice that since σ^* is simulated, the view of $\mathcal{A}$ is still the same as the previous game.

$$\left|\Pr[\mathrm{G}_5(\mathcal{A})=1]-\Pr[\mathrm{G}_4(\mathcal{A})=1]\right| = 0. \tag{57}$$

Game G_6. Now we perform in reverse the changes made from G_4 to G_0 to generate σ^* honestly using the signing set S_{U_1} to obtain the result. □

E More on Parameter Setting

Parameter selection for bit dropping. We choose the bit-dropping parameters $(K_A, K_B, K_{w,0})$ using the same Bai–Galbraith-style heuristic that the extra compression check should contribute only a small slowdown in the signing procedure. In particular, we fix the target restart multiplier of the compressed binary first message to be a near-unit constant

$$\mu^\star_{\mathrm{BG}} := 1+\varepsilon_{\mathrm{BG}},$$

for a small constant $\varepsilon_{\mathrm{BG}} > 0$. In the concrete estimates, $\mu^\star_{\mathrm{BG}}$ is chosen together with the other rejection-sampling parameters so that the overall expected number of full signing attempts is approximately 3.

Choosing K_A from K_B. Recall that

$$\hat{A}_{\mathsf{bin}} = A_{\mathsf{bin}} + xB^{(0)}_{\mathsf{bin}}, \qquad \|xB^{(0)}_{\mathsf{bin}}\|_\infty \le w\,2^{K_B-1},$$

and that the Bai–Galbraith stability check requires

$$\|\hat{A}^{(0)}_{\mathsf{bin}}\|_\infty \le 2^{K_A-1} - w\,2^{K_B-1}.$$

Writing

$$E_A := w\,2^{K_B-1},$$

and using the same coefficient-wise heuristic as in the Dilithium-style acceptance analysis, a single coefficient passes with probability

$$p_{\mathrm{BG},1} = 1-\frac{w2^{K_B}-1}{2^{K_A}}.$$

Therefore, over all $\hat{n}d$ coefficients,

$$p_{\mathrm{BG}} \approx \left(1-\frac{w2^{K_B}-1}{2^{K_A}}\right)^{\hat{n}d} \approx \exp\left(-\hat{n}d\,\frac{w2^{K_B}-1}{2^{K_A}}\right).$$

Hence

$$\mu_{\mathrm{BG}} := p_{\mathrm{BG}}^{-1} \approx \exp\left(\hat{n}d\,\frac{w2^{K_B}-1}{2^{K_A}}\right).$$

To ensure that the compressed binary first message contributes at most the target restart multiplier $\mu^\star_{\mathrm{BG}}$, we require

$$\mu_{\mathrm{BG}} \le \mu^\star_{\mathrm{BG}}.$$

Equivalently, it suffices to choose K_A such that

$$\hat{n}d\,\frac{w2^{K_B}-1}{2^{K_A}} \le \ln \mu^\star_{\mathrm{BG}}.$$

Thus we choose

$$2^{K_A} \ge \frac{\hat{n}d\,(w2^{K_B}-1)}{\ln \mu^\star_{\mathrm{BG}}}.$$

Accordingly, we set

$$K_A := \left\lceil \log_2\left(\frac{\hat{n}d\,(w2^{K_B}-1)}{\ln \mu^\star_{\mathrm{BG}}}\right)\right\rceil.$$

Equivalently, up to lower-order constants,

$$K_A \approx K_B + \log_2(\hat{n}d\,w) - \log_2(\ln \mu^\star_{\mathrm{BG}}).$$

Since $A^{(1)}_{\mathsf{bin}}$ is only hashed and not transmitted, this choice of K_A does not directly increase the signature size. It must, however, be accounted for in the binding/AMSIS parameter estimates.

Role of K_B. The parameter K_B is the primary compression knob for the transmitted binary commitment $B^{(1)}_{\mathsf{bin}}$: increasing K_B saves $\hat{n}d$ bits per extra dropped bit, but it also increases the hidden perturbation $E_A = w2^{K_B-1}$, and therefore forces a larger K_A. Since $A^{(1)}_{\mathsf{bin}}$ is only hashed and not transmitted, K_A is not chosen for size reduction, but solely to absorb the perturbation coming from the dropped bits of B_{bin}.

Choosing $K_{w,0}$ in the case $\kappa = 1$ with compression only on $\tilde{\mathbf{w}}_0$. In this simplified setting, there are no higher-index aggregated commitments $\tilde{\mathbf{w}}_j$ with $j \ge 1$. Hence the hidden perturbation term vanishes:

$$\Delta_w := \sum_{j=1}^{\kappa-1} x^j \tilde{\mathbf{w}}^{(0)}_j = 0, \qquad M_w := \sum_{j=1}^{\kappa-1} w^j\,2^{K_{w,j}-1} = 0.$$

Therefore the verifier-side reconstruction is exact:

$$\widehat{\tilde{\mathbf{w}}}_0 = \sum_{i=0}^{N-1} f_{0,i}\tilde{\mathbf{t}}_i - (\bar{\mathbf{A}}\tilde{\mathbf{z}} + \bar{\mathbf{B}}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}) = \tilde{\mathbf{w}}_0.$$

Now write the coefficient-wise centered decomposition of $\tilde{\mathbf{w}}_0$ modulo $2^{K_{w,0}}$:

$$\tilde{\mathbf{w}}_0 = 2^{K_{w,0}}\tilde{\mathbf{w}}^{(1)}_0 + \tilde{\mathbf{w}}^{(0)}_0, \qquad \|\tilde{\mathbf{w}}^{(0)}_0\|_\infty \le 2^{K_{w,0}-1}.$$

Since $\widehat{\tilde{\mathbf{w}}}_0 = \tilde{\mathbf{w}}_0$, there is no additional stability condition to enforce and no extra restart event associated with $\tilde{\mathbf{w}}$-compression. Accordingly, in the proof one simply defines the proof-local remainder $\boldsymbol{\eta}_w := \tilde{\mathbf{w}}^{(0)}_0$, so that $\widehat{\tilde{\mathbf{w}}}_0 = 2^{K_{w,0}}\tilde{\mathbf{w}}^{(1)}_0 + \boldsymbol{\eta}_w$, $\|\boldsymbol{\eta}_w\|_\infty \le 2^{K_{w,0}-1}$. Hence, in this case, $B_{\eta,w} := 2^{K_{w,0}-1}$.

Therefore $K_{w,0}$ is not chosen from a hidden-perturbation budget, but purely as a size-versus-security tradeoff. Increasing $K_{w,0}$ reduces the number of bits needed to transmit $\tilde{\mathbf{w}}^{(1)}_0$, but increases the proof-side residual bound $B_{\eta,w}$, and hence increases the contribution $kB^2_{\eta,w}$ inside the final β_{SIS} bound. Thus $K_{w,0}$ should be taken as large as possible subject to the condition that the resulting β_{SIS} still yields the desired security level.

Concrete repetition estimate. We provide an example for our parameter setting. In general, the total expected number of full signing attempts is heuristically

$$\mu_{\mathrm{tot}}(T) \approx \mu(\phi)^T\, \mu(\phi_a)\, \mu(\phi_b)\, \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}}\, \mu_w.$$

Here the factor $\mu(\phi)^T$ comes from the T per-signer rejection events in Sign_2, whereas $\mu(\phi_a)$, $\mu(\phi_b)$, and μ_{BG} come from the single common binary subproof and therefore appear only once. The factor μ_{fg} accounts for the additional restart probability caused by the tightened signer-side checks on $\mathbf{f}_1, \mathbf{f}_0, \mathbf{g}_1, \mathbf{g}_0$, and the factor μ_w accounts for any extra restart cost due to compressed aggregated commitments.

In the present simplified setting, we restrict to $\kappa = 1$ and compress only $\tilde{\mathbf{w}}_0$. Hence there are no higher-index compressed terms $\tilde{\mathbf{w}}_j$ with $j \geq 1$, so the hidden perturbation term vanishes:

$$M_w = \sum_{j=1}^{\kappa-1} w^j\, 2^{K_{w,j}-1} = 0.$$

Therefore there is no additional carry-stability restart condition, and thus

$$\mu_w = 1.$$

Moreover, since the tightened bounds on $\mathbf{f}_1, \mathbf{f}_0, \mathbf{g}_1, \mathbf{g}_0$ are chosen so that their combined signer-side failure probability is at most $\varepsilon_{\mathrm{tot}}$, their contribution to the repetition count is heuristically

$$\mu_{\mathrm{fg}} \approx \frac{1}{1-\varepsilon_{\mathrm{tot}}}.$$

Accordingly, in this case the total repetition estimate reduces to

$$\mu_{\mathrm{tot}}(T) \approx \mu(\phi)^T\, \mu(\phi_a)\, \mu(\phi_b)\, \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}}.$$

We now keep ϕ_a fixed, and scale only ϕ linearly with T, say

$$\phi = cT,$$

for a constant $c > 0$, while also keeping ϕ_b fixed. Then

$$\mu(\phi)^T = \exp\left(T\left(\frac{12}{\phi} + \frac{1}{2\phi^2}\right)\right) = \exp\left(\frac{12}{c} + \frac{1}{2c^2T}\right),$$

whereas

$$\mu(\phi_a) = \exp\left(\frac{12}{\phi_a} + \frac{1}{2\phi_a^2}\right)$$

is now independent of T. Hence

$$\mu_{\mathrm{tot}}(T) \approx \exp\left(\frac{12}{c} + \frac{1}{2c^2T} + \frac{12}{\phi_a} + \frac{1}{2\phi_a^2} + \frac{1}{2\phi_b^2}\right) \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}}\, \mu_w.$$

In particular, for moderate or large T, this simplifies to the first-order approximation

$$\mu_{\mathrm{tot}}(T) \approx \exp\left(\frac{12}{c} + \frac{12}{\phi_a} + \frac{1}{2\phi_a^2} + \frac{1}{2\phi_b^2}\right) \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}}\, \mu_w,$$

up to $1 + O(1/T)$ factors. Thus, even with ϕ_a fixed, choosing $\phi \propto T$ keeps the dominant signer-side repetition factor roughly constant as T grows.

More generally, if we target an expected number $\mu_\star$ of sequential signing attempts, then a convenient first-order design rule is to choose

$$\phi = cT,$$

with

$$c \approx \frac{12}{\ln \mu_\star - \ln \mu_{\mathrm{BG}} - \ln \mu_{\mathrm{fg}} - \ln \mu_w - \frac{12}{\phi_a} - \frac{1}{2\phi_a^2} - \frac{1}{2\phi_b^2}},$$

where the lower-order correction $\frac{1}{2c^2T}$ may be added back if one wants a more accurate estimate.

For the concrete setting $N = 100$ and $T = 50$, let us take

$$\phi_a = 24, \qquad \phi_b = 4, \qquad \varepsilon_{\mathrm{tot}} = 0.01, \qquad \mu_{\mathrm{BG}} = 1.01,$$

and target

$$\mu_\star \approx 3$$

expected sequential signing attempts. Then

$$\mu_{\mathrm{fg}} \approx \frac{1}{1-\varepsilon_{\mathrm{tot}}} = \frac{1}{0.99} \approx 1.0101, \qquad \mu_w = 1.$$

The above heuristic therefore gives

$$c \approx \frac{12}{\ln 3 - \ln(1.01) - \ln(1/0.99) - \frac{12}{24} - \frac{1}{2\cdot 24^2} - \frac{1}{2\cdot 4^2}} \approx 21.96,$$

so that we may choose

$$\phi_a = 24, \qquad \phi_b = 4, \qquad \phi \approx 22\,T \approx 1100.$$

With these values,

$$\mu(\phi_a) = \exp\left(\frac{12}{24} + \frac{1}{2\cdot 24^2}\right) \approx 1.6502,$$
$$\mu(\phi_b) = \exp\left(\frac{1}{2\cdot 4^2}\right) \approx 1.0317,$$
$$\mu(\phi) = \exp\left(\frac{12}{1100} + \frac{1}{2\cdot 1100^2}\right) \approx 1.0110.$$

Hence

$$\begin{aligned}\mu_{\mathrm{tot}}(50) &\approx \mu(\phi)^{50}\, \mu(\phi_a)\, \mu(\phi_b)\, \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}}\, \mu_w \\ &= \mu(\phi)^{50}\, \mu(\phi_a)\, \mu(\phi_b)\, \mu_{\mathrm{BG}}\, \mu_{\mathrm{fg}} \\ &\approx 3.\end{aligned}$$

Thus the expected number of full signing attempts is approximately 3, giving an expected round number of about $3 \times 2 = 6$.

□

F Implementation and Performance

This appendix supplements §6.1 with the full benchmark grid, a sign-time decomposition of the two protocol layers, and the two natural standalone reductions of LoTRS to a plain ring signature (T=1) and a plain multi-signature (N=1). All timings are from a single back-to-back run on a single machine, so they are directly comparable. The reference implementation uses `rayon` for the per-signer round-1 / round-2 loops, key aggregation, verifier public-key decoding, and the verifier's ring-keyed selector sum; the CRT-NTT transforms of the public matrices A, G, B are amortized within each signing call. Sizes refer to the canonical Golomb–Rice / fixed-width encoding actually emitted by the reference codec. The performance measurements were obtained with the implementation available from: https://github.com/lotrs-sig/lotrs

Hardware and methodology. For all of our benchmarking we used a laptop with 32GB RAM; AMD Ryzen AI 9 HX 370 (Zen 5, 12 cores / 24 threads, 5.1 GHz max boost); Debian Forky with `Linux 7.0.4 amd64`; `rustc 1.95.0`, release build, `rayon` multi-threaded across all 24 hardware threads. Sign and Verify are arithmetic means over $n_{\text{samp}} = 100$ signatures per cell, uniform across the grid; KeyGen and KAgg are setup measurements, while the pk / $\tilde{\sigma}$ sizes are deterministic given the seeds. The hash of the public-key table is computed once per call and reused inside H_{agg}, H_{com}, and the Fiat–Shamir hash, so the per-call hashing cost scales with |PK| rather than $T \cdot |\text{PK}|$ (cf. Sec. F.2, where this design choice is what keeps Verify / KAgg growing only linearly in T).

F.1 Combined protocol (LoTRS, threshold ring signature)

Table 6 reports the full LoTRS pipeline at $N \in \{32, 100\}$ and a sweep of thresholds T. The rightmost column gives the number of rejection-sampling attempts per accepted signature; the empirical mean is in the 5–7 range across all cells, slightly above the $\mu_{\text{tot}} \approx 3$ multiplicative prediction of Table 3 because the implementation also restarts on the additional $\tilde{\mathbf{w}}_0$-stability check used by the $\kappa = 1$ commitment-compression path. The per-attempt sign cost is the more parameter-stable indicator.

Table 6: LoTRS combined-protocol performance. Times are means over 100 signing runs per cell; sizes are exact bytes emitted by the reference codec.

N	T	Sign (ms)	Verify (ms)	KAgg (ms)	$\tilde{\sigma}$ (KiB)	ring PK (MiB)	att. (num)
32	4	119.6	25.5	9.8	23.89	0.89	5.41
32	8	142.7	31.0	16.2	24.34	1.78	5.15
32	16	149.1	43.2	31.6	24.98	3.56	4.32
32	32	282.2	66.0	51.6	25.42	7.12	6.03
100	5	417.2	60.1	22.9	34.01	3.48	6.84
100	10	441.8	81.4	43.0	34.64	6.96	6.06
100	25	567.4	145.2	103.2	35.36	17.40	6.23
100	50	788.5	250.0	198.3	35.79	34.79	5.51

F.2 Building-block decomposition

LoTRS composes a DualMS-style multi-signature with a one-out-of-many selection proof (Sign_{bin}). Both layers contribute independently to the total cost, and we instrument the implementation to attribute wall-clock time to each: $\text{Sign}_{\text{DualMS}}$ covers Sign_1, Sign_2 minus the Sign_{bin} call, and SAgg; $\text{Sign}_{\text{Sign}_{\text{bin}}}$ is the binary selection proof. Verify is split symmetrically: $\text{Verify}_{\text{DualMS}}$ covers the $\tilde{\mathbf{z}}, \tilde{\mathbf{r}}, \tilde{\mathbf{e}}$ norm bounds, the $\mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\tilde{\mathbf{r}} + \tilde{\mathbf{e}}$ reconstruction, key aggregation, the ring-keyed selector loop, and the closing Fiat–Shamir hash; $\text{Verify}_{\text{Sign}_{\text{bin}}}$ covers the f_1, f_0, g_0, g_1 bounds and the $\hat{A}_{\text{bin}}$ reconstruction.

Figure 15 plots the decomposition. The DualMS layer dominates for $T \geq 4$, scaling roughly linearly with T; the Sign_{bin} proof is computed once per rejection-sampling attempt and broadcast to all T signers, with SAgg asserting that the T transcripts agree, so its per-attempt cost is independent of T and depends only on N (the per-signature cost also reflects the empirical attempt count). The *Context / setup* band on top of the bar attributes the per-call one-shot work — expanding the public matrices A, G, B, hashing the public-key table, and pre-computing the α_u challenges — none of which scale with the empirical attempt count. $\text{Verify}_{\text{Sign}_{\text{bin}}}$ stays at ~ 16 ms for $N = 32$ and ~ 40 ms for $N = 100$ across all T, since it does not interact with the multi-signer aggregation; $\text{Verify}_{\text{DualMS}}$ scales with T and dominates verification at the headline parameters.

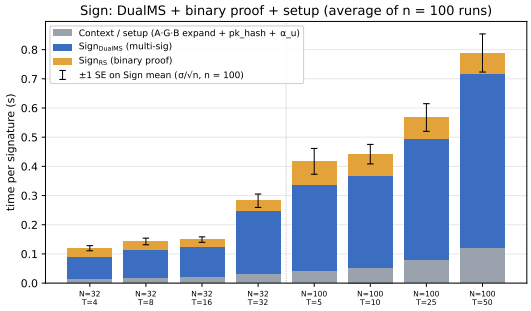

Figure 15: Sign-time decomposition for the combined protocol, stacked bars per (N, T) cell, bottom to top: *Context / setup* (matrix expansion, PK hash, α_u precompute, paid once per sign call), $\text{Sign}_{\text{DualMS}}$ (the per-signer multi-signature work, repeated per rejection-sampling attempt), and $\text{Sign}_{\text{Sign}_{\text{bin}}}$ (the binary selection proof, run once per attempt). Black caps indicate ±1 standard error on the mean over 100 signatures.

F.3 Standalone building blocks

To facilitate comparison with future lattice multi-signature and ring signature constructions, we report the two building blocks in isolation, by running the LoTRS implementation at the corresponding edge of the parameter space. These numbers are from the same single-machine run as the combined-protocol table above.

Plain ring signature ($T = 1$). With a single signer, the DualMS layer collapses to a Schnorr-style proof of knowledge while Sign_{bin} continues to hide the chosen column among N. The resulting object is a plain 1-of-N ring signature on the same d=128 lattice as the combined construction.

Table 7: RS-alone (LoTRS at $T = 1$). Numbers come from the same implementation as Table 6, with 100 signatures per cell.

N	Sign (ms)	Verify (ms)	σ (KiB)	att.
32	108.0	19.4	22.82	6.49
100	245.3	43.3	32.84	5.82

Plain multi-signature ($N = 1$). With a singleton ring, there is no signer-column to hide; $\mathsf{Sign}_{\mathsf{bin}}$ becomes vacuous, and the protocol reduces to a T-party DualMS-style multi-signature on a single key column. We report the $\mathsf{Sign}_{\mathsf{DualMS}}$ and $\mathsf{Verify}_{\mathsf{DualMS}}$ sub-times, which exclude the residual structural cost of the degenerate $\beta = 1$ binary proof that still runs in this implementation.

Table 8: DualMS-alone (LoTRS at $N = 1$, reading the DualMS sub-times to exclude the residual $\beta = 1$ $\mathsf{Sign}_{\mathsf{bin}}$ overhead). Each cell uses 100 signatures.

T	$\mathsf{Sign}_{\mathsf{DualMS}}$ (ms)	$\mathsf{Verify}_{\mathsf{DualMS}}$ (ms)	σ (KiB)	att.
2	27.3	2.6	18.70	5.76
4	35.6	3.0	19.34	6.98
8	35.2	4.0	19.79	5.63
16	45.0	5.9	20.42	5.71

F.4 Caveats on the standalone tables

The numbers in Tables 7 and 8 are obtained by running LoTRS at the corresponding edge of the parameter space, not by running a separately tuned protocol. Two points follow.

(1) **RS-alone parameters are not re-tuned for $T = 1$.** The rejection-sampling slack $\varphi = 22 \cdot T$ is set for the threshold regime and drops to 22 at $T = 1$. Empirical attempt counts at $T = 1$ stay in the same ~ 6 range as the threshold cells, so the signature size is unaffected; the time should be read as a loose upper bound on what a tuned standalone RS would achieve.

(2) **DualMS-alone retains residual $\mathsf{Sign}_{\mathsf{bin}}$ scaffolding.** At $N = 1$ the binary proof is degenerate (f_1 empty, one-hot of length one), but the implementation still expands $\hat{A}_{\mathsf{bin}}/\hat{B}_{\mathsf{bin}}$ and runs RejOp on $\mathbf{z}_b$ (~ 19 ms of sign time at $T = 8$). The $\mathsf{Sign}_{\mathsf{DualMS}}$ / $\mathsf{Verify}_{\mathsf{DualMS}}$ columns exclude this residual cost, so they reflect the genuine multi-signature work.