Pebble: Normal Threshold ML-DSA

Yashvanth Kondi, Oleksandra Lapiha, Iraklis Leontiadis, and Tanushri Sen

Silence Laboratories

July 28, 2026

Abstract

Threshold signatures underpin digital-asset custody and other deployments in which a small online quorum co-signs transactions. As post-quantum migration brings standardized lattice signatures such as ML-DSA into this setting, a central difficulty emerges: unlike the elliptic curve setting, ML-DSA responses are short *integer* vectors produced by rejection sampling, so a threshold signing protocol must specify target ideal distributions for partial signatures that pass standard ML-DSA verification when aggregated. Previous work by Celi et al. (USENIX '26) has shown that partial signatures sampled from hyperballs meet this requirement.

We introduce Pebble, a threshold ML-DSA protocol built on discrete-Gaussian partial nonces. Because Discrete Gaussians enable fine-grained entropy analysis over rings and behave well under linear transformations, we obtain a cleaner and more conservative security analysis than the hyperball approach.

We further present Pebble-ADA, an extension of the above scheme that instantiates *associated-data authentication* (ADA). This enables applications such as proof of quorum identity in the lattice setting.

Contents

1 Introduction

In recent years, we have seen wide deployment of threshold signatures, particularly in the domain of digital asset security. Threshold signatures find use in two- and multi-factor authentication for wallets, as a building block in consensus protocols, and in cross-chain bridges to name a few applications. While the broad principle is the same, each application comes with its own set of requirements, leading to different priorities when choosing a scheme. For instance, consensus-focused applications prioritize non-interactive aggregation as provided natively by BLS. Transactions however favour more traditional schemes such as ECDSA and EdDSA/Schnorr, which enjoy wide support across a range of hardware, as well as fast signing and verification. In this paper, we focus on threshold signing for use in custody applications, which requires threshold signing transactions, typically amongst a small quorum of parties—often just two signing nodes.

Given the post-quantum migration mandates from governments around the world as well as sections of the industry, post-quantum variants of the above signature schemes are under consideration for deployment. As of writing, there are two classes of cryptographic schemes believed to be quantum resilient that have been broadly standardized: those based on hashing, and those on lattices. While hash based schemes generally do not have much structure to leverage for efficient threshold signing protocols, lattice based schemes seem more promising, with several recent works on threshold lattice signatures [GKS24, DKM+24, EKT24, BKL+25, BdCE+25, CdPE+26]. However, threshold signing in the lattice setting introduces fresh challenges relative to traditional elliptic curve based schemes.

Distributed Nonce Sampling in the Lattice Setting

Threshold signing protocols for elliptic curve schemes benefit greatly from the ease of masking secrets with uniform values. A Schnorr signature for instance consists of a random curve point, i.e. the "nonce" $R = r \cdot G$ along with scalar $z = s \cdot c + r$, where $s \in \mathbb{Z}_q$ is the secret key, and $c = \mathsf{Hash}(\mathsf{m}, R)$ is a public Fiat-Shamir challenge. The nonce scalar $r \in \mathbb{Z}_q$ is uniformly random in $\mathbb{Z}_q$, which yields a simple argument that z is distributed uniformly in $\mathbb{Z}_q$ as well, i.e. it can be sampled independently of s in the security proof. In the threshold setting, it is trivial to sample r in a secret shared fashion—each party P_i can locally sample its share r_i uniform in $\mathbb{Z}_q$, so that their sum $r = \sum_i r_i$ is distributed uniformly. Signing is then only a matter of linearly reconstituting $z = \sum_i z_i$ from individual partial signatures $z_i = c \cdot s_i + r_i$.

In the lattice setting, distributed sampling of nonces is not as straightforward. The simplest implementation of Schnorr's template with lattices is arguably Raccoon [dPKPR24], wherein the signature consists of a nonce $\mathbf{w} = \mathbf{A}' \cdot \mathbf{r}$ along with $\mathbf{z} = c \cdot \mathbf{s} + \mathbf{r}$, where $\mathbf{s} \in \mathcal{R}^{k+\ell}$ is the secret vector over a polynomial ring, and c is the Fiat-Shamir challenge once more. However in this case $\mathbf{z}$ is an integer vector (in the embedding), and it is imperative that $\mathbf{z}$ be verified to be "short" per a norm bound. In other words, $\mathbf{s}$ is masked over the integers rather than uniformly modulo a fixed value. One simple method of masking over the integers is to use an $\mathbf{r}$ with larger norm so that the shift by $\mathbf{s}$ dependent value is statistically unnoticeable—also known as noise flooding. After technical refinements, this method is conducive to efficient threshold signing as well; threshold Raccoon [DKM+24] has each party P_i locally sample its share $\mathbf{r}_i$ to be large enough to mask $c \cdot \mathbf{s}$ in isolation.

Rejection Sampling

While noise flooding yields simple constructions, in practice it results in large signatures due to the size of the nonce required for masking. For example threshold Raccoon obtains signatures of size 12.7 KB and modulus of size $q = 2^{49}$ compared to the single signer Dilithium with signature of size < 2.5 KB and modulus $q < 2^{23}$. Lyubashevsky's "Fiat-Shamir with Aborts" technique [Lyu09]—the basis for Dilithium—avoids this overhead by setting an ideal target distribution first, and reject sampling into it when signing. In particular, the Dilithium signature specifies that $\mathbf{z}$ be distributed uniformly in a hypercube of a fixed size, and instructs the signing procedure to simply try again in the event that it produces a putative $\mathbf{z}$ that lies outside of the hypercube. The possibility of restart further reduces the distance between the ideal and the real distribution shifted by a secret value. Shrinking $\mathbf{r}$ still lowers the probability that a putative $\mathbf{z}$ lies within the target hypercube, but $\mathbf{r}$ (and correspondingly the signature) can still be substantially smaller than with statistical masking.

Threshold Signing in the Fiat-Shamir with Aborts paradigm is unfortunately not as straightforward as with noise flooding. The constructions such as those by Damgård et al. [DOTT22], del Pino and Niot [dPN25], and Celi et al. [CdPE+26] essentially have each party derive their own partial signatures by rejection sampling. In this paper we continue improving this line of work.

Choosing the Nonce Distribution

In the original Dilithium paper [DLL+17], the authors introduced two versions of the scheme: one in which signatures are distributed uniformly in a hypercube, and another in which they are distributed per a Gaussian ("Dilithium-G"). The Gaussian variant was ultimately dropped, as the hypercube-based sampling was deemed easier to implement—an important consideration for a generalist signature scheme that is optimized for wide support. However, the Gaussian version in fact achieved significantly better parameters, at around 20% smaller signatures than the hypercube version [DFPS22]. Devevey et al. [DFPS22] established that in the Fiat-Shamir with Aborts paradigm, sampling signatures either from Discrete Gaussian or uniformly from a *hyperball* is asymptotically optimal. Moreover, the rejection condition for the hyperball variant is a simple L2 norm check.

Celi et al. [CdPE+26] translated the benefits of sampling from a hyperball to the threshold setting. In particular, they devised the Mithril scheme, wherein each party samples its partial signature locally within a hyperball, and the final signature is obtained by simply aggregating the partial signatures. As the partial signatures come from a hyperball, for the same security level, they are significantly smaller than what one would obtain by sampling within a hypercube. The implication is that aggregating the hyperball-sampled partial signatures for a handful parties yields a final signature that fits within the MLDSA hypercube, i.e. it respects the standard verifier's norm checks with high probability. Celi et al. were therefore able to obtain a threshold MLDSA signing protocol with a profile more reminiscent of traditional Schnorr signing—each party broadcasts a partial signature that roughly resembles an MLDSA signature itself. However, an important caveat is that a single signing attempt may not succeed; an honest signing attempt may fail due to an individual party rejecting its own partial signature, or the aggregate lying outside the MLDSA hypercube. In order to achieve a target success probability of say 99.9%, several instances of Mithril signing need to be executed concurrently or in sequence.

Devevey et al. argued that in the single signer case, one may as well employ the hyperball approach as it achieves essentially the same efficiency as Gaussians, but with a simpler rejection condition. Celi et al. constructed their threshold signing scheme with the hyperball approach based on this argument as well. However, Gaussians—thus far unexplored in the threshold MLDSA

setting—have proven powerful in a wide range of tasks in the literature, and offer a large toolkit currently absent for the uniform distribution in a ball. It is then natural to ask,

What benefits can Gaussian based sampling provide for threshold MLDSA *signing?*

This Work. We construct Pebble—a Gaussian based threshold MLDSA signing protocol. Our protocol realizes a natural template wherein each signing party contributes a Gaussian-sampled partial signature, so that when aggregated the resulting final signature follows a Gaussian distribution as well. This template enables the following benefits:

- **Clean(er) Analysis.** The hyperball approach taken in Mithril entails a complex, monolithic proof, with several heuristics and assumptions required to reason about the distributions of intermediate values. The Gaussian distributions that our partial and aggregated signatures follow are much easier to analyze, and so we are able to remove many of these assumptions and heuristic steps. This leads to a conceptually cleaner, more conservative proof on which the security of our protocol is based.
- **Associated Data Authentication (ADA).** Introduced in a recent work by Kondi et al. [KLLS26], ADA permits a signature to serve as a commitment to data that can be communicated out of band, without changing the format of the signature itself. This feature enables provable signing quorum identification in the context of permitting signing nodes to claim rewards for participating in signature generation. It was thus far unclear how ADA might be efficiently achieved for a standardized post-quantum signing scheme such as MLDSA. We believe this to serve as evidence that the analytical tools unlocked by Gaussian nonces can be further applied towards realizing advanced functionalities.

In regards to efficiency, our scheme performs slightly worse than Mithril, but is roughly in the same envelope for practical purposes. Concretely, in range 30-50% compared to 50% of [CdPE+26]. For example for success probability of 99% our scheme would require at most 35% more communication size, for more detailed comparison see §6.1. In many settings, this difference is an acceptable price to pay for the more conservative security analysis. We implement our threshold MLDSA scheme Pebble, along with a version Pebble-ADA that supports associated data authentication, and provide a comprehensive analytical and empirical comparison with prior works.

1.1 Our Contributions

Our primary technical contributions concern tools and techniques to analyze threshold MLDSA with Gaussian nonces. We detail them below:

– The standard MLDSA works over the ring $\mathcal{R}_q$ that splits fully hence it contains many ideals. This a notoriously difficult setting in which to bound the entropy of expressions such as $\mathbf{A} \cdot \mathbf{y} \bmod q$ and $\mathbf{A} \cdot \mathbf{y}_0 + \mathbf{y}_1 \bmod q$. The bounds degrade even further when arguing entropy of $\mathsf{HighBits}(\mathbf{A} \cdot \mathbf{y}_0 + \mathbf{y}_1 \bmod q)$. Hence, previous work employed lossy LWE search to decision reduction, or complex embedded LWE reductions. However, for the Discrete Gaussian case previous work [DLL+17] observed that $(\mathbf{y}_0, \mathbf{y}_1)$ are potentially wide enough to allow for a statistical argument. Indeed, unlike most distributions used in lattice-based cryptography, the Discrete Gaussian has fine-grained lower bounds on entropy modulo every ideal of the ring [JLWG25]. In this work we reduce the variance required in the Leftover Hash Lemma

compared to [LPR13] for matrices of the form $[\mathbf{I}\,\mathbf{A}]$ by extending the counting LHL techniques from [BL25] to matrices with prepended identity. This result allows us to prove $\mathbf{A}\cdot\mathbf{y}_0+\mathbf{y}_1 \bmod q$ is close to uniform in terms of Rényi Divergence and argue the entropy of $\mathsf{HighBits}(\mathbf{A}\cdot\mathbf{y}_0+\mathbf{y}_1)$. We believe this result is of independent interest. A remaining interesting question for future work would be to extend our entropy argument to simulate rejected transcripts in Threshold MLDSA, where we currently still rely on the MLWE assumption.

– In addition, in order to accommodate our associated data application, we provide a refined analysis of entropy in $\mathbf{A}\cdot\mathbf{y} \bmod q$, including taking into account noisy additive leakage on $\mathbf{y}$. This is also only currently known for Discrete Gaussian distributions thanks to the statistical arguments in [KLSS23].

– By design in Pebble-ADA the nonce $\mathbf{y}$ is constructed as $\mathbf{y} = \mathbf{y}_{\mathsf{pr}} + \mathbf{y}_{\mathsf{ad}}$ where $\mathbf{y}_{\mathsf{ad}}$ is revealed to the adversary with the opening of associated data. For this reason, the unforgeability of the underlying signature does not immediately follow from standard MLDSA. Instead, it is equivalent to MLDSA with additive noisy leakage $\mathbf{y}_{\mathsf{ad}} = \mathbf{y} - \mathbf{y}_{\mathsf{pr}}$. We provide a full proof of unforgeability of this leaky variant of MLDSA, which may be of independent interest. The same technique may apply to any noisy linear leakage on $\mathbf{y}$ i.e. $\mathbf{M}\cdot\mathbf{y}+\mathbf{y}_{\mathsf{pr}}$, although we leave the full proof of such an extension to future work.

– Lastly, if the leakage parameter is set to 0 our work contains a full unforgeability proof of MLDSA based in modern techniques such as smooth Rényi Divergence based rejection sampling. We believe that it can serve as a useful reference for the community.

2 Preliminaries

Notations. We let $\mathcal{R}$ denote the ring of integers of a cyclotomic number field of degree n where n is a power of 2. Then $\mathcal{R}$ is isomorphic to a polynomial ring $\mathbb{Z}[X]/(X^n+1)$. We also denote $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} \sim \mathbb{Z}_q[X]/(X^n+1)$, for a prime number q. For a ring element $a = \sum_{i=1}^{n} a_i X^i \in \mathcal{R}$ we denote its coefficient embedding as $\mathsf{coeff}(a) \coloneqq (a_1, \ldots, a_n)$. Given $\mathbf{x} \in \mathcal{R}^\ell$, we denote $\|\mathbf{x}\|_2$ the Euclidean norm and $\|\mathbf{x}\|_\infty$ the infinity norm of $\mathsf{coeff}(\mathbf{x}) \in \mathbb{R}^{n\ell}$. Unless specified otherwise, vectors are column vectors. For $n \in \mathbb{N}$ we write $[n] \coloneqq \{0, \ldots, n-1\}$. For $\eta > 0$ we denote a set of ring elements with coefficients bounded by η as $S_\eta \coloneqq \left\{a = \sum_{i=1}^{n} a_i X^i \in \mathcal{R} \mid |a_i| \le \eta\right\}$. For matrices $\mathbf{A}, \mathbf{B}$ we denote $\mathsf{diag}(\mathbf{A}, \mathbf{B}) \coloneqq \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}$.

Modular Reduction. For a positive integer m, let $r \bmod^{\pm} m$ denote the unique representative of r modulo m lying in the centered interval. More precisely, $r_0 = r \bmod^{\pm} m$ is the unique integer satisfying $r_0 \equiv r \pmod m$, where $-\frac{m}{2} < r_0 \le \frac{m}{2}$ if m is even, and $-\frac{m-1}{2} \le r_0 \le \frac{m-1}{2}$ if m is odd. Similarly, $r_0 = r \bmod^{+} m$ is the the unique representative satisfying $r_0 \equiv r \pmod m$ with $0 \le r_0 < m$.

Probabilities. For a finite set S we denote the a uniform distribution over S as $\mathcal{U}(S)$ we also use a shorthand $x \leftarrow S$ for sampling x according to $\mathcal{U}(S)$. For a real parameter $0 \le p \le 1$, the Bernoulli distribution which outputs 1 with probability p and 0 otherwise is denoted by $\mathsf{Ber}(p)$. For two discreet distributions $\mathcal{P}, \mathcal{Q}$ over the domain Ω, we define their statistical distance as

$\mathrm{SD}(\mathcal{P},\mathcal{Q}) = \frac{1}{2}\sum_{x\in\Omega}|\mathcal{P}(x)-\mathcal{Q}(x)|$. When $\mathrm{Supp}(\mathcal{P}) \subseteq \mathrm{Supp}(\mathcal{Q})$, we define Rényi Divergence of order two as $\mathrm{RD}_2(\mathcal{P}\|\mathcal{Q}) = \sum_{x\in\mathrm{Supp}(\mathcal{P})}\frac{\mathcal{P}^2(x)}{\mathcal{Q}(x)}$. Then for any set $E \subseteq \mathrm{Supp}(\mathcal{P})$, we have the probability preservation property of the form $\mathcal{P}(E) \leq \sqrt{\mathcal{Q}(E)\cdot\mathrm{RD}(\mathcal{P}\|\mathcal{Q})}$. We define the Rényi Divergence of infinite order as $\mathrm{RD}_\infty(\mathcal{P}\|\mathcal{Q}) = \sup_{x\in\mathrm{Supp}(\mathcal{P})}\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}$. The corresponding probability preservation property then becomes $\mathcal{P}(E) \leq \mathcal{Q}(E)\cdot\mathrm{RD}_\infty(\mathcal{P}\|\mathcal{Q})$. For $\varepsilon > 0$ we extend the notion above to smooth Rényi Divergence that excludes anomalous points that occur with a bounded probability

$$\mathrm{RD}^{\varepsilon}_{\infty}(\mathcal{P}||\mathcal{Q}) = \inf_{\substack{S\subset\mathrm{Supp}(\mathcal{Q})\\ \Pr(\mathcal{P}(S)>1-\varepsilon)}}\max_{x\in S}\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}.$$

Lattices. A lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. Any submodule $\mathcal{M}$ of the ring $\mathcal{R}^k$ defines a lattice via the coefficient embedding $\mathsf{coeff}(\mathcal{M}) \subset \mathbb{R}^{n\cdot k}$. Let $\sigma > 0$, let $\mathbf{c} \in \mathbb{R}^m$ and Λ be a full-rank m-dimensional lattice. For any vector $\mathbf{x} \in \mathbb{R}^m$ we define the Gaussian mass function $\rho_{\sigma,\mathbf{c}}(\mathbf{x})$ as $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) \coloneqq \exp\left(-\frac{\pi\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$. We define the discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ over Λ as $\forall\mathbf{x}\in\Lambda : \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(\Lambda)$. When $\mathbf{c} = \mathbf{0}$, we omit it from the notation.

Lemma 1 ([Ban93, Lem 1.5])**.** *Let $m > 0$, $c > 1/\sqrt{2\pi}$, $\sigma > 0$ and $\Lambda \subseteq \mathbb{R}^m$ be a a full-rank lattice. Let $\mathbf{x} \leftarrow \mathcal{D}_{\Lambda,\sigma}$. Then*

$$\Pr(\|\mathbf{x}\| > c\cdot\sigma\cdot\sqrt{m}) \leq C^m$$

where $C \coloneqq c\cdot\sqrt{2\pi e}\cdot e^{-\pi c^2} < 1$.

Corollary 1. *We instantiate the lemma for $m = 8\cdot\lambda$ and $c = 0.53$. Then*

$$(c\cdot\sqrt{2\pi e}\cdot e^{-\pi c^2})^{8\cdot\lambda} \leq 0.455^{\lambda} < 2^{-\lambda}$$

Lemma 2 (simplified [BL25, Cor. 5])**.** *Let $\mathcal{R} = \mathbb{Z}[\zeta_{2n}] = \mathbb{Z}[X]/(X^n+1)$ and modulus $q \geq 1$ be such that $\mathcal{R}_q = \prod_{i=1}^{n}\mathbb{Z}[X]/(X-r_i)$ for some $r_i \in \mathbb{Z}_q$. Let $k,\ell > 0$, $\sigma \leq \frac{\sqrt{\pi}}{2}\cdot\frac{q}{\sqrt{\ln(4\ell n)}}$. Let $\mathbf{c} \in \mathcal{K}^{\ell}_{\mathbb{R}}$, $\mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^{\ell},\sigma,\mathbf{c}}$*

$$\mathrm{RD}((\mathbf{A},\mathbf{A}\cdot\mathbf{y} \bmod q),(\mathbf{A},\mathbf{u}\leftarrow\mathcal{U}(\mathcal{R}_q^k))) \leq 2\cdot\left(\frac{q^k}{\sigma^{\ell}}+1\right)^n.$$

Lemma 3 ([KLSS23, Lemma 7])**.** *Let $\ell,\sigma_0,\sigma_1 > 0$ define $\sigma_2 \coloneqq \left(1/\sigma_0^2+1/\sigma_1^2\right)^{-1/2}$. Then the distributions*

$$\mathcal{D}_0 = \left\{(\mathbf{y},\mathbf{y}+\mathbf{y}') \mid \mathbf{y}\leftarrow\mathcal{D}_{\mathcal{R}^{\ell},\sigma_0},\mathbf{y}'\leftarrow\mathcal{D}_{\mathcal{R}^{\ell},\sigma_1}\right\}$$
$$\mathcal{D}_1 = \left\{(\hat{\mathbf{y}}',\mathbf{y}+\mathbf{y}') \mid \mathbf{y}\leftarrow\mathcal{D}_{\mathcal{R}^{\ell},\sigma_0},\mathbf{y}'\leftarrow\mathcal{D}_{\mathcal{R}^{\ell},\sigma_1},\hat{\mathbf{y}}'\leftarrow\mathcal{D}_{\mathcal{R}^{\ell},\sigma_2,\mathbf{c}},\mathbf{c} = \frac{\sigma_2}{\sigma_1}\cdot(\mathbf{y}+\mathbf{y}')\right\}$$

are identical.

The following is an immediate corollary of Lemma 2 and Lemma 3.

Corollary 2. *Let $\sigma_0, \sigma_1 > 0$ define $\sigma_2 := \left(1/\sigma_0^2 + 1/\sigma_1^2\right)^{-1/2}$. Let $\mathcal{R}_q = \prod_{i=1}^{n} \mathbb{Z}[X]/(X - r_i)$ for some $r_i \in \mathbb{Z}_q$. Let $k, \ell > 0$, $\sigma_2 \leq \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4\ell n)}}$. For $\mathbf{A} \leftarrow \mathcal{R}_q^{k \times \ell}$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_0}$ and $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_1}$*

$$\mathrm{RD}((\mathbf{A}, \mathbf{A} \cdot \mathbf{y} \bmod q, \mathbf{y} + \mathbf{y}'), (\mathbf{A}, \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^k), \mathbf{y} + \mathbf{y}')) \leq 2 \cdot \left(\frac{q^k}{\sigma_2^\ell} + 1\right)^n.$$

Lemma 4 ([Pei10])**.** *Let $m, \sigma_0, \sigma_1 > 0$, $\varepsilon \in [0, 1/2)$ s.t. $\sigma_0 \geq \eta_\varepsilon(\mathcal{R}^m)$ and $\left(1/\sigma_0^2 + 1/\sigma_1^2\right)^{-1/2} \geq \eta_\varepsilon(\mathcal{R}^m)$. Then for $\mathbf{y}_i \leftarrow \mathcal{D}_{\mathcal{R}^m, \sigma_i}$, $i = 0, 1$ it holds*

$$\mathrm{SD}(\mathbf{y}_0 + \mathbf{y}_1, \mathcal{D}_{\mathcal{R}^m, \sqrt{\sigma_0^2 + \sigma_1^2}}) \leq 8\varepsilon.$$

Lemma 5 (adapted [Pei08, Corollary 5.3])**.** *Let Λ be an n-dimensional lattice and $\sigma > 0$. Then, for $\mathbf{y} \leftarrow \mathcal{D}_{\Lambda, \sigma}$*

$$\Pr[\|\mathbf{y}\|_\infty > \sigma \cdot t] \leq 2en \cdot e^{-\pi t^2}.$$

Lemma 6 (adapted [Lyu12, Lemma 4.3])**.** *Let Λ be a n-dimensional lattice. Let $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{z} \leftarrow \mathcal{D}_{\Lambda, \sigma}$ then*

$$\Pr\left(\left|\mathsf{coeff}(\mathbf{z})^\intercal \cdot \mathsf{coeff}(\mathbf{v})\right| > \sigma \|\mathbf{v}\| \cdot t/\sqrt{2\pi}\right) \leq 2e^{-t^2/2}.$$

In particular, if $t \geq \sqrt{2(\kappa + 1)\ln 2}$, then the RHS is at most $2^{-\kappa}$.

Short Secret Sharing The secret sharing algorithm RSS from [PN25], considers every subset $I \subset [N]$ of size $N - T + 1$. For each such subset I, it samples an indepedent share $\mathbf{s}_I$, i.e., total $\binom{N}{N-T+1}$ small shares $\mathbf{s}_I$'s. Each $\mathbf{s}_I$ is then distributed to all parties in the corresponding I. Doing so ensures that, atleast one of these secret $\mathbf{s}_I$ remains hidden from corrupted parties, since atleast one of the sets I is fully non-corrupted (atmost $T - 1$ corrupted parties). The algorithm then outputs all the partial secret keys $(\mathsf{sk}_i)_{i \in [N]}$ (each containing $\binom{N-1}{N-T}$ shares) and the master secret $\mathbf{s} = \sum_I \mathbf{s}_I$.

To reconstruct the secret from an active set of parties $\mathsf{act} \subseteq [N]$, the algorithm $\mathsf{RSSRecover}$ computes a partition $\mathbf{m} = (m_i)_{i \in \mathsf{act}}$ of all replicated shares, where $\sqcup_{i \in \mathsf{act}} m_i = \{I \subset [N] \mid |I| = N - T + 1\}$. The partition assigns each replicated share $\mathbf{s}_I$ to exactly one active party (largest index in $I \cap \mathsf{act}$) that possesses it.

$\mathsf{RSS}(N, T, \chi) \to (\mathbf{s}, \mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_N))$

1: **for** $I \subset [N]$ s.t. $|I| = N - T + 1$ **do**
2: $\quad \mathbf{s}_I \leftarrow \chi$
3: $\mathbf{s} := \sum_I \mathbf{s}_I$
4: **for** $i \in [N]$ **do**
5: $\quad \mathsf{sk}_i \leftarrow \{I : \mathbf{s}_I \mid I \subset [N] \text{ s.t. } i \in I \wedge |I| = N - T + 1\}$
6: **return** $(\mathbf{s}, \mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_N))$

$\mathsf{RSSRecover}(\mathsf{act}) \to (m_i)_{i \in \mathsf{act}}$

1: $\mathbf{m} := (m_i)_{i \in \mathsf{act}} = (\varnothing)^{\mathsf{act}}$
2: **for** $I \subset [N]$ s.t. $|I| = N - T + 1$ **do**
3: $\quad i_0 := \max(I \cap \mathsf{act})$
4: $\quad m_i := m_i \sqcup \{I\}$
5: **return** $\mathbf{m}$

Figure 1: Short Secret Sharing algorithms

2.1 Rejection Sampling

Given two distributions χ_r and χ_z and a parameter $M \geq 1$, we define the following rejection sampling algorithms $\mathsf{A}^{\mathsf{real}}, \mathsf{A}^{\mathsf{ideal}}, \mathsf{B}^{\mathsf{real}}, \mathsf{B}^{\mathsf{ideal}}$ in Figure 2.

Algorithm $\mathsf{A}^{\mathsf{real}}$

1: $z \leftarrow \chi_r$
2: $b \leftarrow \mathsf{Ber}\left(\min\left(\frac{\chi_z(z)}{M\chi_r(z)}, 1\right)\right)$
3: **if** $b = 0$ **then** $z = \perp$
4: **return** z

Algorithm $\mathsf{A}^{\mathsf{ideal}}$

1: $z \leftarrow \chi_z$
2: $b \leftarrow \mathsf{Ber}\left(\frac{1}{M}\right)$
3: **if** $b = 0$ **then** $z = \perp$
4: **return** z

Algorithm $\mathsf{B}^{\mathsf{real}}$

1: $x \leftarrow \perp$
2: **while** $x = \perp$ **do** $x \leftarrow \mathsf{A}^{\mathsf{real}}$
3: **return** x

Algorithm $\mathsf{B}^{\mathsf{ideal}}$

1: $x \leftarrow \perp$
2: **while** $x = \perp$ **do** $x \leftarrow \mathsf{A}^{\mathsf{ideal}}$
3: **return** x

Figure 2: Rejection sampling algorithms.

Lemma 7 ([DFPS22, Lemma 4.1])**.** *Assume that $M > 1$ and $\varepsilon < 1$ are s.t. $RD^{\varepsilon}_{\infty}(\chi_z||\chi_r) \leq M$, then $RD_{\infty}(\mathsf{A}^{\mathsf{real}}||\mathsf{A}^{\mathsf{ideal}}) \leq 1 + \frac{\varepsilon}{M-1}$, $\Pr(\mathsf{A}^{\mathsf{real}}(\perp)) \leq \frac{M-1+\varepsilon}{M}$ and $RD_{\infty}(\mathsf{B}^{\mathsf{real}}||\mathsf{B}^{\mathsf{ideal}}) \leq \frac{1}{1-\varepsilon}$.*

Remark 1. *For $\mathbf{v}, \mathbf{z} \in \mathcal{R}^m$, $M, \sigma > 0$, and $\chi_r = \mathcal{D}_{\mathcal{R}^m,\sigma,\mathbf{v}}$, $\chi_z = \mathcal{D}_{\mathcal{R}^m,\sigma,\mathbf{0}}$ we get*

$$\frac{\chi_z(\mathbf{z})}{M \cdot \chi_r(\mathbf{z})} = \frac{\exp(-\pi\|\mathbf{z}\|^2/\sigma^2)}{M \cdot \exp(-\pi\|\mathbf{z}-\mathbf{v}\|^2/\sigma^2)} = \frac{1}{M} \cdot \exp\left(\frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z})^{\mathsf{T}} \cdot \mathsf{coeff}(\mathbf{v}) + \pi \cdot \|\mathbf{v}\|^2}{\sigma^2}\right)$$

Lemma 8 ([Lyu12, Le. 4.5])**.** *For $\mathbf{v}, \mathbf{z} \in \mathcal{R}^m$, and $\chi_r = \mathcal{D}_{\mathcal{R}^m,\sigma,\mathbf{v}}$, $\chi_z = \mathcal{D}_{\mathcal{R}^m,\sigma,\mathbf{0}}$ with $\sigma = t \cdot \sqrt{2\pi}\|\mathbf{v}\|$ we get*

$$RD^{2^{-64}}_{\infty}(\chi_z||\chi_r) \leq e^{9.5/t + /2t^2}.$$

Proof. Similar to Remark 1 the Rényi Divergence between given distribution is by definition

$$\begin{aligned}
RD^{2^{-64}}_{\infty}(\chi_z||\chi_r) &= \exp\left(\frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z})^{\mathsf{T}} \cdot \mathsf{coeff}(\mathbf{v}) + \pi \cdot \|\mathbf{v}\|^2}{\sigma^2}\right) \\
&\leq \exp\left(\frac{\sqrt{4\pi(\kappa+1)\ln 2} \cdot \sigma\|\mathbf{v}\| + \pi \cdot \|\mathbf{v}\|^2}{\sigma^2}\right) \\
&\leq \exp\left(\frac{9.5 \cdot \sigma(\sqrt{2\pi}\|\mathbf{v}\|) + (\sqrt{2\pi} \cdot \|\mathbf{v}\|)^2/2}{\sigma^2}\right) \\
&\leq e^{9.5/t + 1/2t^2}
\end{aligned}$$

where in the first inequality we apply Lemma 6, the second inequality holds with probability at least $1 - 2^{-\kappa}$ for $\kappa = 64$, and in the last inequality we substitute $\sigma = t \cdot \sqrt{2\pi}\|\mathbf{v}\|$. □

2.2 Hardness Assumptions

Definition 1 (MLWE). *Let k, ℓ, q be integers, and $\mathcal{D} : \mathcal{R}_q \to [0,1]$ be a probability distribution, we define the* $\mathsf{MLWE}_{k,\ell,\mathcal{D}}$ *problem over the ring $\mathcal{R}_q$ as follows:*

$$\begin{aligned}&|\Pr\big[b = 1 : \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}; \mathbf{t} \leftarrow \mathcal{R}_q^k; b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{t})\big]\\ &- \Pr\big[b = 1 : \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}; \mathbf{s}_1 \leftarrow \mathcal{D}^\ell; \mathbf{s}_2 \leftarrow \mathcal{D}^k; b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)\big]\,| \le \mathsf{negl}(\lambda).\end{aligned}$$

Definition 2 (SelfTargetMSIS). *Suppose that $\mathsf{H} : \{0,1\}^* \to \mathcal{B}_\tau$ is a cryptographic hash function. We define the* $\mathsf{SelfTargetMSIS}_{\mathsf{H},k,\ell,\gamma}$ *problem as follows:*

$$\Pr\left[\begin{array}{l}0 \le \|\mathbf{y}\|_\infty \le \gamma,\\ \wedge\ \mathsf{H}(\mu \,\|\, [\mathbf{I} \mid \mathbf{A}] \cdot \mathbf{y})\end{array} \;:\; \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}; \left(\mathbf{y} := \begin{pmatrix}\mathbf{r}\\ c\end{pmatrix}, \mu\right) \leftarrow \mathcal{A}^{\mathsf{H}(\cdot)}(\mathbf{A}) \right] \le \mathsf{negl}(\lambda).$$

2.3 Cryptographic Primitives

Definition 3 (Digital Signatures). *We define a digital signature scheme as a tuple of PPT algorithms*

$$\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$$

defined as follows:

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$*: On input the security parameter λ, outputs a public/secret key pair.*
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{m}) \to \sigma$*: On input a secret key* sk*, a message $\mathsf{m} \in \mathcal{M}$, outputs a signature σ.*
- $\mathsf{Verify}(\mathsf{pk}, \mathsf{m}, \sigma) \to \{0,1\}$*: On input a public key* pk*, a message* m*, and a signature σ, outputs 1 if the signature is valid and 0 otherwise.*

Definition 4 (Identification Scheme). *An identification scheme* ID *is a tuple of algorithms* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ *defined as follows:*

- *The key generation algorithm* $\mathsf{IGen}(1^\lambda)$ *outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*.*
- *The prover consists of two algorithms:* P_1 *takes* sk *as input and returns a commitment W and state* St*; where* P_2 *on input (sk, W, c, St) returns a response Z or $\perp$ (abort).*
- *The verifier algorithm* V *takes the* pk *and the conversation transcript as input and outputs a decision bit 1 (accept) or 0 (reject).*

A transcript of the protocol is a tuple (W, c, Z). It is called valid if $\mathsf{V}(\mathsf{pk}, W, c, Z) = 1$. In §2.3 we define a transcript oracle $\mathsf{Trans}(\mathsf{sk})$ *that returns a real interaction (W, c, Z) between prover and verifier. The* $\mathsf{Trans}(\mathsf{sk})$ *oracle models a single execution of the identification protocol* ID *and may abort by returning $\perp$. In contrast, the signing algorithm repeats the protocol until a non-aborting transcript is obtained. To capture the distribution of transcripts actually exposed to the adversary, we define the oracle* $\mathsf{AcceptedTrans}(\mathsf{sk})$ *in §2.3, which repeatedly invokes* $\mathsf{Trans}(\mathsf{sk})$ *until a valid transcript is produced.*

Definition 5 (Rényi No-Abort Honest-Verifier Zero-Knowledge (naHVZK)). *An identification scheme* ID *is said to be $\varepsilon_{\mathsf{zk}}$-perfect* naHVZK *if there exists an algorithm* Sim *that, given only the* pk*, outputs a transcript (W, c, Z) such that the following conditions hold:*

Trans(sk)

1: $(W, \mathsf{St}) \leftarrow \mathsf{P}_1(\mathsf{sk})$
2: $c \leftarrow \mathcal{C}$
3: $Z \leftarrow \mathsf{P}_2(\mathsf{sk}, W, c, \mathsf{St})$
4: If $\mathrm{Z} = \perp$ then return $(\perp, \perp, \perp)$
5: **return** (W, c, Z)

AcceptedTrans(sk)

1: $Z := \perp$
2: **while** $Z = \perp$ **do**
3: $\quad (W, c, Z) \leftarrow \mathsf{Trans}(\mathsf{sk})$
4: **return** (W, c, Z)

Figure 3: Transcript algorithms.

- *The distribution of* $(W, c, Z) \leftarrow \mathsf{Sim}(\mathsf{pk})$ *has Rényi Divergence with the real* $\mathsf{AcceptedTrans}(\mathsf{sk})$ *defined in §2.3 bounded by* $\mathrm{RD}(\mathsf{Sim}(\mathsf{pk}) \| \mathsf{AcceptedTrans}(\mathsf{sk})) \leq 1 + \varepsilon_{\mathsf{zk}}$.
- *The distribution of* c *in* $(W, c, Z) \leftarrow \mathsf{Sim}(pk)$*, is uniform over* $\mathcal{C}$.

Definition 6 (UF-NMA/EUF-CMA Security)**.** *Let* $\mathcal{O}_{\mathsf{Sign}}$ *denote either the signing oracle* $\mathsf{Sign}(\mathsf{sk}, \cdot)$ *or the empty oracle* $\varnothing$*. We define the unforgeable experiment of signature scheme* Π *for every PPT adversary* $\mathcal{A}$ *as:*

$$\Pr\left[\begin{array}{ll} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) & \mathsf{Verify}(\mathsf{pk}, \mathsf{m}^*, \sigma^*) = 1 \wedge \\ (\mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pk}) & : \quad \mathsf{m}^* \notin Q_{\mathcal{A}} \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the signing oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$*, on input* m*, returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$*, and* $Q_{\mathcal{A}}$ *denotes the set of messages queried to the signing oracle. The scheme is said to be*

- UF-NMA secure *if* $\mathcal{O}_{\mathsf{Sign}} = \varnothing$*;*
- EUF-CMA secure *if* $\mathcal{O}_{\mathsf{Sign}} = \mathsf{Sign}(\mathsf{sk}, \cdot)$.

Definition 7 (Pseudorandomness)**.** *Let* $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^k$ *be a function, where* $\mathcal{K}$ *is the key space and* n, k *are integers. We say* PRF *is pseudorandom if for every PPT adversary* $\mathcal{A}$*,*

$$\left|\Pr\left[K \leftarrow \mathcal{K} : \mathcal{A}^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{RF}(\cdot)}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{RF} : \{0,1\}^n \to \{0,1\}^k$ *denotes a perfect random function.*

Definition 8 (Correctness with Abort [CdPE+26, Def 2.2])**.** *For* $p > 0$ *and* $N \geq T \geq 1$ *we say that a* (N, T)*-Threshold Signing Scheme* Π_{TSIG} *p-terminates if for any message* m *and any set of* T *signers* $\mathsf{act} \subset [N]$ *we have*

$$\Pr(\mathsf{Exp}^{\mathsf{TS\text{-}corr}}_{\Pi_{\mathsf{TSIG}}}(1^\lambda, \mathsf{m}, \mathsf{act}) \neq \perp) \geq p$$

and $\Pr(\mathsf{Exp}^{\mathsf{TS\text{-}corr}}_{\Pi_{\mathsf{TSIG}}}(\lambda, \mathsf{m}, \mathsf{act}) = 0) \leq \mathsf{negl}(\lambda)$ *for* $\mathsf{Exp}^{\mathsf{TS\text{-}corr}}_{\Pi_{\mathsf{TSIG}}}(\lambda, \mathsf{m}, \mathsf{act})$ *defined in Figure 4b.*

Definition 9 (Threshold Signature Selective Unforgeability)**.** *Let* $N \geq T \geq 1$ *we say that a* (N, T)*-Threshold Signing Scheme* Π_{TSIG} *satisfies Unforgeability with selective corruptions if for any PPT adversary* $\mathcal{A}$ *it holds* $\Pr(\mathsf{Exp}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{TSIG}}, \mathcal{A}}(1^\lambda) = 1) \leq \mathsf{negl}(\lambda)$ *for* $\mathsf{Exp}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{TSIG}}, \mathcal{A}}(1^\lambda)$ *defined in Figure 4a. We denote the winning probability of* $\mathcal{A}$ *as* $\mathsf{Adv}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{TSIG}}}(\mathcal{A})$.

$\mathsf{Exp}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{TSIG}},\mathcal{A}}(1^\lambda)$

1: $\mathsf{cor} \leftarrow \mathcal{A}$
2: **assert**$\{|\mathsf{cor}| < T\}$
3: $(\mathsf{pk}, \{\mathsf{sk}_i\}) \leftarrow \Pi_{\mathsf{TSIG}}.\mathsf{KeyGen}(1^\lambda)$
4: $(\mathsf{sig}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{Sign}_r(\mathsf{sk}_i,\cdot)}(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathsf{cor}})$
5: **return** $\Pi_{\mathsf{TSIG}}.\mathsf{Verify}(\mathsf{pk}, \mathsf{sig}^*, \mathsf{m}^*)$

(a) Threshold Signature unforgeability with selective corruptions.

$\mathsf{Exp}^{\mathsf{TS\text{-}corr}}_{\Pi_{\mathsf{TSIG}}}(1^\lambda, \mathsf{m}, \mathsf{act})$

1: $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[N]}) \leftarrow \Pi_{\mathsf{TSIG}}.\mathsf{KeyGen}(1^\lambda)$
2: **for** $r \in [3]$: **do**
3: $\quad \left\{\mathsf{contrib}_i^{(r)}\right\}_{i\in\mathsf{act}} \leftarrow \Pi_{\mathsf{TSIG}}.\mathsf{Sign}_r(\mathsf{sk}_i, \mathsf{m}, \mathsf{act})$
4: **if** $\exists i \in \mathsf{act} : \mathsf{contrib}_i^{(2)} = \perp$ **then return** $\perp$
5: $\mathsf{sig} = \Pi_{\mathsf{TSIG}}.\mathsf{Combine}(\mathsf{pk}, \mathsf{m}, \mathsf{act}, \left\{\mathsf{contrib}_i^{(r)}\right\}_{i,r})$
6: **return** $\Pi_{\mathsf{TSIG}}.\mathsf{Verify}(\mathsf{pk}, \mathsf{m}, \mathsf{sig})$

(b) Threshold Signature correctness with abort.

Figure 4: Security experiments for the threshold signature scheme.

2.4 Signatures with Associated Data

In this section, we detail the syntax and the security properties of Signatures with Associated Data from [KLLS26].

Definition 10 (ADA Signatures)**.** *We define a digital signature scheme with associated data (*ad*) as a tuple of PPT algorithms*

$$\Pi_{\mathsf{ADA}} = (\mathsf{KeyGen}, \mathsf{Sign\text{-}AD}, \mathsf{Verify}, \mathsf{Verify\text{-}AD})$$

defined as follows

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$*: On input the security parameter* λ*, outputs a public/secret key pair.*
- $\mathsf{Sign\text{-}AD}(\mathsf{pk}, \mathsf{sk}, \mathsf{m}, \mathsf{ad}) \to (\sigma, \pi)$*: On input a secret key* sk*, a message* $\mathsf{m} \in \mathcal{M}$*, and associated data* $\mathsf{ad} \in \mathcal{A}$*, outputs a signature* σ *together with an associated proof* π*.*
- $\mathsf{Verify}(\mathsf{pk}, \mathsf{m}, \sigma) \to \{0, 1\}$*: On input a public key* pk*, a message* m*, and a signature* σ*, outputs* 1 *if the signature* σ *is valid and* 0 *otherwise.*
- $\mathsf{Verify\text{-}AD}(\mathsf{pk}, \mathsf{m}, \mathsf{ad}, \sigma, \pi) \to \{0, 1\}$*: On input a public key* pk*, a message* m*, associated data* ad*, and a pair* (σ, π)*, outputs* 1 *if the signature* σ *and the proof* π *both are valid and* 0 *otherwise.*

We define the security properties required from an ADA signature scheme Π_{ADA}. Intuitively, the signature σ acts as a commitment to the associated data ad, while the proof π serves as an opening of this commitment.

Definition 11 (EUF-CMA with Associated Data Security)**.** *We say that* Π_{ADA} *is existentially unforgeable under chosen message attacks if for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{ll} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) & \\ (\mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign\text{-}AD}(\mathsf{sk},\cdot,\cdot)}(\mathsf{pk}) & : \begin{array}{l} \mathsf{Verify}(\mathsf{pk}, \mathsf{m}^*, \sigma^*) = 1 \wedge \\ \mathsf{m}^* \notin Q_{\mathcal{A}} \end{array} \end{array}\right] \leq \mathsf{negl}(\lambda)$$

where oracle $\mathsf{Sign\text{-}AD}(\mathsf{sk}, \cdot, \cdot)$ *on input* $(\mathsf{m}, \mathsf{ad})$ *returns* $(\sigma, \pi) \leftarrow \mathsf{Sign\text{-}AD}(\mathsf{sk}, \mathsf{m}, \mathsf{ad})$ *and* $Q_{\mathcal{A}}$ *is the set of* $(\mathsf{m}, \mathsf{ad})$ *pairs queried to the signing oracle. We denote the winning probability of* $\mathcal{A}$ *as* $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{ADA}}}(\mathcal{A})$*.*

Definition 12 (Weak AD-Binding)**.** *We say that* Π_{ADA} *satisfies weak associated data binding if for every PPT adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{ad} \neq \mathsf{ad}' \wedge \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) & \mathsf{Verify\text{-}AD}(\mathsf{pk},\mathsf{m},\mathsf{ad},\sigma,\pi) = 1 \wedge \\ (\mathsf{m},\sigma,\mathsf{ad},\pi,\mathsf{ad}',\pi') \leftarrow \mathcal{A}(\mathsf{pk},\mathsf{sk}) & \mathsf{Verify\text{-}AD}(\mathsf{pk},\mathsf{m},\mathsf{ad}',\sigma,\pi') = 1 \end{array}\right] \leq \mathsf{negl}(\lambda).$$

We denote the winning probability of $\mathcal{A}$ *as* $\mathsf{Adv}^{\mathsf{Weak\text{-}AD\text{-}Binding}}_{\Pi_{\mathsf{ADA}}}(\mathcal{A})$*.*

Definition 13 (Strong AD-Binding)**.** *We say that* Π_{ADA} *satisfies strong associated data binding if for every PPT adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{ad} \neq \mathsf{ad}' \wedge \\ (\mathsf{pk},\mathsf{m},\sigma,\mathsf{ad},\pi,\mathsf{ad}',\pi') \leftarrow \mathcal{A}(1^\lambda) & \mathsf{Verify\text{-}AD}(\mathsf{pk},\mathsf{m},\mathsf{ad},\sigma,\pi) = 1 \wedge \\ & \mathsf{Verify\text{-}AD}(\mathsf{pk},\mathsf{m},\mathsf{ad}',\sigma,\pi') = 1 \end{array}\right] \leq \mathsf{negl}(\lambda).$$

We denote the winning probability of $\mathcal{A}$ *as* $\mathsf{Adv}^{\mathsf{Strong\text{-}AD\text{-}Binding}}_{\Pi_{\mathsf{ADA}}}(\mathcal{A})$*.*

Definition 14 (AD-Hiding)**.** *We say that* Π_{ADA} *satisfies associated data hiding if for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{m},\mathsf{ad}_0,\mathsf{ad}_1) \leftarrow \mathcal{A}(\mathsf{pk},\mathsf{sk}) \\ b \leftarrow \{0,1\}; (\sigma,\pi) \leftarrow \mathsf{Sign\text{-}AD}(\mathsf{sk},\mathsf{m},\mathsf{ad}_b) \\ b' \leftarrow \mathcal{A}(\sigma) \end{array} : b' = b\right] \leq \tfrac{1}{2} + \mathsf{negl}(\lambda)$$

where $|\mathsf{ad}_0| = |\mathsf{ad}_1|$*. We denote the winning probability of* $\mathcal{A}$ *as* $\mathsf{Adv}^{\mathsf{AD\text{-}Hiding}}_{\Pi_{\mathsf{ADA}}}(\mathcal{A})$*.*

Definition 15 (Soundness)**.** *We say that* Π_{ADA} *is sound if for every honestly generated key pair* $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, every message* $\mathsf{m} \in \mathcal{M}$ *and associated data* $\mathsf{ad} \in \mathcal{A}$*, an honestly generated signature-proof pair* $(\sigma,\pi) \leftarrow \mathsf{Sign\text{-}AD}(\mathsf{sk},\mathsf{m},\mathsf{ad})$ *is accepted by the verification algorithm except with negligible probability, i.e.,*

$$\Pr[\mathsf{Verify\text{-}AD}(\mathsf{pk},\mathsf{m},\mathsf{ad},\sigma,\pi) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

2.5 ML-DSA with Gaussian Nonce

In this subsection we describe a version of ML-DSA with Gaussian nonce similar to Dilithium-G from [DLL+17]. We adjust selected algorithms of the scheme, since we require compatibility with [NIS24] verification.

Design choices. Our version of Dilithium-G differs slightly from [DLL+17] since one of our objectives is staying compatible with the standard ML-DSA parameters, verification and key generation algorithms. First, we do not modify the MakeHint and UseHint algorithms. Similarly, we preserve the original verification norm bounds of [NIS24], the choice of the ring, modulus and other parameters stated in Table 1. We detail the algorithms in full in Figure 6.
Note that distribution of the nonce in Figure 6 differs from [NIS24] in two aspects. First, we sample a vector in $(\mathbf{y}_1,\mathbf{y}_2) \in \mathcal{R}^{\ell+k}$ instead of $\mathbf{y}_1 \in \mathcal{R}^\ell$ and adjust the value $\mathbf{w}$ to $\mathbf{w} = \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2 \bmod q$.

Parameter	MLDSA-44	MLDSA-65	MLDSA-87
q	8380417	8380417	8380417
n	256	256	256
d	13	13	13
τ	39	49	60
γ_1	2^{17}	2^{19}	2^{19}
γ_2	95232	261888	261888
(k, ℓ)	$(4, 4)$	$(6, 5)$	$(8, 7)$
η	2	4	2
β	78	196	120
w	80	55	75

Table 1: ML-DSA parameter sets [NIS24].

This is because we use Discrete Gaussian rejection sampling in Algorithm 2 for both key dependent values $\mathbf{z}_1$ and $\mathbf{z}_2$, so $\mathbf{z}_2$ requires an additional masking vector. The rejection condition is modified accordingly.

Second, the norm checks in Line 17 ensure correctness of the signature. The check on $\mathbf{e}$ guarantees that $\mathsf{UseHint}$ returns a correct value following conditions of Lemma 9 stated below. The check on $\mathbf{h}$ ensures the storage format of the signature is compatible with [NIS24] and the check on $\mathbf{z}$ corresponds to the norm condition of the standard ML-DSA verification algorithm. For this reason any signature returned by Algorithm 2 is guaranteed to pass the verification check. Hence, we only analyse the expected runtime of the signing protocol until a valid signature is returned.

The construction uses a number of supporting algorithms introduced in Figure 5. Then the parameters of the scheme are set up to satisfy the following properties.

Lemma 9 ([KLS18, Lemma 4.1])**.** *Suppose that q and α are positive integers satisfying $q > 2\alpha$, $q \equiv 1 \pmod{\alpha}$, and α is even. Let $\mathbf{r}$ and $\mathbf{z}$ be vectors of elements in R_q such that $\|\mathbf{z}\|_\infty \leq \alpha/2$, and let $\mathbf{h}, \mathbf{h}'$ be vectors of bits. Then the $\mathsf{HighBits}_q$, $\mathsf{MakeHint}_q$, and $\mathsf{UseHint}_q$ algorithms satisfy the following properties:*

1. $\mathsf{UseHint}_q\big(\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha), \mathbf{r}, \alpha\big) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \alpha)$.
2. *Let $\mathbf{v}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha)$. Then $\|\mathbf{r} - \mathbf{v}_1 \cdot \alpha\|_\infty \leq \alpha + 1$. In addition, if $\|h\|_1 = w$, then all but at most w coefficients in $\mathbf{r} - \mathbf{v}_1 \cdot \alpha \bmod q$ have magnitude at most $\alpha/2$.*
3. *For any $\mathbf{h}, \mathbf{h}'$, if $\mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha) = \mathsf{UseHint}_q(\mathbf{h}', \mathbf{r}, \alpha)$, then $\mathbf{h} = \mathbf{h}'$.*

Security. We note that the original security proof of Dilithium-G applies to the variant we present here with little modification. It can be derived following the arguments in [DLL+17] and [Lyu12], we do not detail here. The expected runtime of Algorithm 2 can be derived from Lemma 7 and Lemma 8 together with norm bounds similar to the discussion in §6.1.

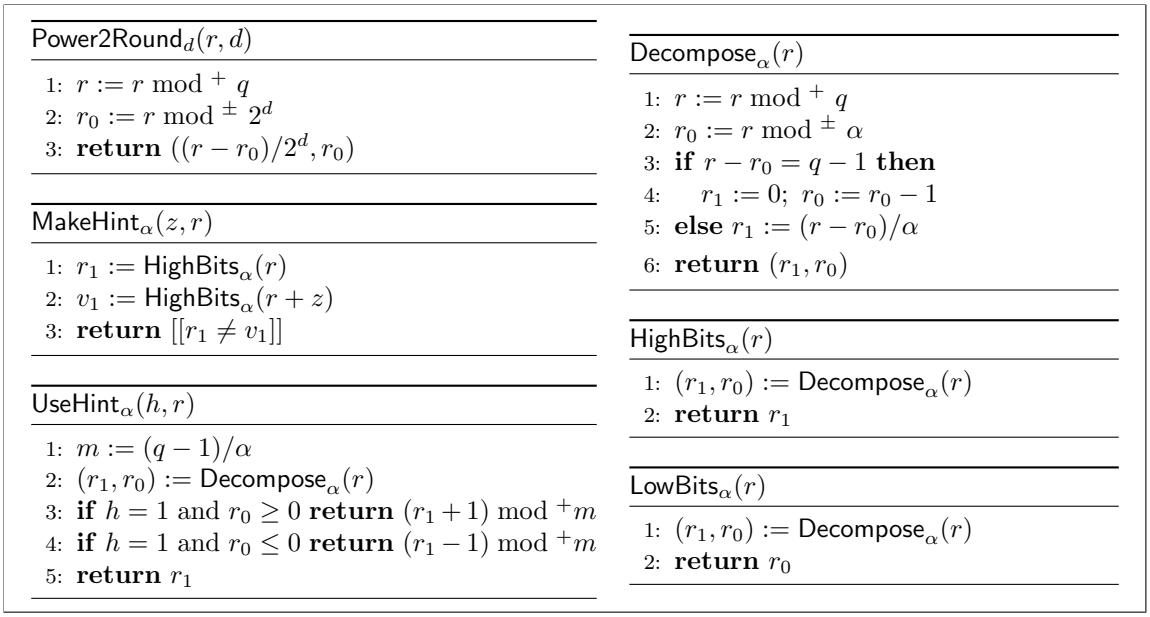

Figure 5: Supporting algorithms from Dilithium.

3 ML-DSA with Associated Data

The structure of Dilitium-G from §2.5 allows us to equip the signatures with additional features such as the associated data. In this section we detail the ML-DSA with associated data construction for a single signer.

3.1 Construction

The idea is that we randomise the nonce commitment twice. Then the opening phase we reveal that $\mathbf{w}$ was computed as a sum of $\mathbf{w}_{\mathsf{pr}}$ from the original construction and $\mathbf{w}_{\mathsf{tw}}$ that depends on the associated data. On the high level the entropy of $\mathbf{w}_{\mathsf{pr}}$ ensures the hiding property of the signature with associated data and the entropy of the random oracle in $\mathbf{w}_{\mathsf{tw}} = \mathbf{A} \cdot \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk}))$ ensures the binding property.

To verify that a signature corresponds to given associated data the verifier could recover the commitment value $\mathbf{w}$ from the given signature and compute $\mathbf{w}_{\mathsf{tw}} = \mathbf{A} \cdot \mathsf{Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}, \dots))$ from $\mathbf{w}_{\mathsf{pr}}$ presented in the opening phase. If $\mathbf{w} = \mathbf{w}_{\mathsf{pr}} + \mathbf{w}_{\mathsf{tw}}$ the verifier returns 1. However, by construction of the MLDSA standard the value $\mathbf{w}$ can only be reconstructed up to a short error. Hence, we modify the verification from an equality check to be a norm check. To minimise the norm of the error factor equal to $c\mathbf{t}_0 - c\mathbf{s}_2 - \mathbf{y}_2$ we include $\mathbf{t}_0$ as part of the proof opening. The value $\mathbf{t}_0$ can also be stored on the device and only transmitted once to optimise the size of the proof. We think removing the remaining error factor of $c\mathbf{s}_2 + \mathbf{y}_2$ is an interesting open problem.

We define the signing with associated data and the verification of associated data protocols in Figure 7. The remaining algorithms such as key generation and signature verification stay the same as in Figure 6.

Algorithm 1 $\mathsf{MLDSA.KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$

$(\rho, \rho') \leftarrow \{0,1\}^{256} \times \{0,1\}^{256}$
$\mathbf{A} \in \mathcal{R}_q^{k \times \ell} = \mathsf{ExpandA}(\rho)$
$(\mathbf{s}_1, \mathbf{s}_2) \leftarrow S_\eta^{k+\ell}$
$\mathbf{t} = \mathbf{A} \cdot \mathbf{s}_1 + \mathbf{s}_2 \bmod q$
$(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}_d(\mathbf{t})$
$\mathsf{pk} = (\rho, \mathbf{t}_1)$ ▷ $\mathsf{pk} = (\rho, \mathbf{t}_0, \mathbf{t}_1)$ for $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$
$\mathsf{tr} \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{pk})$
$\mathsf{sk} = (\rho, \mathsf{tr}, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$
return $(\mathsf{pk}, \mathsf{sk})$

Algorithm 2 $\mathsf{MLDSA.Sign}(\mathsf{sk} = (\rho, \mathsf{tr}, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0), \mathsf{m} \in \{0,1\}^*, \mathsf{ad}) \to (\mathsf{sig}, \pi)$

$\mu \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
$(\mathbf{z}, \mathbf{h}) = \perp$
while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
  $(\mathbf{y}_1, \mathbf{y}_2) \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y}$
  $\mathbf{w} = \mathbf{A} \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod q$
  $\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
  $\tilde{c} \in \{0,1\}^{2\lambda} = \mathsf{H_c}(\mu, \mathbf{w}_1)$
  $c \in \mathcal{R} = \mathsf{SampleInBall}_\tau(\tilde{c})$ ▷ $c = \sum c_i X^i \text{s.t.} c_i \in \{-1, 0, 1\}, \mathsf{HW}(c) \leq \tau$
  $\mathbf{z}_1 = \mathbf{y}_1 + c \cdot \mathbf{s}_1$
  $\mathbf{z}_2 = \mathbf{y}_2 + c \cdot \mathbf{s}_2$
  $p \leftarrow \mathcal{U}([0,1))$
  if $p \geq \min(\frac{1}{M} \cdot \exp\left(\frac{-2\pi \cdot \mathsf{coeff}((\mathbf{z}_1, \mathbf{z}_2))^\intercal \cdot \mathsf{coeff}(c \cdot (\mathbf{s}_1, \mathbf{s}_2)) + \pi \cdot \|c \cdot (\mathbf{s}_1, \mathbf{s}_2)\|^2}{\sigma_{\mathsf{pre}}^2}\right), 1)$ **then**
    $(\mathbf{z}, \mathbf{h}) = \perp$
  else
    $\mathbf{e} = \mathbf{z}_2 - c\mathbf{t}_0$
    $\mathbf{h} = \mathsf{MakeHint}_{2\gamma_2}(\mathbf{e}, \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d)$ ▷ $= \mathsf{MakeHint}(\mathbf{e}, \mathbf{w} - \mathbf{e})$
    if $\|\mathbf{e}\|_\infty \geq \gamma_2 \vee \|\mathbf{h}\|_1 \geq w \vee \|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ **then**
      $(\mathbf{z}, \mathbf{h}) = \perp$
return $\mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h})$

Algorithm 3 $\mathsf{MLDSA.Verify}(\mathsf{pk} = (\rho, \mathbf{t}_1), \mathsf{m}, \mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h})) \to \{0,1\}$

$\mathbf{A} = \mathsf{ExpandA}(\rho)$
$\mu = \mathsf{H}(\mathsf{H}(\mathsf{pk}), \mathsf{m})$
$c = \mathsf{SampleInBall}(\tilde{c})$
$\mathbf{w}'_{\mathsf{approx}} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d \bmod q$
$\mathbf{w}'_1 = \mathsf{UseHint}_{2\gamma_2}(\mathbf{h}, \mathbf{w}'_{\mathsf{approx}})$
$\tilde{c}' = \mathsf{H_c}(\mu, \mathbf{w}'_1)$
return $[[\|\mathbf{z}\|_\infty < \gamma_1 - \beta \wedge \tilde{c} = \tilde{c}']]$

Figure 6: ML-DSA with Gaussian Nonce algorithms.

Security. This simple embedding of associated data was more difficult to analyse in practice compared to the classical construction in [KLLS26]. Firstly, since $\mathbf{y}'$ is revealed in the proof and the signature has to stay unforgeable the security of the scheme reduces to a variant of ML-DSA with leakage. Second, since nonces are short this construction also allows the adversary to obtain $\mathbf{y} + \mathbf{y}'$ in the hiding experiment. This means we need to argue about the residual entropy in $\mathbf{A} \cdot \mathbf{y}$ given linear leakage on $\mathbf{y}$. Discrete Gaussian distribution of the nonce enables us to prove all given properties in §3.2.

3.2 Security

We first prove that the honestly computed proof and signature pass the associated data verification with overwhelming probability.

Lemma 10 (Soundness)**.** *Let* $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ *be the ADA signature scheme from Figures 6 and 7. Let* $\tau, \eta, \gamma_2, q, \beta, n > 0$ *be as in Table 1. Let* $\sigma_{\mathsf{ad}}, \sigma_{\mathsf{pre}} \geq 0$ *and* $\varepsilon = 2^{-64}$*. Then for* $(\mathsf{sig}, \pi) \leftarrow \mathsf{MLDSA.Sign\text{-}AD}(\mathsf{sk}, \mathsf{m} \in \{0,1\}^*, \mathsf{ad})$

$$\Pr(\mathsf{MLDSA.Verify\text{-}AD}(\mathsf{pk}, \mathsf{m}, \mathsf{ad}, \mathsf{sig}, \pi = (\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)) = 1) \geq 1 - 2^{-\lambda} \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right).$$

Proof. In the honest execution of Algorithm 6, we have $\mathbf{w}_{\mathsf{approx}} = \mathbf{w} - c\mathbf{s}_2 - \mathbf{y}_2$, where $\mathrm{RD}_\infty(\mathbf{y}_2 + c\mathbf{s}_2, \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}}) \leq 1 + \frac{\varepsilon}{1-\varepsilon}$ using Lemma 7. Then, by Corollary 1, the norm is bounded as $\|c\mathbf{s}_2 + \mathbf{y}_2\| \leq 0.53 \cdot \sigma_{\mathsf{pre}} \cdot \sqrt{kn}$ with probability at least $1 - 2^{-\lambda} \cdot (1 + \frac{\varepsilon}{1-\varepsilon})$. According to Figure 7 when the norm bound is satisfied the verification is guaranteed to succeed. □

The standard MLDSA works in a fully splitting ring $\mathcal{R}_q \sim \prod_{i=1}^{n} \mathbb{Z}_q/(X - r_i)$. This means that $\mathcal{R}_q$ has many ideals of size as small as q. Then for a fixed matrix $\mathbf{A}$ the coordinates of a matrix vector product $\mathbf{A} \cdot \mathbf{y}$ may belong to ideals and it's entropy would be bounded by the size of the ideal. We derive a tighter bound on the entropy of $\mathbf{A} \cdot \mathbf{y}$ taking into account the randomness of $\mathbf{A}$ and the ideals it generates.

Lemma 11 (Entropy of $\mathbf{A} \cdot \mathbf{y}$)**.** *Let* $q, k, \ell, n > 0$*,* $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$*. Let* $\sigma > 0$ *such that* $q^k/\sigma^\ell \geq n$ *and* $\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4\ell n)}}$*. Then for any set* $S \subseteq \mathcal{R}_q^k$*,* $\mathbf{c} \in \mathcal{K}_{\mathbb{R}}^\ell$*,* $\mathbf{A} \leftarrow \mathcal{R}_q^{k \times \ell}$ *and* $\mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma, \mathbf{c}}$ *we have*

$$\Pr(\mathbf{A} \cdot \mathbf{y} \bmod q \in S) \leq \sqrt{\frac{|S|}{\sigma^{\ell n}} \cdot 2\exp(1)}.$$

Proof. By Lemma 2, $\mathrm{RD}((\mathbf{A}, \mathbf{A} \cdot \mathcal{D}_{\mathcal{R}^\ell, \sigma}) \| (\mathbf{A}, \mathcal{U}(\mathcal{R}_q^k))) \leq 2 \cdot \left(\frac{q^k}{\sigma^\ell} + 1\right)^n$. Then for any set $S \subseteq \mathcal{R}_q^k$ by

Algorithm 4 $\mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y}) \to \mathcal{R}^{\ell}$

$\mathbf{return}\ \mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^{\ell},\sigma_{\mathsf{ad}}}(\mathsf{coins} = (\tilde{y}))$

Algorithm 5 $\mathsf{MLDSA.Sign\text{-}AD}(\mathsf{sk} = (\rho, \mathsf{tr}, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0), \mathsf{m} \in \{0,1\}^*, \mathsf{ad}) \to (\mathsf{sig}, \pi)$

$\mu \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
$(\mathbf{z}, \mathbf{h}) = \perp$
$\mathbf{while}\ (\mathbf{z}, \mathbf{h}) = \perp\ \mathbf{do}$
  $(\mathbf{y}_1, \mathbf{y}_2) \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_{\mathsf{pre}}}$
  $\mathbf{w}_{\mathsf{pr}} = \mathbf{A} \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod q$
  $\tilde{y} \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})$
  $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y})$
  $\mathbf{w}_{\mathsf{tw}} = \mathbf{A} \cdot \mathbf{y}' \bmod q$
  $\mathbf{w} = \mathbf{w}_{\mathsf{pr}} + \mathbf{w}_{\mathsf{tw}}$
  $\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
  $\tilde{c} \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{c}}(\mu, \mathbf{w}_1)$
  $c \in \mathcal{R} = \mathsf{SampleInBall}_{\tau}(\tilde{c})$
  $\mathbf{z}_1 = \mathbf{y}_1 + c \cdot \mathbf{s}_1$
  $\mathbf{z}_2 = \mathbf{y}_2 + c \cdot \mathbf{s}_2$
  $p \leftarrow \mathcal{U}([0,1))$
  $\mathbf{if}\ p \geq \min(\frac{1}{M} \cdot \exp\left(\frac{-2\pi \cdot \mathsf{coeff}((\mathbf{z}_1,\mathbf{z}_2))^{\mathsf{T}} \cdot \mathsf{coeff}(c \cdot (\mathbf{s}_1,\mathbf{s}_2)) + \pi \cdot \|c \cdot (\mathbf{s}_1,\mathbf{s}_2)\|^2}{\sigma_{\mathsf{pre}}^2}\right), 1)\ \mathbf{then}$
    $(\mathbf{z}, \mathbf{h}) = \perp$
  $\mathbf{else}$
    $\mathbf{e} = \mathbf{z}_2 - c\mathbf{t}_0$
    $\mathbf{z} = \mathbf{z}_1 + \mathbf{y}'$
    $\mathbf{h} = \mathsf{MakeHint}_{2\gamma_2}(\mathbf{e}, \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d)$ ▷ $= \mathsf{MakeHint}(\mathbf{e}, \mathbf{w} - \mathbf{e})$
    $\mathbf{if}\ \|\mathbf{e}\|_\infty \geq \gamma_2 \vee \|\mathbf{h}\|_1 \geq w \vee \|\mathbf{z}\|_\infty \geq \gamma_1 - \beta\ \mathbf{then}$
      $(\mathbf{z}, \mathbf{h}) = \perp$
$\mathsf{sig} = (\tilde{c}, \mathbf{z} \bmod q, \mathbf{h})$
$\pi = (\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0)$
$\mathbf{return}\ (\mathsf{sig}, \pi)$

Algorithm 6 $\mathsf{MLDSA.Verify\text{-}AD}(\mathsf{pk} = (\rho, \mathbf{t}_1), \mathsf{m}, \mathsf{ad}, \mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h}), \pi = (\mathbf{w}', \mathbf{t}_0)) \to \{0,1\}$

$\mathbf{A} = \mathsf{ExpandA}(\rho)$
$\mathbf{t} = \mathbf{t}_1 \cdot 2^d + \mathbf{t}_0$
$\mu = \mathsf{H}(\mathsf{H}(\mathsf{pk}), \mathsf{m})$
$c = \mathsf{SampleInBall}(\tilde{c})$
$\mathbf{w}_{\mathsf{approx}} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t} \bmod q$ ▷ $= \mathbf{w} - c\mathbf{s}_2 - \mathbf{y}_2$ computed from sig and π
$\tilde{y} = \mathsf{H}_{\mathsf{ad}}(\mathbf{w}', \mathbf{t}_0, \mathsf{ad}, \mathsf{m}, \mathsf{pk})$
$\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y})$
$\mathbf{w}'' = \mathbf{A} \cdot \mathbf{y}' \bmod q$
$\mathbf{w} = \mathbf{w}' + \mathbf{w}''$ ▷ $= \mathbf{w}$ computed from π and ad.
$\mathbf{return}\ [[\|\mathbf{w}_{\mathsf{approx}} - \mathbf{w}\| \leq 0.53 \cdot \sigma_{\mathsf{pre}} \cdot \sqrt{kn}]]$

Figure 7: Associated data signing and verification.

the probability preservation property for Rényi Divergence we have

$$\begin{aligned}\Pr(\mathbf{A}\cdot\mathbf{y} \bmod q \in S \mid \mathbf{A}\leftarrow\mathcal{R}_q^{k\times\ell}, \mathbf{y}\leftarrow\mathcal{D}_{\mathcal{R}^\ell,\sigma}) &\le \sqrt{\Pr(\mathbf{u}\in S \mid \mathbf{u}\leftarrow\mathcal{R}_q^k)\cdot 2\cdot\left(\frac{q^k}{\sigma^\ell}+1\right)^n}\\ &= \sqrt{\frac{|S|}{q^{kn}}\cdot 2\cdot\left(\frac{q^k}{\sigma^\ell}+1\right)^n}\\ &\le \sqrt{\frac{|S|}{q^{kn}}\cdot 2\cdot\frac{q^{kn}}{\sigma^{\ell n}}\cdot\left(1+\frac{1}{n}\right)^n}\\ &\le \sqrt{\frac{|S|}{\sigma^{\ell n}}\cdot 2\exp(1)}\end{aligned}$$

where the last two inequalities come from $1 \le \frac{q^k}{n\cdot\sigma^\ell}$ and $(1+1/n)^n \le \exp(1)$. □

Then by Corollary 2 the same approach applies given noisy linear leakage on $\mathbf{y}$.

Corollary 3. *Let $q, k, \ell, n, \sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \ge 0$ and $\sigma_2 \coloneqq \left(1/\sigma_{\mathsf{ad}}^2 + 1/\sigma_{\mathsf{ad}}^2\right)^{-1/2}$ such that $\sigma_2 \le \frac{\sqrt{\pi}}{2}\cdot\frac{q}{\sqrt{\ln(4\ell n)}}$, $q^k/\sigma_2^\ell \ge n$. Let $\mathbf{A}\leftarrow\mathcal{R}_q^{k\times\ell}, \mathbf{y}'\leftarrow\mathcal{D}_{\mathcal{R}^\ell,\sigma_{\mathsf{pre}}}, \mathbf{y}'\leftarrow\mathcal{D}_{\mathcal{R}^\ell,\sigma_{\mathsf{ad}}}$ and $S\subseteq\mathcal{R}_q^k$. Then by Corollary 2 we also get*

$$\Pr(\mathbf{A}\cdot\mathbf{y}' \bmod q \in S \mid \mathbf{y} = \mathbf{y}'+\mathbf{y}') \le \sqrt{\frac{|S|}{\sigma_2^{\ell n}}\cdot 2\exp(1)}.$$

As part of the proof we need an upper bound on the number of integer points of bounded norm. We include the proof of the following proposition for completeness.

Proposition 1. *Let $R > 0$ and $d \ge 1$ be an integer. Then the cardinality of $\mathbb{Z}^d \cap \mathcal{B}_d(R)$ is bounded as*

$$\left|\mathbb{Z}^d \cap \mathcal{B}_d(R)\right| \le \left(\frac{\sqrt{2\pi e}\cdot(R+\sqrt{d})}{\sqrt{d+1}}\right)^d$$

Proof. We use a volume argument where each integer point $\mathbf{x}$ is associated with a hypercube $\mathcal{P}(\mathbf{x}) \coloneqq \mathbf{x} + [0,1)^d$ of volume 1. For all $\mathbf{x}\in\mathcal{B}_d(R)$ this tiling is contained within a larger ball of radius $R+\sqrt{d}$. Then

$$\left|\mathbb{Z}^d \cap \mathcal{B}_d(R)\right| \le \frac{\mathsf{Vol}(\mathcal{B}_d(R+\sqrt{d}))}{\mathsf{Vol}(\mathcal{P})} = \mathsf{Vol}(\mathcal{B}_d(R+\sqrt{d})).$$

using $\mathsf{Vol}(\mathcal{B}_d(R)) = \frac{\pi^{d/2}R^d}{\Gamma(1+d/2)}$ and by [Bat08, Thm. 1.5] $\Gamma(1+d/2) \ge \sqrt{2e}\cdot\left(\frac{d+1}{2e}\right)^{(d+1)/2}$. Putting the arguments together we get $\mathsf{Vol}(\mathcal{B}_d(R+\sqrt{d})) \le \frac{1}{\sqrt{d+1}}\left(\frac{\sqrt{2\pi e}\cdot(R+\sqrt{d})}{\sqrt{d+1}}\right)^d$ and the statement follows. □

Lemma 12 (Weak AD-Binding)**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1. Let $\sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \ge 0$ and $R = 0.53\cdot\sigma_{\mathsf{pre}}\cdot\sqrt{kn}$. Then for any PPT adversary $\mathcal{A}$ making at most $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries, the ADA signature scheme $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ defined in Figures 6 and 7 satisfies*

$$\mathsf{Adv}^{\mathsf{Weak\text{-}AD\text{-}Binding}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{A}) \le {\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}^2\cdot\sqrt{\frac{R_1^{kn}}{\sigma_{\mathsf{ad}}^{\ell n}}\cdot 2\exp(1)} + \mathsf{negl}(\lambda).$$

where $R_1 = \sqrt{2\pi e} \cdot (1 + 2 \cdot 0.53 \cdot \sigma_{\mathsf{pre}})$.

Proof. Following the definition of Weak-AD-Binding experiment, in this proof we assume an honestly sampled public key $\mathbf{A} \leftarrow \mathcal{R}_q^{k \times \ell}$. The winning condition of the adversary is equivalent to returning two tuples $(\mathsf{ad}^{(0)}, \mathbf{w}^{(0)}, \mathbf{t}_0^{(0)})$, $(\mathsf{ad}^{(1)}, \mathbf{w}^{(1)}, \mathbf{t}_0^{(1)})$ and a signature $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ such that for $\mathbf{w} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d \bmod q$ and $c = \mathsf{SampleInBall}(\tilde{c})$, it holds

$$\left\| \mathbf{w} - c \cdot \mathbf{t}_0^{(0)} - \mathbf{w}^{(0)} - \mathbf{A} \cdot \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(0)}, \mathbf{t}_0^{(0)}, \mathsf{ad}^{(0)}) \right\| \leq R,$$
$$\left\| \mathbf{w} - c \cdot \mathbf{t}_0^{(1)} - \mathbf{w}^{(1)} - \mathbf{A} \cdot \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(1)}, \mathbf{t}_0^{(1)}, \mathsf{ad}^{(1)}) \right\| \leq R.$$

Then by Proposition 1, the number of integer vectors satisfying the constraint above is bounded by $\left(\frac{\sqrt{2\pi e} \cdot (R + \sqrt{kn})}{\sqrt{kn+1}}\right)^{kn} \leq (\sqrt{2\pi e} \cdot (1 + c \cdot \sigma_{\mathsf{pre}}))^{kn} =: R_0^{kn}$. If either of the two tuples has not been queried to the H_{ad} oracle before, the probability that the above holds is bounded by $2 \cdot \sqrt{\frac{R_0^{kn}}{\sigma_{\mathsf{ad}}^{\ell n}} \cdot 2\exp(1)}$ by Lemma 11. In the remainder of the proof, we assume that both tuples have been queried to H_{ad}. Then by triangle inequality, we have

$$\left\| c \cdot (\mathbf{t}_0^{(1)} - \mathbf{t}_0^{(0)}) + (\mathbf{w}^{(1)} - \mathbf{w}^{(0)}) + \mathbf{A} \cdot (\mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(1)}, \mathbf{t}_0^{(1)}, \mathsf{ad}^{(1)}) - \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(0)}, \mathbf{t}_0^{(0)}, \mathsf{ad}^{(0)})) \right\| \leq 2R.$$

For any fixed $\mathbf{v} \coloneqq c \cdot (\mathbf{t}_0^{(1)} - \mathbf{t}_0^{(0)}) + (\mathbf{w}^{(1)} - \mathbf{w}^{(0)}) \in \mathcal{R}_q^k$, random oracle returns $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(0)}, \mathbf{t}_0^{(0)}, \mathsf{ad}^{(0)}) = \mathbf{y}_0 \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}$ and $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(1)}, \mathbf{t}_0^{(1)}, \mathsf{ad}^{(1)}) = \mathbf{y}_1 \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}$. Then for $p = \Pr(\mathbf{A} \cdot \mathbf{y}_0 - \mathbf{A} \cdot \mathbf{y}_1 \in \{\mathbf{v} + \mathbf{u} \mid \mathbf{u} \in \mathcal{B}_{kn}(2R)\})$ we have

$$\begin{aligned} p &= \sum_{\mathbf{y}_1 \in \mathcal{R}^\ell} \Pr(\mathbf{y}_1 \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}) \cdot \Pr(\mathbf{A} \cdot \mathbf{y}_0 \in \{\mathbf{v} + \mathbf{u} + \mathbf{A} \cdot \mathbf{y}_1 \mid \mathbf{u} \in \mathcal{B}_{kn}(2R)\}) \\ &\leq \sum_{\mathbf{y}_1 \in \mathcal{R}^\ell} \Pr(\mathbf{y}_1 \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}) \cdot \sqrt{\frac{R_1^{kn}}{\sigma_{\mathsf{ad}}^{\ell n}} \cdot 2\exp(1)} \\ &\leq \sqrt{\frac{R_1^{kn}}{\sigma_{\mathsf{ad}}^{\ell n}} \cdot 2\exp(1)}. \end{aligned}$$

where using similar calculations we get $R_1 = \sqrt{2\pi e} \cdot (1 + 2\,c \cdot \sigma_{\mathsf{pre}})$. Taking into account that any two of $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries may result in a collision, we get $\binom{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{2} \cdot \sqrt{\frac{R_1^{kn}}{\sigma_{\mathsf{ad}}^{\ell n}} \cdot 2\exp(1)}$ simplifying the expressions derived we get the statement. □

Remark 2. *The construction we consider does not satisfy Strong AD-Binding. This is because we cannot assume the public key* $\mathbf{A}$ *follows a uniform distribution. For example, when* $\mathbf{A} = 0$ *the value* $\mathbf{A} \cdot \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(b)}, \mathbf{t}_0^{(b)}, \mathsf{ad}^{(b)})$ *no longer affects the signature, or when a row of* $\mathbf{A}$ *only contains elements in a small ideal of* $\mathcal{R}_q$ *the product* $\mathbf{A} \cdot \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(b)}, \mathbf{t}_0^{(b)}, \mathsf{ad}^{(b)})$ *has reduced entropy.*

We prove associated data hiding property using a different entropy argument relying on both the entropy of $\mathbf{A} \cdot \mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk})$ and $\mathbf{A} \cdot \mathbf{y}$ with leakage.

Lemma 13 (AD-Hiding)**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1 and $\sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \geq 0$. Then for any PPT adversary $\mathcal{A}$ making atmost $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries, the ADA signature scheme $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ defined in Figures 6 and 7 satisfies:*

$$\mathsf{Adv}^{\mathsf{AD\text{-}Hiding}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{A}) \leq \sqrt{\frac{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{\sigma_2^{\ell n}} \cdot 2\exp(1)} + \sqrt{\frac{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{\sigma_{\mathsf{pre}}^{\ell n}} \cdot 2\exp(1)}$$

where $\sigma_2 = \left(1/\sigma_{\mathsf{pre}}^2 + 1/\sigma_{\mathsf{ad}}^2\right)^{-1/2}$.

Proof. We proceed in a series of equivalent experiments. In Hyb_0 adversary's view is distributed as

$$\mathcal{D}_0 = \{\sigma = (\tilde{c}, \mathbf{z} = \mathbf{y} + \mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk}) + c\mathbf{s}_1, \mathbf{h}), \mathsf{pk}, \mathsf{sk}, \mathsf{m}, \mathsf{ad}_0, \mathsf{ad}_1\}.$$

In addition, we denote $L_{\mathsf{H}_{\mathsf{ad}}}$ the list of queries to the random oracle H_{ad}.

Hyb_1**:** Here we define $\mathcal{D}_1$ that only contains values dependent on $\mathsf{ad}_0, \mathsf{ad}_1$, the remaining view of $\mathcal{A}$ is computed as a function of $\mathcal{D}_1$ and the secret key the adversary has access too

$$\mathcal{D}_1 = \left\{\mathbf{y} + \mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk}), \mathsf{pk}, \mathsf{m}, \mathsf{ad}_0, \mathsf{ad}_1 \mid \mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{pre}}}, \mathbf{w}_{\mathsf{pr}} = \mathbf{A} \cdot \mathbf{y} \bmod q, b \leftarrow \{0,1\}\right\}.$$

Hyb_2**:** Here we sample $\mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{pre}}}$ and abort the experiment if $\mathbf{w}_{\mathsf{pr}} = \mathbf{A} \cdot \mathbf{y} \bmod q$ was previously queried to the random oracle. By Lemma 11, $\mathbf{A} \cdot \mathbf{y} \bmod q$ is equal to a previous oracle query with probability bounded by $\left(\frac{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{\sigma_{\mathsf{pre}}^{\ell n}} \cdot 2\exp(1)\right)^{1/2}$. This change ensures that $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk})$ is sampled as $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk}) =: \mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}$ and not already present in $L_{\mathsf{H}_{\mathsf{ad}}}$ maintained by the random oracle.

Hyb_3**:** Here we add an abort condition if $\mathbf{w}_{\mathsf{pr}}$ is queried to the random oracle after sampling $\mathbf{y}'$. Before such query is made, by Lemma 3, the distribution $\mathcal{D}_2$ is equal to

$$\mathcal{D}_3 = \left\{\hat{\mathbf{y}} + \mathsf{H}_{\mathsf{ad}}(\mathbf{A} \cdot \mathbf{y}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk}), \mathsf{pk}, \mathsf{m}, \mathsf{ad}_0, \mathsf{ad}_1 \mid \hat{\mathbf{y}} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{pre}}}, \mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}, \mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_2, \mathbf{c}}\right\}.$$

where $\sigma_2 = \left(1/\sigma_{\mathsf{pre}}^2 + 1/\sigma_{\mathsf{ad}}^2\right)^{-1/2}$, and $\mathbf{c} = \frac{\sigma_2}{\sigma_{\mathsf{ad}}} \cdot (\hat{\mathbf{y}} + \mathsf{H}_{\mathsf{ad}}(\mathbf{A} \cdot \mathbf{y}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk}))$. Hence, the query probability is the same for $\mathcal{D}_2$ and $\mathcal{D}_3$. By Lemma 11 it is bounded by $\sqrt{\frac{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{\sigma_2^{\ell n}} \cdot 2\exp(1)}$ for the independent random variable $\mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_2, \mathbf{c}}$. Consequently, $\mathrm{SD}(\mathsf{Hyb}_2, \mathsf{Hyb}_3) \leq \sqrt{\frac{\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}}{\sigma_2^{\ell n}} \cdot 2\exp(1)}$.

Hyb_4**:** Finally, in $\mathcal{D}_4$ the Challenger samples $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}$ instead of $\mathbf{y}' = \mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk})$. As follows from Hyb_2 and Hyb_3, the experiment aborts if H_{ad} is called on input $(\mathbf{w}_{\mathsf{pr}}, \mathsf{ad}_b, \mathsf{m}, \mathsf{pk})$. Hence, the change in $\mathbf{y}'$ does not impact the view of the Adversary. Note that the distribution $\mathcal{D}_4$ is now independent of the bit b, implying the Adversary's advantage in Hyb_4 is equal to 0. □

It remains to prove the properties of the underlying signature. We first prove correctness of the signing algorithm.

Lemma 14 (Signature Correctness)**.** *For any message* m*, associated data* ad *and valid* $(\mathsf{sk}, \mathsf{pk})$ *for the signature scheme* $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ *defined in Figures 6 and 7. Let* $t \geq 1$*,* $M = e^{9.5/t+1/2t^2}$*,* $\varepsilon = 2^{-64}$*,* $\varepsilon' > 0$*, and* $0.53 \cdot (\sigma_{\mathsf{pre}} + \sigma_{\mathsf{ad}}) \cdot \sqrt{\ell n} \leq \gamma_1 - \beta$*,* $\tau \cdot 2^{d-1} + 0.53 \cdot \sigma_{\mathsf{pre}} \sqrt{kn} \leq \gamma_2$*. Then every signing attempt returns a valid signature* sig *with probability*

$$\Pr(\mathsf{sig} \neq \perp) \geq 1 - \frac{M - 1 + \varepsilon}{M} - 2 \cdot 2^{-\lambda} \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right) - 2^{-\lambda} \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right),$$

and for every $\perp \neq \mathsf{sig} \leftarrow \mathsf{MLDSA.Sign\text{-}AD}(\mathsf{sk}, \mathsf{m}, \mathsf{ad})$ *we have* $\mathsf{MLDSA.Verify}(\mathsf{pk}, \mathsf{m}, \mathsf{sig}) = 1$ *with probability* 1*.*

Proof. In the notations of Figure 7, note that when $\mathsf{sig} \neq \perp$ we have $\|\mathbf{e}\|_\infty < \gamma_2 \wedge \|\mathbf{z}\|_\infty < \gamma_1 - \beta$. Then by Lemma 9 and $\|\mathbf{e}\|_\infty < \gamma_2$, we have $\mathsf{UseHint}_q\big(\mathsf{MakeHint}_q(\mathbf{e}, \mathbf{w} - \mathbf{e}, 2\gamma_2), \mathbf{w} - \mathbf{e}, 2\gamma_2\big) = \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$. Where for an honestly computed signature $\mathbf{A} \cdot \mathbf{z} - c\mathbf{t} \cdot 2^d = \mathbf{w} - \mathbf{e}$. Hence, $\tilde{c} = \mathsf{H_c}(\mu, \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)) = \mathsf{H_c}(\mu, \mathsf{UseHint}_q\big(\mathbf{h}, \mathbf{A} \cdot \mathbf{z} - c\mathbf{t} \cdot 2^d, 2\gamma_2\big)) = \tilde{c}'$ with probability 1. Together with $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$ this implies that the verification algorithm returns 1 with probability 1. It remains to compute the likelihood of producing a valid signature.

By Lemma 7, the rejection sampling accepts a given sample with probability $\geq 1 - \frac{M-1+\varepsilon}{M}$. If a given sample is accepted by the rejection sampling, then we have $\mathrm{RD}_\infty(\mathbf{e}, \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}} - c \cdot \mathbf{t}_0) \leq 1 + \varepsilon/(1-\varepsilon)$, and $\mathrm{RD}_\infty(\mathbf{z}, \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{pre}}}) \leq 1 + \varepsilon/(1-\varepsilon)$. Then by Corollary 1 we have $\|\mathbf{e}\|_\infty \leq \tau \cdot 2^{d-1} + 0.53 \cdot \sigma_{\mathsf{pre}} \sqrt{kn} \leq \gamma_2$ with probability at least $1 - 2^{-\lambda} \cdot (1 + \frac{\varepsilon}{1-\varepsilon})$. Lastly, for $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ the signature component $\mathbf{z}$ is defined as $\mathbf{z} = \mathbf{z}' + \mathbf{y}'$ where, $\mathbf{z}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{pre}}}$ and $\mathbf{y}'$ is sampled independently from a discrete Gaussian distribution $\mathcal{D}_{\mathcal{R}^\ell, \sigma_{\mathsf{ad}}}$. Therefore, by Corollary 1, $\|\mathbf{y}'\| \leq 0.53 \cdot \sigma_{\mathsf{ad}} \sqrt{\ell n}$ and $\|\mathbf{z}'\| \leq 0.53 \cdot \sigma_{\mathsf{pre}} \sqrt{\ell n}$ with probability at least $1 - 2 \cdot 2^{-\lambda} \cdot (1 + \frac{\varepsilon}{1-\varepsilon})$. □

Lastly, we prove the unforgeability of $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ signature itself. The modification in the commitment implies that we cannot rely on unforgeability of MLDSA from §2.5. Instead, we reduce the security to the unforgeability with MLDSA with a specific nonce leakage as defined in Figure 8. This result may be of independent interest. For completeness we include the full unforgeability proof with nonce leakage and Rényi Divergence based transcript simulation.

Lemma 15 (adapted [KLS18, Thm. 3.2])**.** *Let* Π *be a signature scheme based on Fiat-Shamir with aborts transformation of an underlying identification protocol* ID*. Assume that* ID *is* $\varepsilon_{\mathsf{zk}}$*-naHVZK (defined in Definition 5) and has at least* α *bit of commitment entropy. Then*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi}(\mathcal{A}) \leq \mathsf{Q_{Sign}} \cdot 2^{-\alpha} + (\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\Pi}(\mathcal{B}) + \mathsf{Q_{H_c}} \cdot \mathsf{Q_{Sign}} \cdot 2^{-\alpha}) \cdot e^{\varepsilon_{\mathsf{zk}} \cdot \mathsf{Q_{Sign}}}$$

where $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(\mathcal{A}) + (\mathsf{Q_{H_c}} + \mathsf{Q_{Sign}})$*.*

Proof. We provide a proof sketch for the lemma. Hyb_0. is the standard EUF-CMA experiment.

Hyb_1. Here, we replace $c = \mathsf{H_c}(\mathsf{HighBits}(\mathbf{w})||\mathsf{m})$ with $c \leftarrow \mathcal{U}(\mathcal{C})$ and program $\mathsf{H_c}(\mathsf{HighBits}(\mathbf{w})||\mathsf{m}) \coloneqq c$. Since the random oracle is programmed at points $(\mathbf{w}, \mathsf{m})$, and the commitment has α bits of entropy (i.e., $\Pr[\mathsf{HighBits}(\mathbf{w}) = \mathsf{HighBits}(\mathbf{w}^*)] \leq 2^{-\alpha}$), the probability that the adversary previously queried a programmed point is at most $2^{-\alpha}$. This incurs an additive loss of at most $\mathsf{Q_{Sign}} \cdot 2^{-\alpha}$ over all signing queries $\mathsf{Q_{Sign}}$.

Hyb_2. Now, we replace the real signing oracle with an $\varepsilon_{\mathsf{zk}}$-naHVZK simulator $\mathsf{Sim}(\mathsf{pk})$ and program the random oracle $\mathsf{H_c}$ accordingly. We also initialise a list of requested signatures as $\mathcal{L}_{\mathsf{Sign}}$. For every signing query on a message m, the simulator produces $(\mathbf{w}, c, \mathsf{sig})$ as

– $(\mathbf{w}, c, \mathsf{sig}) \leftarrow \mathsf{Sim}(\mathsf{pk})$

– Program $\mathsf{H_c}(\mathsf{HighBits}(\mathbf{w})||\mathsf{m}) := c$ and add $(\mathsf{m}, \mathbf{w}, c, \mathsf{sig})$ to $\mathcal{L}_{\mathsf{Sign}}$

with $\mathrm{RD}_\infty(\mathsf{Sim}(\mathsf{pk})\|\mathsf{AcceptedTrans}(\mathsf{sk})) \leq 1 + \varepsilon_{\mathsf{zk}}$ and $c \sim \mathcal{U}(\mathcal{C})$ as before. This removes the dependence on the secret key sk. Now, by data processing and probability preservation property, we get $\mathsf{Adv}_{\mathsf{Hyb}_1}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{Hyb}_2}(\mathcal{A}) \cdot \mathrm{RD}(\mathsf{Hyb}_1\|\mathsf{Hyb}_2) \leq \mathsf{Adv}_{\mathsf{Hyb}_2}(\mathcal{A}) \cdot \mathrm{RD}(\mathsf{AcceptedTrans}(\mathsf{sk})\|\mathsf{Sim}(\mathsf{pk}))^{\mathsf{Q}_{\mathsf{Sign}}} \leq \mathsf{Adv}_{\mathsf{Hyb}_2}(\mathcal{A}) \cdot e^{\varepsilon_{\mathsf{zk}} \cdot \mathsf{Q}_{\mathsf{Sign}}}$.

Hyb_3. (*Simulating the random oracle*) Then we change the computation of $\mathsf{H_c}(\mathsf{HighBits}(\mathbf{w})||\mathsf{m})$. If $\mathsf{m}, \mathbf{w} \in \mathcal{L}_{\mathsf{Sign}}$, fetch respective c from $\mathcal{L}_{\mathsf{Sign}}$. Else return $\mathsf{H_c}'(\mathsf{HighBits}(\mathbf{w})||\mathsf{m})$ where $\mathsf{H_c}'$ if a fresh random oracle. The change can only be detected if one of the $\mathsf{m}, \mathbf{w} \in \mathcal{L}_{\mathsf{Sign}}$ was previously queried to the random oracle which happens with probability at most $\mathsf{Q}_{\mathsf{H_c}} \cdot \mathsf{Q}_{\mathsf{Sign}} \cdot 2^{-\alpha}$. Hence, $\mathsf{Adv}_{\mathsf{Hyb}_3}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{Hyb}_4}(\mathcal{A}) + \mathsf{Q}_{\mathsf{H_c}} \cdot \mathsf{Q}_{\mathsf{Sign}} \cdot 2^{-\alpha}$.

Reduction to UF-NMA. Finally, we now construct an adversary $\mathcal{B}$ against the UF-NMA game. Adversary $\mathcal{B}$ internally runs $\mathcal{A}$ and simulates the signing oracle as above. Also, $\mathcal{B}$ perfectly simulates the random oracle $\mathsf{H_c}$ using the list $\mathcal{L}_{\mathsf{Sign}}$ of signature queries and the random oracle of the UF-NMA experiment as $\mathsf{H_c}'$.

When $\mathcal{A}$ outputs a forgery $(\mathsf{m}^*, (\mathbf{w}^*, c^*, \mathsf{sig}^*))$, the adversary $\mathcal{B}$ returns the same tuple as its forgery in the UF-NMA game. If $\mathcal{A}$'s forgery is successful then $\mathsf{m}^* \notin \mathcal{L}_{\mathsf{Sign}}$ and consequently $\mathsf{H_c}(\mathsf{HighBits}(\mathbf{w}^*), \mathsf{m}^*) = \mathsf{H_c}'(\mathsf{HighBits}(\mathbf{w}^*), \mathsf{m}^*)$. Hence, $(\mathsf{m}^*, (\mathbf{w}^*, c^*, \mathsf{sig}^*))$ is a successful forgery for $\mathcal{B}$ in the UF-NMA experiment and $\mathsf{Adv}_{\mathsf{Hyb}_4}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\Pi}(\mathcal{B})$. Therefore, the running time of $\mathcal{B}$ can be derived by adding the running time of $\mathcal{A}$ and the overhead of simulating the signing oracle and programming the random oracle, which is at most $(\mathsf{Q}_{\mathsf{H_c}} + \mathsf{Q}_{\mathsf{Sign}})$. □

We then establish how to simulate the signing oracle in the following lemma.

Lemma 16 ([DFPS22, Sec 4.])**.** *Let $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ be signature scheme defined in Figure 8. Let $t \geq 1$, $M = e^{9.5/t+1/2t^2}$, $\varepsilon = 2^{-64}$, Let $B \leq \tau\eta N\sqrt{n \cdot (k+\ell)}$ such that $\|c \cdot \mathbf{s}_i\| \leq B$ with probability $1 - \mathsf{negl}(\lambda)$, $\sigma_{\mathsf{pre}} > t \cdot \sqrt{2\pi} \cdot B$ and $\sigma_{\mathsf{ad}} > 0$. Then the underlying identification protocol is Rényi $\varepsilon_{\mathsf{zk}}$-naHVZK with*

$$\varepsilon_{\mathsf{zk}} \leq \frac{\varepsilon}{1-\varepsilon} + \mathsf{negl}(\lambda).$$

Proof. For each signing attempt on a message m, the simulator computes $(\mathbf{w}, c, (\mathbf{z}_1, \mathbf{z}_2), \mathbf{y}')$ as:

1: $b \leftarrow 0$
2: **while** $b = 0$ **do**
3: Sample $(\mathbf{z}_1', \mathbf{z}_2) \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_{\mathsf{pre}}}$, $c \leftarrow \mathcal{C}$, $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^{\ell}, \sigma_{\mathsf{ad}}}$
4: Compute $\mathbf{z}_1 = \mathbf{z}_1' + \mathbf{y}'$
5: Compute $\mathbf{w} = \mathbf{A} \cdot \mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t}$
6: $b \leftarrow \mathsf{Ber}\left(\frac{1}{M}\right)$
7: **return** $(c, \mathbf{w}, (\mathbf{z}_1, \mathbf{z}_2), \mathbf{y}')$

Observe that the simulator Sim defined above is precisely the $\mathsf{B}^{\mathsf{ideal}}$ algorithm from Figure 2, whereas the $\mathsf{AcceptedTrans}(\mathsf{sk})$ corresponds to the algorithm $\mathsf{B}^{\mathsf{real}}$. Since $\sigma_{\mathsf{pre}} > t \cdot \sqrt{2\pi} \cdot B$, by Lemma 8, we have $\mathrm{RD}^{2^{-64}}_{\infty}(\mathcal{D}_{\mathcal{R}^{(\ell+k)}, \sigma_{\mathsf{pre}}, \mathbf{0}}, \mathcal{D}_{\mathcal{R}^{(\ell+k)}, \sigma_{\mathsf{pre}}, \mathbf{c}\cdot\mathbf{s}_1}) \leq e^{9.5/t+1/2t^2} = M$ and we obtain the corresponding $\varepsilon_{\mathsf{zk}}$ from Lemma 7. □

It remains to simulate the public keys of $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ and reduce the singing forgery of the adversary to SelfTargetMSIS.

Lemma 17 ([KLS18, Lem. 4.10])**.** *Let* $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ *be signature scheme defined in Figures 6 and 8 then*

$$\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\Pi_{\mathsf{MLDSA\text{-}Leaky}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{PR}}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{MLWE}}_{k,\ell,\mathcal{U}(S_\eta)}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{SelfTargetMSIS}}_{\mathsf{H_c}',k,\ell+1,B'}(\mathcal{B}_2)$$

where $B' = \max(\gamma_1 - \beta, \tau \cdot 2^{d-1} + 2\gamma_2 + 1)$.

Proof. We consider the $\mathsf{UF\text{-}NMA}$ experiment and begin by modifying the key generation procedure. First, we replace the pseudorandom values, which are generated from seeds via deterministic expansion algorithms (e.g., $\mathsf{ExpandA}$), with truly random values. Since, distinguishing the modified game means breaking security of the PRG, the advantage difference is bounded by that of an adversary $\mathcal{B}_0$ against the PR experiment, namely $\mathsf{Adv}^{\mathsf{PR}}(\mathcal{B}_0)$.

Next, we replace the public key $(\mathbf{A}, \mathbf{t})$ generated as $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ with a uniformly random $(\mathbf{A}, \mathbf{t})$. Any distinguisher between these two distributions yields an algorithm $\mathcal{B}_1$ that breaks the $\mathsf{MLWE}_{k,\ell,\mathcal{U}(S_\eta)}$ assumption.

Lastly, we analyse the experiment where the adversary receives a uniformly random $(\mathbf{A}, \mathbf{t})$ and outputs a valid message/signature pair $\mathsf{m}, (\tilde{c}, \mathbf{z}, \mathbf{h}, \mathbf{y}')$ which satisfies the verification condition $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$ and the equation $\tilde{c} = \mathsf{H_c}(\mu, \mathbf{w}_1')$, where

$$\mathbf{w}_1' = \mathsf{UseHint}(\mathbf{h}, \mathbf{w}'_{\mathsf{approx}}), \quad \mathbf{w}'_{\mathsf{approx}} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d \bmod q.$$

By Lemma 9, there exists a vector $\mathbf{u}$ such that $2\gamma_2 \cdot \mathbf{w}_1' = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d + \mathbf{u}$, with $\|\mathbf{u}\|_\infty \leq 2\gamma_2 + 1$. Using the decomposition $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + \mathbf{t}_0$, we rewrite:

$$\mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d + \mathbf{u} = \mathbf{A} \cdot \mathbf{z} - c(\mathbf{t} - \mathbf{t}_0) + \mathbf{u} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t} + (c\mathbf{t}_0 + \mathbf{u}) = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t} + \mathbf{u}'.$$

where $\mathbf{u}' = c\mathbf{t}_0 + \mathbf{u}$ and the worst case upper bound for $\mathbf{u}'$ is

$$\|\mathbf{u}'\|_\infty \leq \|c\|_1 \cdot \|\mathbf{t}_0\|_\infty + \|\mathbf{u}\|_\infty \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1.$$

Thus, an adversary who is successful at forging signature of a new message, is able to find $\mathbf{z}, c, \mathbf{u}', \mathsf{m}$ such that $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$, $\|\mathbf{u}'\|_\infty \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1$, and $\mathsf{m} \in \{0,1\}^*$ such that

$$\mathsf{H_c}(\mu, \frac{1}{2\gamma_2} \cdot [\mathbf{A} \mid \mathbf{t} \mid \mathbf{I}_k] \cdot \begin{bmatrix} \mathbf{z} \\ c \\ \mathbf{u}' \end{bmatrix}) = \tilde{c}$$

which yields a valid $\mathsf{SelfTargetMSIS}$ problem solution[1] $\mathbf{r}$ such that $\mathsf{H_c}'(\mu, \mathbf{A}' \cdot \mathbf{r}) = \tilde{c}$ where $\mathbf{r} = (\mathbf{z}, c, \mathbf{u}')$ with bound $B' = \|\mathbf{r}\|_\infty = \max(\|\mathbf{z}\|_\infty, \|\mathbf{u}'\|_\infty) = \max(\gamma_1 - \beta, \tau \cdot 2^{d-1} + 2\gamma_2 + 1)$. □

As part of proving unforgeability we need multiple arguments concerning the entropy of $\mathbf{w} = \mathbf{A}\mathbf{y}_0 + \mathbf{y}_1 \bmod q$ and $\mathbf{w}_1 = \mathsf{HighBits}(\mathbf{w})$. We improve the leftover hash lemma for the Discrete Gaussian distribution and we notice that our parameter sets imply $\mathbf{w}$ is Rényi close to uniform.

Lemma 18 (Entropy of $\mathbf{A}\mathbf{y}_0 + \mathbf{y}_1 \bmod q$)**.** *Let* $\mathcal{R}$ *be a power-of-2 cyclotomic ring of degree* n*, let* $q > 1$ *be a prime such that* $\mathcal{R}_q \sim \prod_{j=1}^{n} \mathbb{Z}_q/(X - r_j)$*. Let* $k \geq \ell \geq 1$*,* $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{k \times \ell})$ *and denote* $\mathbf{A}' = [\mathbf{A} \mid \mathbf{I}]$*. Let* $0 < \sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4(\ell+k)n)}}$ *and* $\mathbf{x} \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma}$*. Then*

$$\mathrm{RD}((\mathbf{A}', \mathbf{A}' \cdot \mathbf{y} \bmod q), (\mathbf{A}', \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^k))) \leq 2 \cdot \left(\frac{q^k}{\sigma^{\ell+k}} + 1\right)^n.$$

In particular, $\mathrm{RD}((\mathbf{A}', \mathbf{A}' \cdot \mathbf{y} \bmod q), (\mathbf{A}', \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^k))) \leq 2e$ *when* $\sigma^{k+\ell} \geq n \cdot q^k$.

[1]The hash function $\mathsf{H_c}'$ includes multiplying the integer input by $1/2\gamma_2$ as $\mathsf{H_c}'(\mu, x) = \mathsf{H_c}(\mu, \frac{1}{2\gamma_2}x)$.

Algorithm 7 $\mathsf{MLDSA.LeakySign}(\mathsf{sk} = (\rho, \mathsf{tr}, \mathbf{s}_1, \mathbf{s}_2), \mathsf{m} \in \{0,1\}^*; \mathsf{rnd}) \to \mathsf{sig}$

$\mu \in \{0,1\}^{64\cdot 8} = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
$(\rho_0'', \rho_1'') \in \{0,1\}^{128\cdot 8} = \mathsf{H}(\mathsf{rnd}, \mu)$
$\mathsf{counter} = 0$
$(\mathbf{z}, \mathbf{h}) = \perp$
while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
  $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) = \mathsf{MLDSA.SampleNonce}_{\sigma_{\mathsf{pre}}}(\rho_0'', \mathsf{counter})$
  $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\rho_1'', \mathsf{counter})$
  $\tilde{y} = (\rho_1'', \mathsf{counter})$
  $\mathbf{w} = \mathbf{A} \cdot (\mathbf{y}_1 + \mathbf{y}') + \mathbf{y}_2 \bmod q$
  $\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
  $\tilde{c} \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{c}}(\mu, \mathbf{w}_1)$
  $c \in \mathcal{R} = \mathsf{SampleInBall}_{\tau}(\tilde{c})$
  $\mathbf{z}_1 = \mathbf{y}_1 + c \cdot \mathbf{s}_1$
  $\mathbf{z}_2 = \mathbf{y}_2 + c \cdot \mathbf{s}_2$
  $p \leftarrow \mathsf{Ber}\left(\min\left(\frac{1}{M} \cdot \exp\left(\frac{-2\pi \cdot \mathsf{coeff}((\mathbf{z}_1, \mathbf{z}_2))^{\mathrm{T}} \cdot \mathsf{coeff}(c\cdot(\mathbf{s}_1, \mathbf{s}_2)) + \pi \cdot \|c\cdot(\mathbf{s}_1, \mathbf{s}_2)\|^2}{\sigma_{\mathsf{pre}}^2}\right), 1\right)\right)$
  if $p = 0$ **then**
    $(\mathbf{z}, \mathbf{h}) = \perp$
  else
    $\mathbf{e} = \mathbf{z}_2 - c\mathbf{t}_0$
    $\mathbf{z} = \mathbf{z}_1 + \mathbf{y}'$
    $\mathbf{h} = \mathsf{MakeHint}_{2\gamma_2}(\mathbf{e}, \mathbf{A} \cdot \mathbf{z} - c\mathbf{t}_1 \cdot 2^d)$ ▷ $= \mathsf{MakeHint}(\mathbf{e}, \mathbf{w} - \mathbf{e})$
    if $\|\mathbf{e}\|_\infty \geq \gamma_2 \vee \|\mathbf{h}\|_1 \geq w \vee \|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ **then**
      $(\mathbf{z}, \mathbf{h}) = \perp$
  $\mathsf{counter} = \mathsf{counter} + \ell$
return $\mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h}, \mathbf{w}, \tilde{y})$[a]

[a] Since we are using an algorithm with fixed coins to sample $\mathbf{y}'$ we include $\tilde{y} = (\rho_1'', \mathsf{counter})$ in the final signature instead of $\mathbf{y}'$. We also include $\mathbf{w}$ as part of the signature to help simulate the proofs in $\Pi_{\mathsf{MLDSA\text{-}ADA}}$, however in standalone $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ it can be omitted.

Figure 8: ML-DSA signing algorithm with nonce leakage.

Remark 3. *We note a significant difference between Lemma 2 and the lemma above. Now σ is raised to the power $k + \ell$ in the denominator hence this new result applies even in the setting when $k = \ell$ unlike the statement in previous work. We also note that the lemma above improves the condition on σ compare to previous leftover hash lemma for the Gaussian distribution e.g. [LPR13].*

Proof. By definition for any distribution $\mathcal{P}$ with support S we have $\mathrm{RD}(\mathcal{P}; \mathcal{U}(S)) = |S| \cdot \mathsf{Coll}(\mathcal{P})$. Hence start by bounding the collision probability of $\mathcal{P} = (\mathbf{A}', \mathbf{A}' \cdot \mathbf{y} \bmod q)$. For $\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{U}(\mathcal{R}_q^{k \times \ell})$ and $\mathbf{x}_0, \mathbf{x}_1 \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma}$ we have

$$\begin{aligned}\mathsf{Coll}(\mathcal{P}) &= \Pr(\mathbf{A}_0 = \mathbf{A}_1 \wedge [\mathbf{A}_0\, \mathbf{I}] \cdot \mathbf{x}_0 = [\mathbf{A}_1\, \mathbf{I}] \cdot \mathbf{x}_1) \\ &= \Pr(\mathbf{A}_0 = \mathbf{A}_1) \cdot \Pr([\mathbf{A}_0\, \mathbf{I}] \cdot (\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0} | \mathbf{A}_0 = \mathbf{A}_1)\end{aligned}$$

$$= \frac{1}{q^{k\cdot\ell\cdot n}} \cdot \Pr([\mathbf{A}_0\,\mathbf{I}](\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0}).$$

Given a vector $\alpha = (\alpha', \alpha'') = (\alpha_0, \dots \alpha_{k+\ell-1})$ consider the equation $\mathbf{A}_0\alpha' = \alpha'' \bmod q$. Let coefficients of $\alpha' = (\alpha_0, \dots, \alpha_{\ell-1})$ together generate an ideal $\mathcal{I} = \langle \alpha_0, \dots, \alpha_{\ell-1}\rangle$. Then every coefficient of $\mathbf{A}_0\alpha'$ is an element of $\mathcal{I}$. If any of $\alpha_\ell, \dots \alpha_{k+\ell-1}$ are not in $\mathcal{I}$ the equation above has no solutions. Otherwise, taking into account that rows of $\mathbf{A}_0$ are uniformly random we get $\Pr(\mathbf{A}_0\alpha' = \alpha'' \bmod q) = 1/|\mathcal{I}|^k$. Then over the randomness of $\mathbf{A}_0, \mathbf{x}_0, \mathbf{x}_1$ we get

$$\begin{aligned}
\Pr([\mathbf{A}_0\,\mathbf{I}](\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0}) &= \sum_{\mathcal{I}\subset\mathcal{R}_q} \Pr([\mathbf{A}_0\,\mathbf{I}](\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0} | \langle \mathbf{x}_{0,0} - \mathbf{x}_{1,0}, \dots, \mathbf{x}_{0,\ell-1} - \mathbf{x}_{1,\ell-1}\rangle = \mathcal{I}) \\
&\cdot \Pr(\langle \mathbf{x}_{0,0} - \mathbf{x}_{1,0}, \dots, \mathbf{x}_{0,\ell-1} - \mathbf{x}_{1,\ell-1}\rangle = \mathcal{I}) \\
&= \sum_{\mathcal{I}\subset\mathcal{R}_q} \frac{1}{|\mathcal{I}|^k} \cdot \Pr(\langle \mathbf{x}_{0,0} - \mathbf{x}_{1,0}, \dots, \mathbf{x}_{0,\ell-1} - \mathbf{x}_{1,\ell-1}\rangle = \mathcal{I}) \\
&\cdot \Pr(\forall i \in [\ell, \ell+k] : \mathbf{x}_{0,i} - \mathbf{x}_{1,i} \in \mathcal{I}) \\
&\leq \sum_{\mathcal{I}\subset\mathcal{R}_q} \frac{1}{|\mathcal{I}|^k} \cdot \Pr(\forall i \in [0, \ell+k] : \mathbf{x}_{0,i} - \mathbf{x}_{1,i} \in \mathcal{I}) \\
&= \sum_{\mathcal{I}\subset\mathcal{R}_q} \frac{1}{|\mathcal{I}|^k} \cdot \Pr(\mathbf{x}_0 = \mathbf{x}_1 \bmod \mathcal{I})
\end{aligned}$$

Now we note that we have obtained exactly the same formula as in the original proof of Lemma 2 in [BL25, Thm.5]. Hence applying the same results on the entropy of Gaussian distribution modulo ideals we get the statement. □

Lemma 19 (Unforgeability)**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1. Let $t \geq 1$, $M = e^{9.5/t+1/2t^2}$, $\varepsilon = 2^{-64}$, $\max(t \cdot \tau\eta\sqrt{2\pi(\ell+k)n}, n^{1/(k+\ell)} \cdot q^{k/(k+\ell)}) \leq \sigma_{\mathsf{pre}} \leq \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4(\ell+k)n)}}$, $\sigma_{\mathsf{ad}} > 0$ and $2^{-\alpha} = \sqrt{2e} \cdot \left(\frac{2\gamma_2+1}{q}\right)^{kn/2}$. Let $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ be the Leaky ML-DSA signature scheme defined in Figures 6 and 8. Then for any PPT adversary $\mathcal{A}$ with $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries, $\mathsf{Q}_{\mathsf{Sign}}$ signature queries and ADA signature scheme $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ defined in Figures 6 and 7 it holds*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}Leaky}}}(\mathcal{B}) &\leq \mathsf{Q}_{\mathsf{Sign}} \cdot 2^{-\lambda} + \mathsf{Q}_{\mathsf{Sign}} \cdot 2^{-\alpha} + \Big(\mathsf{Adv}^{\mathsf{PR}}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{MLWE}}_{k,\ell,\mathcal{U}(S_\eta)}(\mathcal{B}_1) \\
&\qquad + \mathsf{Adv}^{\mathsf{SelfTargetMSIS}}_{\mathsf{H}_{\mathsf{c}}',k,\ell+1,B'}(\mathcal{B}_2) + \mathsf{Q}_{\mathsf{H}_{\mathsf{c}}} \cdot \mathsf{Q}_{\mathsf{Sign}} \cdot 2^{-\alpha}\Big) \cdot e^{\left(\frac{\varepsilon}{1-\varepsilon}\cdot \mathsf{Q}_{\mathsf{Sign}}\right)}, \\
\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{A}) &\leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}Leaky}}}(\mathcal{B}) + \mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot \sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)}.
\end{aligned}$$

where $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{B}_i$ for $i \in [3]$ run in approximately the same time and $B' = \max(\gamma_1 - \beta, \tau \cdot 2^{d-1} + 2\gamma_2 + 1)$.

Proof. The first bound follows by combining Lemmas 15 to 17, which respectively capture the reduction from EUF-CMA to UF-NMA, the rejection sampling analysis, and the reduction to the underlying hardness assumptions.

For the second bound, we prove the claim via a series of hybrid experiments. The Hyb_0 is the standard EUF-CMA experiment for $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ as defined in Definition 11, where adversary's advantage is $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{A}) = \Pr[\mathsf{Hyb}_0 = 1]$.

Hyb_1. We modify Hyb_0 by programming the random oracle H_{ad} and answering the signing queries m as follows. On query m:

1. Set $\mathsf{ad} = \perp$
2. Run $\mathsf{MLDSA.LeakySign}$ using sk to obtain a signature $(\tilde{c}, \mathbf{z}, \mathbf{h}, \mathbf{w}, \tilde{y})$ on m.
3. Compute $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y})$, $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + \mathbf{t}_0$.
4. Compute $\mathbf{w}_{\mathsf{pr}} = \mathbf{w} - \mathbf{A} \cdot \mathbf{y}' \bmod q$.
5. Program the random oracle as $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk}) := \tilde{y}$.
6. Return $(\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h}), \pi = (\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0))$.

This simulation is identical to the real signing algorithm unless the adversary queries the point $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}_{\mathsf{pr}}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})$ before it is programmed. Since $\mathbf{w}_{\mathsf{pr}}$ does not occur in adversary's view prior to the signing response, by Lemma 11 each query contains $\mathbf{w}_{\mathsf{pr}}$ with probability at most $\sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)}$. By a union bound over all $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ queries, we obtain $|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot \sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)}$.

Reduction to $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$. We now construct an adversary $\mathcal{B}$ against the EUF-CMA security of $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$. Adversary $\mathcal{B}$ is given $\mathsf{pk} = (\rho, \mathbf{t}_0, \mathbf{t}_1)$ and access to the signing oracle for $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$. It runs $\mathcal{A}(\mathsf{pk}' = (\rho, \mathbf{t}_1))$ and simulates its environment as in Hyb_1:

- Direct H_{ad} queries of $\mathcal{A}$ are answered by lazy sampling and stored in a list.
- A signing query $(\mathsf{m}, \mathsf{ad})$ is answered as in Hyb_1. However, instead of using sk, the adversary $\mathcal{B}$ obtains $(\tilde{c}, \mathbf{z}, \mathbf{h}, \tilde{y})$ by querying m to $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$ and programs the H_{ad} according to the queried value ad. Finally, $\mathcal{B}$ produces simulated proof π as in Hyb_1.

Observe that the view of $\mathcal{A}$ in this simulation is identical to its view in Hyb_1. When $\mathcal{B}$ receives a forgery $(\mathsf{m}^*, \mathsf{sig}^* = (\tilde{c}^*, \mathbf{z}^*, \mathbf{h}^*))$ from $\mathcal{A}$ they pick arbitrary $\mathbf{w}^*, \tilde{y}^*$ and returns $(\mathsf{m}^*, \mathsf{sig}^* = (\tilde{c}^*, \mathbf{z}^*, \mathbf{h}^*, \mathbf{w}^*, \tilde{y}^*))$. Since $\mathbf{w}^*, \tilde{y}^*$ are not a part of the signature verification in $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$, if $\mathcal{A}$ succeeds in Hyb_1 then $\mathcal{B}$ succeeds in the $\mathsf{EUF\text{-}CMA}$ game for $\Pi_{\mathsf{MLDSA\text{-}Leaky}}$. Therefore, $\Pr[\mathsf{Hyb}_1 = 1] \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}Leaky}}}(\mathcal{B})$. Combining the above, we obtain the bound. □

4 Threshold ML-DSA with Gaussian Nonce

We build upon the ideas of [CdPE+26] and [DOTT22] to construct Threshold ML-DSA scheme with Discrete Gaussian nonce. We detail the full algorithm in Figure 9.

It is easy to see that the construction maintains compatibility with standard ML-DSA verification algorithm. Below we study the success probability of the signing algorithm and the unforgeability of the signature scheme. The correctness proof below is inspired by the argument in [CdPE+26, Theorem 3.1].

Lemma 20 (Signature Correctness)**.** *Consider the (T, N)-threshold signature scheme defined in Figure 9. Let K denote the number of signing sessions run in parallel. Let $B, \sigma_y > 0$, $\varepsilon = 2^{-64}$, $\forall i \in [N] : \|c \cdot \mathbf{s}_i\| \leq B$, $\mathrm{RD}^{\varepsilon}_{\infty}(\mathcal{D}_{\mathcal{R}^{k+\ell}, \sigma_y} \| \mathcal{D}_{\mathcal{R}^{k+\ell}, \sigma_y, c \cdot \mathbf{s}_i}) \leq M$. Let $\sigma_y^{k+\ell} \geq n \cdot q^k$ and $\sigma_y \leq \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4(\ell+k)n)}}$.*

For $j \geq 1$ let ν_j denote a random variable equal to 1 if given j successful sessions of rejection sampling in Algorithm 13 at least one of them reconstructs a valid signature after Algorithm 9[2]. Then the introduced scheme p-terminates according to the Definition 8 for

$$p \geq \left(1 + \frac{\varepsilon}{M-1}\right)^K \cdot \sum_{j \in K} \Pr(j \leftarrow \mathsf{Bin}(K, (1-1/M)^T)) \cdot \Pr(\nu_j = 1) - \sqrt{2e}K^2 \cdot \left(\frac{2\gamma_2+1}{q}\right)^{kn/2}.$$

Proof. Note that by design of Algorithm 9 any signature not equal $\perp$ passes the verification algorithm with probability 1. For an arbitrary choice of message m and set of participants $\mathsf{act} \subset [N]$ we proceed in a series of hybrids where $\mathsf{Hyb}_0 = \mathsf{Exp}_{\Pi}^{\mathsf{TS\text{-}corr}}(\lambda, \mathsf{m}, \mathsf{act})$. Let p_i denote the termination probability of Hyb_i with $p = p_0$.

Hyb_1: First we modify the experiment so that $c = \mathsf{SampleInBall}(\mathsf{H}_\mathsf{c}(\mu, \mathbf{w}_1))$ is now sampled uniformly from the challenge space $c \leftarrow \mathcal{C}$. The difference can only be detected if the value $\mathbf{w}_1 = \mathsf{HighBits}(\mathbf{w})$ is the same for two different parallel executions. By Lemma 18 $\mathrm{RD}((\mathbf{A}, \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2 \bmod q), (\mathbf{A}, \mathcal{U}(\mathcal{R}_q^k))) \leq 2e$. Then for $\mathbf{w} = \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2 \bmod q$ and $\mathbf{w}' = \mathbf{A}\mathbf{y}_1' + \mathbf{y}_2' \bmod q$ the probability $\mathsf{HighBits}(\mathbf{w}) = \mathsf{HighBits}(\mathbf{w}')$ is bounded by

$$\sqrt{2e \cdot \Pr(\mathsf{HighBits}(\mathbf{u}) = \mathsf{HighBits}(\mathbf{u}') \mid \mathbf{u}, \mathbf{u}' \leftarrow \mathcal{U}(\mathcal{R}_q^k))} \leq \sqrt{2e} \cdot \left(\frac{2\gamma_2+1}{q}\right)^{kn/2}.$$

The collision between any pair of session and the termination probability can be bounded by $p_0 \leq p_1 + \sqrt{2e}K^2 \cdot \left(\frac{2\gamma_2+1}{q}\right)^{kn/2}$.

Hyb_2: In this hybrid we simulate the signing transcripts using the secret independent distributions. By Lemma 8 for every $i \in \mathsf{act}$ we have $\mathrm{RD}_\infty(\mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y} \| \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y, c \cdot \mathbf{s}_i}) \leq M$. By Lemma 7 the Rényi Divergence between Hyb_6 and Hyb_5 is bounded by $\left(1 + \frac{\varepsilon}{M-1}\right)^K$. Consequently the termination probability is bounded as $p_1 \leq p_2 \cdot \left(1 + \frac{\varepsilon}{M-1}\right)^K$.

Hyb_3: For the ideal distribution of the signing algorithm the probability of returning a valid output is bounded by definition by $(1-1/M)^T$. Then the probability of success for rejection sampling for at least one of K session is $\mathsf{Bin}(K, (1-1/M)^T)$. It remains to incorporate the probability that the norm checks during the Combine procedure are satisfied, which we model with ν_i random variable. □

In the proof of unforgeability we follow the ideas of [CdPE+26, Theorem G.1]. However, the discrete Gaussian distribution of the nonce allows for tighter entropy analysis. We also modify the simulation of the signing transcript to Discrete Gaussian based rejection sampling. We start by bound the Rényi Divergence for imbalanced rejection sampling for Discrete Gaussian distributions.

Remark 4. *Let $\mathbf{v}, \mathbf{z} \in \mathcal{R}^m$, $M > 0$, $\Sigma \in \mathcal{R}^{m \times m}$ be a symmetric positive-definite covariance matrix and $\chi_r = \mathcal{D}_{\mathcal{R}^m, \Sigma, \mathbf{v}}, \chi_z = \mathcal{D}_{\mathcal{R}^m, \Sigma, \mathbf{0}}$, we get:*

$$\frac{\chi_z(\mathbf{z})}{M \cdot \chi_r(\mathbf{z})} = \frac{\exp(-\pi \mathbf{z}^\mathsf{T} \Sigma^{-1} \mathbf{z})}{M \cdot \exp(-\pi(\mathbf{z}-\mathbf{v})^\mathsf{T} \Sigma^{-1}(\mathbf{z}-\mathbf{v}))} = \frac{1}{M} \cdot \exp\left(-2\pi \mathbf{z}^\mathsf{T} \Sigma^{-1} \mathbf{v} + \pi \mathbf{v}^\mathsf{T} \Sigma^{-1} \mathbf{v}\right).$$

[2] We give a more comprehensive analysis of ν_j in §6.1 relying in part on a heuristic argument.

For Algorithm 13, $\mathbf{v} = (\mathbf{v}^{(0)}, \mathbf{v}^{(1)}), \mathbf{z} = (\mathbf{z}^{(0)}, \mathbf{z}^{(1)}) \in \mathcal{R}^{\ell+k}$*,* $\mathbf{\Sigma} = \mathsf{diag}(\sigma_y^2 \mathbf{I}_{\ell n}, \sigma_y'^2 \mathbf{I}_{kn})$*, and* $\chi_r = \mathcal{D}_{\mathcal{R}^\ell, \sigma_y, \mathbf{v}^{(0)}} \times \mathcal{D}_{\mathcal{R}^k, \sigma_y', \mathbf{v}^{(1)}}$*,* $\chi_z = \mathcal{D}_{\mathcal{R}^\ell, \sigma_y, 0} \times \mathcal{D}_{\mathcal{R}^k, \sigma_y', \mathbf{0}}$*, with* $\sigma_y, \sigma_y' > 0$*. Therefore,*

$$\frac{\chi_z(\mathbf{z})}{M \cdot \chi_r(\mathbf{z})} = \frac{1}{M} \exp\left(\frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z}^{(0)})^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(0)}) + \pi \left\|\mathbf{v}^{(0)}\right\|^2}{\sigma_y^2} + \frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z}^{(1)})^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(1)}) + \pi \left\|\mathbf{v}^{(1)}\right\|^2}{\sigma_y'^2} \right).$$

Lemma 21. *Let* $\sigma_y, \sigma_y' > 0$ *and* $\nu \geq 1$*. For* $\mathbf{v} = (\mathbf{v}^{(0)}, \mathbf{v}^{(1)}), \mathbf{z} = (\mathbf{z}^{(0)}, \mathbf{z}^{(1)}) \in \mathcal{R}^{\ell+k}$*,* $\chi_r = \mathcal{D}_{\mathcal{R}^\ell, \sigma_y, \mathbf{v}^{(0)}} \times \mathcal{D}_{\mathcal{R}^k, \sigma_y', \mathbf{v}^{(1)}}$*, and* $\chi_z = \mathcal{D}_{\mathcal{R}^\ell, \sigma_y, \mathbf{0}} \times \mathcal{D}_{\mathcal{R}^k, \sigma_y', \mathbf{0}}$*, with* $\sigma_y = \nu \cdot \sigma_y'$ *and* $\sigma_y' = t\sqrt{2\pi} \cdot \left\| (\frac{\mathbf{v}^{(0)}}{\nu}, \mathbf{v}^{(1)}) \right\|$ *for some* $t > 0$*, we have*

$$RD_\infty^{2^{-64}}(\chi_z || \chi_r) \leq \exp\left(\frac{9.5}{t} + \frac{1}{2\,t^2} \right).$$

Proof. From Remark 4 follows

$$\begin{aligned}
R &= \exp\left(\pi \frac{-2 \cdot \mathsf{coeff}(\mathbf{z}^{(0)})^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(0)}) + \left\|\mathbf{v}^{(0)}\right\|^2}{\nu^2 \sigma_y'^2} + \pi \frac{-2 \cdot \mathsf{coeff}(\mathbf{z}^{(1)})^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(1)}) + \left\|\mathbf{v}^{(1)}\right\|^2}{\sigma_y'^2} \right) \\
&= \exp\left(\pi \cdot \frac{-2\mathsf{coeff}(\mathbf{z}^{(0)}/\nu)^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(0)}/\nu) + \left\|\mathbf{v}^{(0)}/\nu\right\|^2 - 2\mathsf{coeff}(\mathbf{z}^{(1)})^\intercal \cdot \mathsf{coeff}(\mathbf{v}^{(1)}) + \left\|\mathbf{v}^{(1)}\right\|^2}{\sigma_y'^2} \right) \\
&= \exp\left(\pi \cdot \frac{-2 \cdot \mathsf{coeff}(\mathbf{z})^\intercal \cdot \mathsf{coeff}((\mathbf{v}^{(0)}/\nu, \mathbf{v}^{(1)})) + \left\|(\mathbf{v}^{(0)}/\nu, \mathbf{v}^{(1)})\right\|^2}{\sigma_y'^2} \right) \\
&= \exp\left(\frac{-2\pi \cdot \mathsf{coeff}((\mathbf{z}^{(0)}/\nu, \mathbf{z}^{(1)}))^\intercal \cdot \mathsf{coeff}((\mathbf{v}^{(0)}/\nu, \mathbf{v}^{(1)})) + \sigma_y'^2/2t^2}{\sigma_y'^2} \right) \\
&\leq \exp\left(\frac{\sigma_y' \cdot \sqrt{4\pi \cdot (\kappa+1)\ln 2} \left\|(\mathbf{v}^{(0)}/\nu, \mathbf{v}^{(1)})\right\| + \sigma_y'^2/2t^2}{\sigma_y'^2} \right) \\
&= \exp\left(9.5/t + 1/2t^2\right)
\end{aligned}$$

Here in the first inequality we use Lemma 6 and in the last equality we substitute $\kappa = 2^{64}$ and $\left\|(\mathbf{v}^{(0)}/\nu, \mathbf{v}^{(1)})\right\| = \sigma_y'/t\sqrt{2\pi}$. □

Lemma 22 ([CdPE+26, Lem. G.2])**.** *Let* $\mathbf{s} \in \mathcal{R}^{\ell+k}$*,* $c \in \mathcal{R}$ *and* $(\mathbf{z} \mid \mathsf{rej})_{c \cdot \mathbf{s}} = \mathbf{c} \cdot \mathbf{s} + \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y}$ *conditioned on* $b = 0$*, for* $b \leftarrow \mathsf{Ber}\left(\min\left(\frac{1}{M} \cdot \mathsf{ratio}, 1\right)\right)$ *and* $\mathsf{ratio} = \exp\left(\frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z}_i)^\intercal \cdot \mathsf{coeff}(\mathbf{v}) + \pi \cdot \|\mathbf{v}\|^2}{\sigma_y^2}\right)$*. Let* $\mathrm{RD}_\infty(\mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y} \| \mathbf{c} \cdot \mathbf{s} + \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y}) \leq M$ *for some* $M, \varepsilon \geq 0$*. Let* $\mathcal{A}$ *be an adversary against* $\mathsf{MLWE}_{q,k,\ell,(\mathbf{z}|\mathsf{rej})_{c\cdot\mathbf{s}}}$ *then there exist PPT adversaries* $\mathcal{B}_1, \mathcal{B}_2$ *that run in approximately the same time such that*

$$\mathsf{Adv}^{\mathsf{MLWE}_{q,k,\ell,(\mathbf{z}|\mathsf{rej})_{c\cdot\mathbf{s}}}}(\mathcal{A}) \leq \frac{M}{M-1}\left(2 \cdot \mathsf{Adv}^{\mathsf{MLWE}_{q,k,\ell,\mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}}}(\mathcal{A}) + \frac{\varepsilon}{1-\varepsilon}\right)$$

Lemma 23 (Threshold Unforgeability with Selective Corruptions)**.** *Let* Π_{Pebble} *the signature scheme defined in Figure 9. Let* $\tau, \eta, \gamma_2, q, \beta, n > 0$ *be as in Table 1 and* $\sigma_y, \sigma_{\mathsf{ad}} \geq 0$ *such that* $\sigma_y \geq$

$\eta_\varepsilon(\mathcal{R}^\ell)$ *and* $\sigma_2 := \left(1/\sigma_y^2 + 1/\sigma_{\mathsf{ad}}^2\right)^{-1/2} \geq \eta_\varepsilon(\mathcal{R}^\ell)$. *Let* $t \geq 1$, $M = e^{9.5/t+1/2t^2}$, $\varepsilon = 2^{-64}$. *Let* $B \leq \tau\eta N\sqrt{n\cdot(k+\ell)}$ *such that* $\|c\cdot\mathbf{s}_i\| \leq B$ *with probability* $1-\mathsf{negl}(\lambda)$ *for all* $i \in N$ *and* $\mathbf{c}, \{\mathbf{s}_i\}$ *generated in Algorithms 8 and 13. Let* $\sigma_y > \max(t\cdot\sqrt{2\pi}\cdot B, n^{1/k+\ell}\cdot q^{k/(k+\ell)})$, $\sigma_y \leq \frac{\sqrt{\pi}}{2}\cdot\frac{q}{\sqrt{\ln(4(\ell+k)n)}}$. *Then for any PPT adversary* $\mathcal{A}$ *with* $\mathsf{Q}_{\mathsf{H}_\mathsf{c}}, \mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}$ *random oracle queries,* Q_Sign *signature queries it holds*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{Pebble}}}(\mathcal{A}) \leq\ & \mathsf{Q}_\mathsf{Sign}\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}\cdot\sqrt{2e}\cdot q^{-nk/2} + {\mathsf{Q}_\mathsf{Sign}}^2\mathsf{Q}_{\mathsf{H}_\mathsf{c}}\cdot\left(\frac{2\gamma_2+1}{q}\right)^{-nk/2}\sqrt{2e} + \mathsf{Q}_{\mathsf{H}_\mathsf{c}}\mathsf{Q}_\mathsf{Sign}\cdot 2^{-\lambda} \\
&+ \frac{T\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}\mathsf{Q}_\mathsf{Sign} + (\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}+\mathsf{Q}_\mathsf{Sign})^2}{2^{2\lambda}} + {\mathsf{Q}_{\mathsf{H}_\mathsf{c}}}^2\cdot 2^{-512} + \mathsf{negl}(\lambda) \\
&+ \left(1+\frac{\varepsilon}{1-\varepsilon}\right)^{\mathsf{Q}_\mathsf{Sign}}\cdot\left(2\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\mathcal{U}(S_\eta)}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA}}}(\mathcal{B}_1)\right) \\
&+ \mathsf{Q}_\mathsf{Sign}\cdot\frac{3M-2}{M-1}\cdot\left(2\cdot\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\mathcal{D}_{\mathcal{R},\sigma_y}}(\mathcal{B}_2) + \frac{\varepsilon}{1-\varepsilon}\right).
\end{aligned}$$

where $\mathcal{A}$ *and* $\mathcal{B}_i$ *for* $i \in [2]$ *run in approximately the same time.*

Proof. We proceed in a series of hybrids where the $\mathsf{Hyb}_0 = \mathsf{Exp}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{Pebble}},\mathcal{A}}$ for a given PPT adversary $\mathcal{A}$. We denote the advantage of the Adversary in Hyb_i by ε_i where $\varepsilon_0 = \mathsf{Adv}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{TSIG}}}(\mathcal{A})$

Hyb_1**:** In this hybrid for each signing query of the adversary we delay computing $\mathbf{w}_i$ of the honest parties to Sign_1. By Lemma 18 we have $\mathbf{w}_i$ is Rényi close to uniform. Then $\varepsilon_0 \leq \varepsilon_1 + \mathsf{Q}_\mathsf{Sign}\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}\cdot\sqrt{2e}\cdot q^{-nk/2}$.

Hyb_2**:** We sample the challenge $\mathbf{c}$ first and then program the random oracle $\mathsf{H}_\mathsf{c}(\mu, \mathbf{w}_1) = \mathbf{c}$. The difference between Hyb_1 and Hyb_2 ca only be detected if $\mathbf{w}_1$ was previously queried to the random oracle, since $\mathbf{w}_i$ of the honest parties are Rényi to uniform the entropy of $\mathsf{HighBits}(\sum_i \mathbf{w}_i)$ is at least $\left(\frac{2\gamma_2+1}{q}\right)^{-nk/2}\cdot\sqrt{2e}$. Then $\varepsilon_1 \leq \varepsilon_2 + {\mathsf{Q}_\mathsf{Sign}}^2\mathsf{Q}_{\mathsf{H}_\mathsf{c}}\cdot\left(\frac{2\gamma_2+1}{q}\right)^{-nk/2}\sqrt{2e}$.

Hyb_3**:** In Hyb_3 the experiment aborts if the adversary can find a second preimage for any queried commitment produced by H_cmt and if H_cmt has a collision. By the properties of the random oracle we have $\varepsilon_2 \leq \varepsilon_3 + \frac{T\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}\mathsf{Q}_\mathsf{Sign} + (\mathsf{Q}_{\mathsf{H}_\mathsf{cmt}}+\mathsf{Q}_\mathsf{Sign})^2}{2^{2\lambda}}$.

Hyb_4**:** In this hybrid we change the order of sampling of $\mathbf{z}_i$ and $\mathbf{w}_i$ for the honest parties. Now we sample $\mathbf{z}_i = c\cdot\mathbf{s}_i + \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}$ and compute $\mathbf{w}_i = \mathbf{A}\cdot\mathbf{z}_i - c\cdot\mathbf{t}$. This change in the distributions is only syntactic, hence the advantage of the Adversary does not change.

Hyb_5**:** For a fixed $\mathbf{v}$ we denote $(\mathbf{z}_i \mid \mathsf{acc})_\mathbf{v} = \mathbf{v} + \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}$ conditioned on $b=1$, for $b \leftarrow \mathsf{Ber}\left(\min\left(\frac{1}{M}\cdot\mathsf{ratio}, 1\right)\right)$ as defined in Algorithm 13 and $\mathsf{ratio} = \exp\left(\frac{-2\pi\cdot\mathsf{coeff}(\mathbf{z}_i)^\intercal\cdot\mathsf{coeff}(\mathbf{v})+\pi\cdot\|\mathbf{v}\|^2}{\sigma_y^2}\right)$. Similarly we define $(\mathbf{z}_i \mid \mathsf{rej})_\mathbf{v} = \mathbf{v} + \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}$ conditioned on $b=0$. In Hyb_6 we first sample $\mathbf{z}_i$ and change the distribution of $\mathbf{w}_i$ only for rejected $\mathbf{z}_i$. If $\mathbf{z}_i$ is rejected we sample $\mathbf{w}_i \leftarrow \mathcal{U}(\mathcal{R}_q^k)$. Since rejected $\mathbf{z}_i$ are never revealed to the Adversary this change is indistinguishable based MLWE

problem with the secret distributed as $(\mathbf{z}_i \mid \mathsf{rej})_{c\cdot\mathbf{s}_i}$. By Lemma 22 the advantage change as $\varepsilon_4 \leq \varepsilon_5 + \frac{M}{M-1}\left(2 \cdot \mathsf{Adv}^{\mathsf{MLWE}_{q,k,\ell,\mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}}}(\mathcal{B}_2) + \frac{\varepsilon}{1-\varepsilon}\right)$.

Hyb_6**:** We now also simulate the transcripts for the accepted signatures as follows

1: Sample $(\mathbf{z}_i) \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}$, $c \leftarrow \mathcal{C}$
2: Compute $\mathbf{w}_i = [\mathbf{A}\ \mathbf{I}] \cdot \mathbf{z}_i - c\mathbf{t}$
3: **return** $(c, \mathbf{w}_i, \mathbf{z}_i)$

For each honest party by Lemma 8 we have $\mathrm{RD}_\infty(\mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y} \| \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y,c\cdot\mathbf{s}_i}) \leq M$ then by Lemma 7 the Rényi Divergence between Hyb_6 and Hyb_5 is bounded by $\left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^{\mathsf{Q}_{\mathsf{Sign}}}$ so the advantage of the adversary changes by at most $\varepsilon_5 \leq \varepsilon_6 \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^{\mathsf{Q}_{\mathsf{Sign}}}$.

Hyb_7**:** Now the signing queries of the advresary are simulate without the secret key of the honest parties. Note that the public key $\mathbf{t} = \sum_i [\mathbf{A}\ \mathbf{I}] \cdot \mathbf{s}_i$ where at least on the shares is only known to the honest signers. Let us denote this share $\mathbf{s}_h$. In this hybrid we change the distribution of $[\mathbf{A}\ \mathbf{I}] \cdot \mathbf{s}_h$ to $\mathcal{U}(\mathcal{R}_q^k)$, hence, $\mathbf{t}$ is now also distributed as $\mathcal{U}(\mathcal{R}_q^k)$. By the MLWE assumption the advantage of the adversary changes as $\varepsilon_6 \leq \varepsilon_7 + \mathsf{Adv}^{\mathsf{MLWE}_{q,k,\ell,\mathcal{U}(S_\eta^{k+\ell})}}(\mathcal{B}_0)$.

Hyb_8**:** Consider an adversary $\mathcal{B}_1$ against $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA}}}$ with public key $(\mathbf{A}', \mathbf{t}')$ We set the public key $\mathbf{t} = \mathbf{t}'$ instead the uniformly distributed value and use $\mathbf{A} = \mathbf{A}'$ as the public matrix. It is easy to see that a valid forgery $(\mathbf{z}, \tilde{c}, \mathbf{h})$ in Hyb_7 corresponds to a valid forgery for $\mathcal{B}_1$ in $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA}}}$. Hence $\varepsilon_8 \leq \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\mathcal{U}(S_\eta)}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA}}}(\mathcal{B}_1)$ which completes the proof. □

5 Threshold ML-DSA with Associated Data

We now extend the associated data authenticated ML-DSA construction from §3 to the threshold setting. Our construction builds on the threshold Gaussian ML-DSA signing protocol presented in §4, which allows us to equip the associated data features.

5.1 Construction

We present the signing and verification with associated data algorithms in Figure 10. The remaining algorithms like $\mathsf{Pebble.KeyGen}$, $\mathsf{Pebble.Combine}$ and verification of the signature $\mathsf{Pebble.Verify}$ stay same from the Figure 9.

The main idea remains identical to the single-signer construction in Figure 7. During signing, the parties first jointly compute the public nonce commitment $\mathbf{w}^{(\mathsf{pr})} = \sum_{i\in\mathsf{act}} \mathbf{w}_i$, which is independent of the associated data. Now, to bind the associated data (ad) to the signature, each party use $\mathbf{w}^{(\mathsf{pr})}$ to derive the same tweak $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}\left(\mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})\right)$, and compute the same $\mathbf{w}^{(\mathsf{tw})} = \mathbf{A}\mathbf{y}'$. Consequently, the final commitment becomes $\mathbf{w} = \mathbf{w}^{(\mathsf{pr})} + \mathbf{w}^{(\mathsf{tw})}$.

Unlike the single-signer construction, the threshold setting requires the tweak $\mathbf{y}'$ to be incorporated into the distributed response in $\mathsf{Pebble.Sign\text{-}AD}_2$ without changing the underlying threshold signing protocol. To achieve this, only one designated signer (here, the party with index 1) adds the

Algorithm 8 $\mathsf{Pebble.KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$

$\rho \leftarrow \{0,1\}^{256}$
$\mathbf{A} \in \mathcal{R}_q^{k \times \ell} = \mathsf{ExpandA}(\rho)$
$(\mathbf{s}, \{\mathbf{s}_i\}_{i \in [N]}) = \mathsf{RSS}(N, T, \eta)$
$\mathbf{t} = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{s} \bmod q$
$(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}_d(\mathbf{t})$
$\mathsf{pk} = (\rho, \mathbf{t}_1)$
$\mathsf{tr} \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{pk})$
$\mathsf{sk} = \{\mathsf{sk}_i = (\rho, \mathsf{tr}, \mathbf{t}_0, \mathbf{s}_i)\}_{i \in N}$
return $(\mathsf{pk}, \mathsf{sk})$

Algorithm 9 $\mathsf{Pebble.Combine}(\mathsf{pk} = (\rho, \mathbf{t}_1), \mathsf{m}, \mathsf{act}, \left\{\mathsf{contrib}_i^{(j)}\right\}_{i,j}) \to \mathsf{sig}$

$\mathbf{A} = \mathsf{ExpandA}(\rho)$
$\mu = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
$\mathbf{w} = \sum_{j \in \mathsf{act}} \mathbf{w}_j$
$\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
$\tilde{c} = \mathsf{H_c}(\mu, \mathbf{w}_1)$
$c = \mathsf{SampleInBall}(\tilde{c})$
$\mathbf{z}^{(0)} = \sum_{i \in \mathsf{act}} \mathbf{z}_i^{(0)}$
$\mathbf{e} = \mathbf{w} - (\mathbf{A} \cdot \mathbf{z}^{(0)} - 2^d \cdot c\,\mathbf{t}_1)$
$\mathbf{h} = \mathsf{MakeHint}_{2\gamma_2}(\mathbf{e}, \mathbf{A} \cdot \mathbf{z}^{(0)} - 2^d \cdot c\,\mathbf{t}_1)$
if $\left\|\mathbf{z}^{(0)}\right\|_\infty \geq \gamma_1 - \beta \vee \|\mathbf{e}\|_\infty > \gamma_2 \vee \|\mathbf{h}\|_1 > w$ **then**
  $\mathsf{sig} = \perp$
return $\mathsf{sig} = (\tilde{c}, \mathbf{z}^{(0)}, \mathbf{h})$

Algorithm 10 $\mathsf{Pebble.Verify}(\mathsf{pk} = (\rho, \mathbf{t}_1), \mathsf{m}, \mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h})) \to \{0,1\}$

$\mathbf{A} = \mathsf{ExpandA}(\rho)$
$\mu = \mathsf{H}(\mathsf{tr} = \mathsf{H}(\mathsf{pk}), \mathsf{m})$
$c = \mathsf{SampleInBall}(\tilde{c})$
$\mathbf{w}'_{\mathsf{approx}} = \mathbf{A} \cdot \mathbf{z}^{(0)} - c\mathbf{t}_1 \cdot 2^d \bmod q$
$\mathbf{w}'_1 = \mathsf{UseHint}_{2\gamma_2}(\mathbf{h}, \mathbf{w}'_{\mathsf{approx}})$
$\tilde{c}' = \mathsf{H_c}(\mu, \mathbf{w}'_1)$
return $[[\|\mathbf{z}\|_\infty < \gamma_1 - \beta \wedge \tilde{c} = \tilde{c}']]$

Algorithm 11 $\mathsf{Pebble.Sign}_0(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{act}) \to \mathsf{contrib}_i^{(0)}$

$\mu \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
$\mathbf{y}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k}, \sigma_y}$
$\mathbf{w}_i = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{y}_i \bmod q$
$\mathsf{cmt}_i = \mathsf{H}_{\mathsf{cmt}}(\mathsf{pk}, i, \mathbf{w}_i)$
return $\mathsf{contrib}_i^{(0)} = \mathsf{cmt}_i$

Algorithm 12 $\mathsf{Pebble.Sign}_1(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{act}) \to \mathsf{contrib}_i^{(1)}$

return $\mathsf{contrib}_i^{(1)} = \mathbf{w}_i$

Algorithm 13 $\mathsf{Pebble.Sign}_2(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{act}) \to \mathsf{contrib}_i^{(2)}$

assert$\{\forall j \in \mathsf{act} : \mathsf{cmt}_j = \mathsf{H}_{\mathsf{cmt}}(\mathsf{pk}, j, \mathbf{w}_j)\}$
$\mathbf{w} = \sum_{j \in \mathsf{act}} \mathbf{w}_j$
$\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
$\tilde{c} \in \{0,1\}^{2\lambda} = \mathsf{H_c}(\mu, \mathbf{w}_1)$
$c \in \mathcal{R} = \mathsf{SampleInBall}_\tau(\tilde{c})$
$\mathbf{s}_i = \mathsf{RSSRecover}(\mathsf{act}, i)$
$\mathbf{z}_i = \mathbf{y}_i + c \cdot \mathbf{s}_i$
$\mathbf{\Sigma} = \mathsf{diag}(\sigma_y^2 \mathbf{I}_{\ell n}, \sigma_y'^2 \mathbf{I}_{kn})$
$\mathsf{ratio} = \exp\big(-2\pi \mathsf{coeff}(\mathbf{z}_i)^{\mathrm{T}} \mathbf{\Sigma}^{-1} \mathsf{coeff}(c \cdot \mathbf{s}_i) + \pi \mathsf{coeff}(c \cdot \mathbf{s}_i)^{\mathrm{T}} \mathbf{\Sigma}^{-1} \mathsf{coeff}(c \cdot \mathbf{s}_i)\big)$
if $0 \leftarrow \mathsf{Ber}\left(\min\left(\frac{1}{M} \cdot \mathsf{ratio}, 1\right)\right)$ **then**
  $\mathsf{contrib}_i^{(2)} = \perp$
$(\mathbf{z}_i^{(0)}, \mathbf{z}_i^{(1)}) = \mathbf{z}_i \in \mathcal{R}_q^\ell \times \mathcal{R}_q^k$
return $\mathsf{contrib}_i^{(2)} = \mathbf{z}_i^{(0)}$

Figure 9: Threshold ML-DSA algorithms with Gaussian Nonce.

tweak $\mathbf{y}'$ to its ℓ-dimensional response share $\mathbf{z}_i^{(0)}$ in $\mathsf{Pebble.Sign\text{-}AD}_2$, so that the aggregated response $\mathbf{z}^{(0)} = \sum_{i\in\mathsf{act}\setminus\{1\}} \mathbf{z}_i^{(0)} + \mathbf{z}_1^{(0)}$; which implies the combined response $\mathbf{z}^{(0)} = (\sum_{i\in\mathsf{act}} \mathbf{y}_i) + c\sum_{i\in\mathsf{act}} \mathbf{s}_i + \mathbf{y}'$, is distributed exactly as in the single-party construction.

To verify the proof $\pi = (\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)$, we allow the $\mathsf{Verify\text{-}AD}$ to proceed as in the single-party ad-verification in Algorithm 6, except that now the verifier should increase the bound for the norm check of $\|\mathbf{w}_{\mathsf{approx}} - \mathbf{w}\|$ to $0.53 \cdot \sigma_{\mathsf{pre}} \cdot \sqrt{T \cdot kn}$ to incoporate the $|\mathsf{act}|(\leq T)$ no. of error factors $c\mathbf{s}_2 + \mathbf{y}_2$ in $\mathbf{w}_{\mathsf{approx}} - \mathbf{w}$. We discuss the soundness of the verification of the associated data below in Lemma 24.

5.2 Security

The security of the threshold associated-data scheme in Figure 10 follows closely from that of the single-party AD-construction in Figure 7. Since the associated-data authentication is implemented by applying a tweak $\mathbf{y}'$ sampled from the same distribution to the jointly generated nonce commitment $\mathbf{w}$, the entropy argument from Lemma 11 naturally carry over to the threshold setting. In particular, the associated-data properties of the threshold scheme, as well as the unforgeability of the signature sig, reduce directly to their single-party counterparts.

Below, we first show that an honestly generated signature sig and associated-data proof π produced by $\mathsf{Pebble.Combine}$ pass the associated-data verification algorithm with overwhelming probability.

Lemma 24 (Soundness)**.** *Let $\Pi_{\mathsf{Pebble\text{-}ADA}}$ be the ADA signature scheme defined in Figures 9 and 10. Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1. Let $\sigma_{\mathsf{ad}}, \sigma_{\mathsf{pre}} \geq 0$ and $\varepsilon = 2^{-64}$. Then for $(\mathsf{sig}, \pi) \leftarrow \mathsf{Pebble.Combine}(\mathsf{pk}, \mathsf{m}, \mathsf{act}, \{\mathsf{contrib}_i^{(j)}\}_{i,j})$*

$$\Pr(\mathsf{Pebble.Verify\text{-}AD}(\mathsf{pk}, \mathsf{m}, \mathsf{ad}, \mathsf{sig}, \pi = (\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)) = 1) \geq (1 - 2^{-\lambda} + 8\varepsilon') \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^T$$

Proof. Let $\mathbf{x}_i = c\mathbf{s}_{i,2} + \mathbf{y}_{i,2}$, where for every honest signer i, $\mathrm{RD}_\infty(\mathbf{x}_i \| \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}, \mathbf{0}}) \leq 1 + \frac{\varepsilon}{1-\varepsilon}$ using Lemma 7. The verifier computes $\mathbf{w}_{\mathsf{approx}} = \mathbf{w} - \sum_{i=1} \mathbf{x}_i$, and accepts whenever $\|\sum_{i=1} \mathbf{x}_i\| \leq 0.53 \cdot \sigma_{\mathsf{pre}} \sqrt{T \cdot kn}$. We proceed in the following series of hybrids. In Hyb_0, we have the real distribution

$$\mathcal{D}_0 = \{\sum_i \mathbf{x}_i \mid \mathbf{x}_i \sim \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}, c\mathbf{s}_{i,2}} \ \forall i\}.$$

Hyb_j $(1 \leq j \leq |\mathsf{act}|)$. In this hybrid, we replace the first j variables $\mathbf{x}_i (i = 1, \ldots, j)$ by zero centered discrete gaussian. Namely,

$$\mathcal{D}_j = \left\{ \sum_i \mathbf{x}_i \mid \mathbf{x}_i \sim \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}, \mathbf{0}} \text{ for } 1 \leq i \leq j, \text{ and } X_i \sim \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}, c\mathbf{s}_{i,2}} \text{ for } i \geq j+1 \right\}.$$

where $\mathrm{RD}_\infty(\mathcal{D}_j \| \mathcal{D}_{j+1}) \leq 1 + \frac{\varepsilon}{1-\varepsilon}$. Then, by probability preservation, $\Pr[\mathsf{Hyb}_j = 1] \leq \Pr[\mathsf{Hyb}_{j+1} = 1] \cdot (1 + \frac{\varepsilon}{1-\varepsilon})$ then , $\Pr[\mathsf{Hyb}_0 = 1] \leq \Pr[\mathsf{Hyb}_T = 1] \cdot (1 + \frac{\varepsilon}{1-\varepsilon})^T$ since $j \leq |\mathsf{act}| \leq T$, where in $\mathcal{D}_T$, all variables are independent centered discrete Gaussians, i.e.,

$$\mathcal{D}_T = \{\sum_i \mathbf{x}_i \mid \mathbf{x}_i \sim \mathcal{D}_{\mathcal{R}^k, \sigma_{\mathsf{pre}}, 0} \ \forall i\}$$

Algorithm 14 $\mathsf{Pebble.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y}) \to \mathcal{R}^{\ell}$

1: **return** $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^{\ell},\sigma_{\mathsf{ad}}}(\mathsf{coins} = (\tilde{y}))$

Algorithm 15 $\mathsf{Pebble.Sign\text{-}AD}_0(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{ad}, \mathsf{act}) \to \mathsf{contrib}_i^{(0)}$

1: $\mu \in \{0,1\}^{1024} = \mathsf{H}(\mathsf{tr}, \mathsf{m})$
2: $\mathbf{y}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_{\mathsf{pre}}}$
3: $\mathbf{w}_i = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{y}_i \bmod q$
4: $\mathsf{cmt}_i \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{cmt}}(\mathsf{pk}, i, \mathbf{w}_i)$
5: **return** $\mathsf{contrib}_i^{(0)} = \mathsf{cmt}_i$

Algorithm 16 $\mathsf{Pebble.Sign\text{-}AD}_1(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{ad}, \mathsf{act}) \to \mathsf{contrib}_i^{(1)}$

1: **return** $\mathsf{contrib}_i^{(1)} = \mathbf{w}_i$

Algorithm 17 $\mathsf{Pebble.Sign\text{-}AD}_2(\mathsf{sk}_i, \mathsf{m} \in \{0,1\}^*, \mathsf{ad}, \mathsf{act}) \to \mathsf{contrib}_i^{(2)}$

1: **assert**$\{\forall j \in \mathsf{act} : \mathsf{cmt}_j = \mathsf{H}_{\mathsf{cmt}}(\mathsf{pk}, j, \mathbf{w}_j)\}$
2: $\mathbf{w}^{(\mathsf{pr})} = \sum_{j \in \mathsf{act}} \mathbf{w}_j$
3: $\tilde{y} \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})$
4: $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y})$
5: $\mathbf{w}^{(\mathsf{tw})} = \mathbf{A} \cdot \mathbf{y}' \bmod q$
6: $\mathbf{w} = \mathbf{w}^{(\mathsf{pr})} + \mathbf{w}^{(\mathsf{tw})}$
7: $\mathbf{w}_1 = \mathsf{HighBits}_{2\gamma_2}(\mathbf{w})$
8: $\tilde{c} \in \{0,1\}^{2\lambda} = \mathsf{H}_{\mathsf{c}}(\mu, \mathbf{w}_1)$
9: $c \in \mathcal{R} = \mathsf{SampleInBall}_{\tau}(\tilde{c})$
10: $\mathbf{s}_i = \mathsf{RSSRecover}(\mathsf{act}, i, \mathsf{sk}_i)$
11: $\mathbf{z}_i = \mathbf{y}_i + c \cdot \mathbf{s}_i$
12: $p \leftarrow \mathsf{Ber}\left(\min\left(\frac{1}{M} \cdot \exp\left(\frac{-2\pi \cdot \mathsf{coeff}(\mathbf{z}_i)^{\intercal} \cdot \mathsf{coeff}(c \cdot \mathbf{s}_i) + \pi \cdot \|c \cdot \mathbf{s}_i\|^2}{\sigma_{\mathsf{pre}}^2}\right), 1\right)\right)$
13: **if** $p = 0$ **then**
14: $\quad \mathsf{contrib}_i^{(2)} = \perp$
15: $(\mathbf{z}_i^{(0)}, \mathbf{z}_i^{(1)}) = \mathbf{z}_i \in \mathcal{R}_q^{\ell} \times \mathcal{R}_q^{k}$
16: **if** $i = 1$ **then** set $\mathbf{z}_i^{(0)} = \mathbf{z}_i^{(0)} + \mathbf{y}'$
17: **return** $\mathsf{contrib}_i^{(2)} = (\mathbf{z}_i^{(0)}, \pi = (\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0))$

Algorithm 18 $\mathsf{Pebble.Verify\text{-}AD}(\mathsf{pk}, \mathsf{m}, \mathsf{ad}, \mathsf{sig} = (\tilde{c}, \mathbf{z}, \mathbf{h}), \pi = (\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)) \to \{0,1\}$

1: $\mathbf{A} = \mathsf{ExpandA}(\rho)$
2: $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + \mathbf{t}_0$
3: $\mu = \mathsf{H}(\mathsf{tr} = \mathsf{H}(\mathsf{pk}), \mathsf{m})$
4: $c = \mathsf{SampleInBall}(\tilde{c})$
5: $\mathbf{w}_{\mathsf{approx}} = \mathbf{A} \cdot \mathbf{z} - c\mathbf{t} \bmod q$ $\quad \triangleright = \mathbf{w} - c\mathbf{s}_2 - \sum_i \mathbf{y}_{i,2}$ computed from sig and π
6: $\tilde{y} = \mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{m}, \mathsf{pk})$
7: $\mathbf{y}' = \mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y})$
8: $\mathbf{w}^{(\mathsf{tw})} = \mathbf{A} \cdot \mathbf{y}' \bmod q$
9: $\mathbf{w} = \mathbf{w}^{(\mathsf{pr})} + \mathbf{w}^{(\mathsf{tw})}$ $\quad \triangleright = \mathbf{w}$ computed from π and ad.
10: **return** $[[\|\mathbf{w}_{\mathsf{approx}} - \mathbf{w}\| \le 0.53 \cdot \sigma_{\mathsf{pre}} \cdot \sqrt{T \cdot kn}]]$

Figure 10: Threshold Associated data signing and verification.

Hyb_{T+1}. In this hybrid we replace $\sum_i \mathbf{x}_i$ by $\mathbf{x} \sim \mathcal{D}_{\mathcal{R}^k,\sqrt{T}\sigma_{\mathsf{pre}},0}$. By Lemma 4, we have $\mathrm{SD}(\sum_i \mathbf{x}_i, \mathcal{D}_{\mathcal{R}^k,\sqrt{T}\sigma_{\mathsf{pre}},0}) \leq 8\varepsilon'$ when $\mathbf{x}_i \sim \mathcal{D}_{\mathcal{R}^k,\sigma_{\mathsf{pre}},0}$. Therefore, $\Pr[\mathsf{Hyb}_0 = 1] \leq (\Pr[\mathsf{Hyb}_{T+1} = 1] + 8\varepsilon') \cdot (1 + \frac{\varepsilon}{1-\varepsilon})^T$ Now, by Corollary 1, $\|\mathbf{x}\| \leq 0.53 \cdot \sigma_{\mathsf{pre}}\sqrt{T \cdot kn}$ holds with probability $1 - 2^{-\lambda}$, and therefore $\Pr[\mathsf{Hyb}_0 = 1] \geq (1 - 2^{-\lambda} + 8\varepsilon') \cdot \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^T$, which proves the claim. □

As in the single-party ADA scheme, our threshold ADA scheme satisfies only the Weak-AD-Binding property and not the Strong-AD-Binding property, for the same reason as in the single-party case: we cannot assume that the adversary's matrix $\mathbf{A}$ is uniformly distributed.

Lemma 25 (Weak AD-Binding)**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1. Let $\sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \geq 0$. Then for any PPT adversary $\mathcal{A}$ making atmost $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries, the ADA signature scheme $\Pi_{\mathsf{Pebble\text{-}ADA}}$ defined in Figures 9 and 10 satisfies*

$$\mathsf{Adv}^{\mathsf{Weak\text{-}AD\text{-}Binding}}_{\Pi_{\mathsf{Pebble\text{-}ADA}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{Weak\text{-}AD\text{-}Binding}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{B})$$

Proof. Here, since $|\mathsf{cor}| < T$, every signing session run by $\mathcal{A}$ requires the participation of at least one honest party; $\mathcal{B}$ simulates the (at most $T-1$) corrupted parties itself using the shares it holds, and simulates the missing honest contribution $(\mathbf{w}_h, \mathbf{z}_h)$ by forwarding the session to its own Sign-AD oracle.

As $\mathbf{w}_{(\mathsf{pr})} = \sum_{i \in \mathsf{act}} \mathbf{w}_i$ and the pair (sig, π) produced by Pebble.Combine (Algorithm 9) have exactly the same distribution as in the correspoding execution of the single party scheme $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ under the aggregated secret key. Therefore, the view of $\mathcal{A}$ is perfectly indistinguishable from a real execution of $\Pi_{\mathsf{Pebble\text{-}ADA}}$. Hence, any winning tuple $(\mathsf{m}, \mathsf{sig}, \mathsf{ad}, \pi, \mathsf{ad}', \pi')$ for $\mathcal{A}$ against $\Pi_{\mathsf{Pebble\text{-}ADA}}$ is also a winning tuple for $\mathcal{B}$ against $\Pi_{\mathsf{MLDSA\text{-}ADA}}$. □

The AD-Hiding property of the threshold scheme follows analogously and reduces directly to the AD-Hiding of $\Pi_{\mathsf{MLDSA\text{-}ADA}}$.

Lemma 26 (AD-Hiding)**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1. Let $\sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \geq 0$. Then for any PPT adversary $\mathcal{A}$ making atmost $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ random oracle queries, the ADA signature scheme $\Pi_{\mathsf{Pebble\text{-}ADA}}$ defined in Figures 9 and 10 satisfies*

$$\mathsf{Adv}^{\mathsf{AD\text{-}Hiding}}_{\Pi_{\mathsf{Pebble\text{-}ADA}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{AD\text{-}Hiding}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{B})$$

Proof. Adversary $\mathcal{B}$ against AD-Hiding of $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ simulates the threshold protocol for $\mathcal{A}$ of $\Pi_{\mathsf{Pebble\text{-}ADA}}$. On $\mathcal{A}$'s challenge $(\mathsf{m}, \mathsf{ad}_0, \mathsf{ad}_1)$, $\mathcal{B}$ forwards $(\mathsf{m}, \mathsf{ad}_0, \mathsf{ad}_1)$ to its own challenger to obtain $(\mathsf{sig}, \pi) \leftarrow \mathsf{Sign\text{-}AD}(\mathsf{m}, \mathsf{ad}_b)$. Now, $\mathcal{B}$ can forward (sig, π) to $\mathcal{A}$ to continue the simulation. Therefore, if $\mathcal{A}$ outputs a guess b' with non-negligible probability, $\mathcal{B}$ outsputs the same guess, succeding with the same probability. □

Now, it remains to prove the properties of underlying signature. For the unforgrability of the signature sig, we closely follows the hybrids of the (non-AD) threshold ML-DSA unforgeability from Lemma 23 and make the additional hybrids, isolating the associated-data (ad) layer, since our AD-scheme has dependence on $\mathsf{ad}, \mathsf{H}_{\mathsf{ad}}, \mathbf{y}'$.

Lemma 27 (Unforgeability adapted from [CdPE+26, Theorem G.1])**.** *Let $\tau, \eta, \gamma_2, q, \beta, n > 0$ be as in Table 1 and $\sigma_{\mathsf{pre}}, \sigma_{\mathsf{ad}} \geq 0$ such that $\sigma_{\mathsf{pre}} \geq \eta_\varepsilon(\mathcal{R}^\ell)$ and $\sigma_2 \coloneqq \left(1/\sigma^2_{\mathsf{pre}} + 1/\sigma^2_{\mathsf{ad}}\right)^{-1/2} \geq \eta_\varepsilon(\mathcal{R}^\ell)$. Let $B \leq \tau\eta N\sqrt{n \cdot (k+\ell)}$ such that $\|c \cdot \mathbf{s}_i\| \leq B$ with probability $1 - \mathsf{negl}(\lambda)$ for all $i \in N$ over the*

randomness of $\mathbf{c}, \{\mathbf{s}_i\}$. *Let* $t \geq 1$, $M = e^{9.5/t+1/2t^2}$, $\varepsilon = 2^{-64}$, $\max(t \cdot \sqrt{2\pi} \cdot B, n^{1/k+\ell} \cdot q^{k/(k+\ell)}) < \sigma_{\mathsf{pre}} < \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sqrt{\ln(4(\ell+k)n)}}$. *Let* $\Pi_{\mathsf{MLDSA\text{-}ADA}}$ *the signature scheme defined in Figures 6 and 7. Then for any PPT adversary* $\mathcal{A}$ *with* $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ *random oracle queries,* $\mathsf{Q}_{\mathsf{Sign}}$ *signature queries and ADA signature scheme* $\Pi_{\mathsf{Pebble\text{-}ADA}}$ *defined in Figures 9 and 10 it holds*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{Pebble\text{-}ADA}}}(\mathcal{A}) \leq\ & \mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{cmt}}} \cdot \sqrt{2e} \cdot q^{-nk/2} + {\mathsf{Q}_{\mathsf{Sign}}}^2\mathsf{Q}_{\mathsf{H}_{\mathsf{c}}} \cdot \left(\frac{2\gamma_2+1}{q}\right)^{nk/2} \sqrt{2e} \\
& + \mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot \sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)} + \mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot 2^{-2\lambda} \\
& + \mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{c}}} \cdot 2^{-2\lambda} + \frac{T\mathsf{Q}_{\mathsf{H}_{\mathsf{cmt}}}\mathsf{Q}_{\mathsf{Sign}} + (\mathsf{Q}_{\mathsf{H}_{\mathsf{cmt}}} + \mathsf{Q}_{\mathsf{Sign}})^2}{2^{2\lambda}} + {\mathsf{Q}_{\mathsf{H}_{\mathsf{c}}}}^2 \cdot 2^{-512} + \mathsf{negl}(\lambda) \\
& + \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^{\mathsf{Q}_{\mathsf{Sign}}} \cdot (2\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\mathcal{U}(S_\eta)}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}(\mathcal{B}_1)) \\
& + \mathsf{Q}_{\mathsf{Sign}} \cdot \frac{3M-2}{M-1} \cdot \left(2 \cdot \mathsf{Adv}^{\mathsf{MLWE}}_{q,k,l,\mathcal{D}_{\mathcal{R},\sigma_{\mathsf{pre}}}}(\mathcal{B}_2) + \frac{\varepsilon}{1-\varepsilon}\right).
\end{aligned}$$

where $\mathcal{A}$ *and* $\mathcal{B}_i$ *for* $i \in [2]$ *run in approximately the same time.*

Proof. We sketch the proof for our AD-scheme by proceeding as the hybrids of the non-AD proof (Lemma 23).

Hyb_0. This is $\mathsf{Exp}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{Pebble\text{-}ADA}},\mathcal{A}}$ (defined in Figure 4a), where the signing oracle $\mathcal{O}\mathsf{Sign}_r$ on query $(\mathsf{m}, \mathsf{ad}, \mathsf{act})$ runs $\mathsf{Pebble.Sign\text{-}AD}_r$ (Algorithms 15 to 17) to answer queries. We denote the advantage of adversary in Hyb_i by $\Pr[\mathsf{Hyb}_i = 1]$, therefore, $\mathsf{Adv}^{\mathsf{sel\text{-}EUF\text{-}CMA}}_{\Pi_{\mathsf{Pebble\text{-}ADA}}}(\mathcal{A}) = \Pr[\mathsf{Hyb}_0 = 1]$.

Hyb_1. Identical to the non-AD proof Lemma 23, where the commitment entropy is $\sqrt{2e} \cdot q^{-nk/2}$, so the loss is : $\Pr[\mathsf{Hyb}_0 = 1] \leq \Pr[\mathsf{Hyb}_1 = 1] + \mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{cmt}}} \cdot \sqrt{2e} \cdot q^{-nk/2}$.

Hyb_{2a}. We modify the computation of the honest $\mathsf{contrib}^{(2)}_i$ in $\mathsf{Pebble.Sign\text{-}AD}_2$ (Algorithm 17). Once all commitments are opened and $\mathbf{w}^{(\mathsf{pr})} := \sum_i \mathbf{w}_i$ is fixed, we replace the honest computation of $\mathbf{y}'$ as follows:

1. if $(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})$ was already queried to H_{ad} by $\mathcal{A}$, set a flag $\mathsf{f}_{\mathsf{fail}} := \top$ and abort;
2. otherwise sample $\tilde{y} \leftarrow \{0,1\}^{2\lambda}$ and $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\sigma_{\mathsf{ad}}}$ freshly and program $\mathsf{H}_{\mathsf{ad}}(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk}) := \tilde{y}$ and $\mathsf{MLDSA.Sample\text{-}AD}_{\sigma_{\mathsf{ad}}}(\tilde{y}) = \mathbf{y}'$; then proceed to compute $\mathbf{w} := \mathbf{w}^{(\mathsf{pr})} + \mathbf{w}^{(tw)}$.

This does not change the distribution of $(\mathsf{contrib}^{(2)}_i)$ unless $\mathcal{A}$ queried (1) the random oracle H_{ad} on the relevant $(\mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0, \mathsf{ad}, \mathsf{pk})$ before it was programmed and (2) $\mathsf{MLDSA.Sample\text{-}AD}$ on $\tilde{y}$ before it was programmed. In case of (1), we use the same entropy argument from Hyb_1 for $\mathbf{w}^{(\mathsf{pr})}$, so a union bound over $\mathsf{Q}_{\mathsf{Sign}}$ sessions and $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ queries gives a loss of $\mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot \sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)}$ by Lemma 11. Now, in case of (2), it is easy to see the loss is $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot 2^{-2\lambda}$ over $\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}}$ queries since $\tilde{y}$ is sampled uniformly from $\{0,1\}^{2\lambda}$. Hence $\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_{2a} = 1]| \leq \mathsf{Q}_{\mathsf{Sign}}\mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot \sqrt{\frac{1}{\sigma^{\ell n}} \cdot 2\exp(1)} + \mathsf{Q}_{\mathsf{H}_{\mathsf{ad}}} \cdot 2^{-2\lambda}$.

Hyb_2–Hyb_5. Identical to Hyb_2–Hyb_5 of non-AD proof in Lemma 23. Here, we emphasise that in Hyb_5, we keep the same loss following the same Réyni argument, since the rejection sampling step is unaffected by the AD-twist, as $\mathbf{y}'$ is added to $\mathbf{z}^{(0)}_i$ after the rejection test.

Hyb_6 Here, we simulate the transcripts for the accepted signatures given $(\mathsf{pk}, \mathbf{t}_0)$ as follows:

1. Sample $\mathsf{cmt}_i \leftarrow \{0,1\}^{2\lambda}$, $\mathbf{y}' \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\sigma_{\mathsf{ad}}}$, $c \leftarrow \mathcal{C}$ and $\mathbf{z}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y}$
2. Compute $\mathbf{w}_i = [\mathbf{A}\ \mathbf{I}] \cdot \mathbf{z}_i - c\mathbf{t}$ and $\mathbf{w}^{(\mathsf{pr})} = \sum_i \mathbf{w}_i$
3. **if** $i = 1$ **return** $(\mathsf{cmt}_i, \mathbf{w}_i, c, \mathbf{z}_i^{(0)} + \mathbf{y}', \mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)$
4. **else return** $(\mathsf{cmt}_i, \mathbf{w}_i, c, \mathbf{z}_i^{(0)}, \mathbf{w}^{(\mathsf{pr})}, \mathbf{t}_0)$

For each honest party, by Lemma 8 we have $\mathrm{RD}_\infty(\mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y} \| \mathcal{D}_{\mathcal{R}^{\ell+k},\sigma_y,c\cdot\mathbf{s}_i}) \le M$, then by Lemma 7 the Rényi Divergence between Hyb_6 and Hyb_5 is bounded by $(1+\frac{\varepsilon}{1-\varepsilon})^{\mathsf{Q}_{\mathsf{Sign}}}$ so the advantage of the adversary changes by at most $\Pr[\mathsf{Hyb}_5 = 1] \le \Pr[\mathsf{Hyb}_6 = 1] \cdot (1+\frac{\varepsilon}{1-\varepsilon})^{\mathsf{Q}_{\mathsf{Sign}}}$.

Now, we have everything independent of secret share $\mathbf{s}_i$, and by Hyb_7 from Lemma 23, we can also replace the public key $\mathbf{t}$ by an uniform value in $\mathcal{U}(\mathcal{R}_q^k)$.

Reduction to $\Pi_{\mathsf{MLDSA\text{-}ADA}}$. Consider an adversary $\mathcal{B}_1$ against $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}$ with public key $(\mathbf{A}', \mathbf{t}_1')$. $\mathcal{B}_1$ first issues one query $\mathsf{Sign\text{-}AD}(\mathsf{m}_0, \mathsf{ad}_0)$ on an arbitrary message, and reads $\mathbf{t}_0$ from the returned proof $\pi_0 = (\mathbf{w}_{(\mathsf{pr})}, \mathbf{t}_0)$. It then sets $\mathbf{A} \coloneqq \mathbf{A}'$, $\mathbf{t} \coloneqq \mathbf{t}_1' \cdot 2^d + \mathbf{t}_0$. Now, it is easy to see that a valid forgery $(\tilde{c}^*, \mathbf{z}^*, \mathbf{h}^*, \mathsf{m}^*)$ in Hyb_7 is also a valid forgery for $\mathcal{B}_1$ in $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MLDSA\text{-}ADA}}}$. Hence, combining all the bounds, completes the proof. □

Lemma 28 (Correctness)**.** *For any message* m*, associated data* ad *and valid* $(\mathsf{sk}, \mathsf{pk})$ *for the signature scheme* $\Pi_{\mathsf{Pebble\text{-}ADA}}$ *defined in Figures 9 and 10. Let* $\varepsilon = 2^{-64}$ *and* $(T \cdot \sigma_{\mathsf{pre}} + \sigma_{\mathsf{ad}}) \cdot \sqrt{\ell n} \le \gamma_1 - \beta$*,* $\tau \cdot 2^{d-1} + T \cdot \sigma_{\mathsf{pre}} \sqrt{kn} \le \gamma_2$*. Then every signing attempt* $\mathsf{sig} \leftarrow \mathsf{Pebble.Combine}(\mathsf{sk}, \mathsf{m}, \mathsf{ad})$ *returns a valid signature with probability*

$$\Pr(\mathsf{sig} \neq \perp) \ge 1 - T \cdot \frac{M - 1 + \varepsilon}{M} - (T+1) \cdot 2^{-2\ell n} \cdot (1 + \frac{\varepsilon}{1-\varepsilon}) - T \cdot 2^{-2kn} \cdot (1 + \frac{\varepsilon}{1-\varepsilon}),$$

and for every $\perp \neq \mathsf{sig} \leftarrow \mathsf{Pebble.Combine}(\mathsf{sk}, \mathsf{m}, \mathsf{ad})$ *we have* $\mathsf{Pebble.Verify}(\mathsf{pk}, \mathsf{m}, \mathsf{sig}) = 1$ *with probability* 1*.*

Proof. The proof follows the same argument as the single party version in Lemma 14. □

6 Parameters

In this section we describe the choices we make to instantiate the parameters of Pebble as well as Pebble-ADA.

6.1 Threshold ML-DSA with Gaussian Nonce

Note that the Combine algorithm in Algorithm 9 contains the same norm checks as the verification procedure. Then, any successfully reconstructed signature passes the verification algorithm with probability 1. Hence, we only analyse the success probability of the signing protocol. From Lemma 20 it follows that larger ν_i and smaller M imply a better success probability of the signing protocol. By Lemma 21, $\sigma_y \ge \sqrt{2\pi} \cdot t \cdot \max\|[\mathbf{s}_1/\nu, \mathbf{s}_2] \cdot c\|$ implies a lower bound on $M \ge e^{9.5/t + 1/2t^2}$. Hence, as σ_y grows the value of M decreases. On the other hand, the correctness constraints that

affect ν_i are

$$\begin{aligned}
&\sigma'_y \cdot \sqrt{\kappa \cdot T} \leq \gamma_1 - \beta, \\
&\sqrt{\kappa\, T} \cdot \sigma_y + \max(\|c \cdot \mathbf{t}_0\|_\infty) \leq \gamma_2 - \beta, \\
&\|\mathbf{h}\|_1 \leq w
\end{aligned}$$

so increasing σ_y lowers ν_i. We compute a suitable trade-off for σ_y in Table 2. We note that parameters in Table 2 rely on two heuristic arguments. Denote shares_i the set of shares from replicated secret sharing a party is using in the current signing session. For $\mathbf{s}_i \leftarrow \sum_{\mathsf{shares}_i} S_\eta^{k+\ell}$ and $c \leftarrow \mathcal{U}(\{c = \sum c_i X^i \mid c_i \in \{-1, 0, 1\}, \mathsf{HW}(c) \leq \tau\})$ a provable bound on the norm $\|c \cdot \mathbf{s}_i\| \leq \sqrt{n} \cdot \|c\| \cdot \|[\mathbf{s}_1, \mathbf{s}_2]\| \leq n \cdot \eta \cdot \sqrt{\tau \cdot (k+\ell)} \cdot |\mathsf{shares}_i|$ is loose compared to the experimental behavior of the product[3]. To circumvent this issue the literature uses heuristics. In [DKL+18, Appendix B] the authors repeatedly sample random matrices $\mathsf{rot}(\mathbf{s}_i)$ and compute the median (denote it $B_{\max}$) of the largest singular value distribution. Then we conjecture that $s_{\max}(\mathsf{rot}(\mathbf{s}_i)) \leq B_{\max}$ with probability at least 1/2 and add a check of this form to the key generation in Algorithm 1. Note that if our conjecture is correct the check in the key generation increases the success probability of the Adversary against MLWE problem with secret $\mathbf{s}_i$ by at most one bit in Lemma 19. Now $\|c \cdot \mathbf{s}_i\| \leq \|c\| \cdot s_{\max}(\mathsf{rot}(\mathbf{s}_i)) \leq \sqrt{\tau} \cdot B_{\max}$. On the other hand the work of [CdPE+26] uses a different heuristic. In [CdPE+26, Section 3.4] the authors conjecture that $\|c \cdot \mathbf{s}_i\|$ behaves as a Gaussian random variable with standard deviation $\sigma = 0.1 \cdot \mathbb{E}(\|c\|) \cdot \mathbb{E}(\|[\mathbf{s}_1, \mathbf{s}_2]\|)$ and take the tail bound corresponding to 13 standard deviations. If the authors conjecture is correct the bound $\|c \cdot [\mathbf{s}_1, \mathbf{s}_2]\| \leq 1.3 \cdot \sqrt{\tau} \cdot \sqrt{n(k+\ell)} \cdot \sqrt{\frac{\eta(\eta+1)}{3}} \cdot \sqrt{|\mathsf{shares}|}$ holds with probability at least $1 - 2^{-121}$ by the Chernoff bound $\Pr(\|c \cdot [\mathbf{s}_1, \mathbf{s}_2]\| \geq \kappa) \leq e^{-\frac{\kappa^2}{2\sigma^2}}$. In this case if the heuristic is incorrect, the number of allowed signature queries in the security proof would decrease by the corresponding amount[4]. It is worth noting that reducing the number of tolerated signing queries can also be translated into the leakage on $\mathbf{s}_i$ and a higher advantage against the MLWE problem even though the dependency is more difficult to describe as a closed form expression. We think that the singular value heuristic has a more predictable effect on the construction, however, in Table 2 we include the results for both heuristics.

The other numerical argument only affects the success probability of the signing algorithm. The worst case analysis of $\|c \cdot \mathbf{t}_0\|_\infty$ for equations $\sqrt{\kappa\, T} \cdot \sigma_y + \max(\|c \cdot \mathbf{t}_0\|_\infty) \leq \gamma_2 - \beta$ and $\|\mathbf{h}\|_1 \leq w$ does not correspond to the success probability we see in practice. Hence, we do not lower bound success probability ν_i from Lemma 20, but we set σ_y to be small enough so that the observed success probability is maximised. In the table below, (N, T) are the total number of participants and the signing threshold, $\varepsilon = 2^{-64}$, $K \cdot \mathsf{Q}_{\mathsf{Sign}} = 2^{64}$.

In all reported tables, (t, n) is the signing threshold and total number of parties, and k is the number of parallel repetitions per outer attempt. The widths σ_y, σ'_y on the complementary component and σ_{ad} parameterize the discrete-Gaussian ExpandMask, while M is the Bernoulli repetition factor used in rejection sampling. The quantity p_{outer} is the empirical success probability of a single outer signing attempt, estimated from N_{start} attempt starts and N_{rej} rejection-sampling failures over 200 independent sessions ($p_{\mathrm{outer}} = 1 - N_{\mathrm{rej}}/N_{\mathrm{start}}$). The counters N_{bern} and N_{ℓ_2} record local per-iteration Gaussian rejects (Bernoulli and ℓ_2 gate, respectively); N_{skip} counts combine

[3] For example for the first line in Table 2 the provable bound would be 9044 instead of 585 or 425 heuristic results.

[4] For example if the norm is α times larger than the conjectured value the security proof would only allow for $2^{64/\alpha^4}$ queries.

(T,N)	$\|c\cdot\mathbf{s}\|_\nu \le$	σ_y	σ'_y	M
(2,2)	585	12000	12000	2.99
(2,3)	850	12000	12000	4.95
(3,3)	585	12000	12000	2.99
(2,2)	1400	30000	30000	2.86
(2,3)	1884	30000	30000	4.12
(3,3)	1400	30000	30000	2.86
(2,2)	860	30000	30000	1.91
(2,3)	1290	30000	30000	2.63
(3,3)	860	30000	30000	1.91

(a) [DKL+18] heuristic.

(T,N)	$\|c\cdot\mathbf{s}\|_\nu \le$	σ_y	σ'_y	M
(2,2)	334.1	50000	12000	1.86
(2,3)	472.5	50000	12000	2.43
(3,3)	334.1	50000	12000	1.95
(2,2)	873.7	200000	30000	2.00
(2,3)	1235.6	200000	30000	2.68
(3,3)	883.9	165000	30000	2.64
(2,2)	559.8	200000	30000	1.57
(2,3)	791.6	200000	30000	1.88
(3,3)	566.6	165000	30000	1.86

(b) [CdPE+26] heuristic.

Table 2: Pebble parameters.

iterations discarded because at least one party contributed $\perp$. Among iterations that reach combine, N_z, N_{γ_2}, and N_ω are rejects of the standard ML-DSA checks on the aggregated z-norm, high-order bits (γ_2), and hint weight (ω).

Finally, the outer-cause columns partition the failed outer attempts counted by N_{rej} according to why that attempt aborted. In the Pebble tables this is $N_z^{\text{out}}+N_\omega^{\text{out}}+N_\perp^{\text{out}} = N_{\text{rej}}$; in the Pebble-ADA tables the combine-check failures are aggregated into a single column, so $N_{\text{cmb}}^{\text{out}}+N_\perp^{\text{out}} = N_{\text{rej}}$. Here N_z^{out} / N_ω^{out} (or $N_{\text{cmb}}^{\text{out}}$) count attempts that reached combine but still failed an ML-DSA check and restarted, whereas $N_\perp^{\text{out}}$ counts attempts that never produced a usable contribution slot (Bernoulli / combine-skip). For example, in the Pebble row ML-DSA-44 $(2,2)$ one has $N_{\text{rej}} = 440$ with $N_z^{\text{out}} = 2$, $N_\omega^{\text{out}} = 30$, and $N_\perp^{\text{out}} = 408$, and $2+30+408 = 440$; likewise, in the Pebble-ADA Table 4(a) row ML-DSA-44 $(2,3)$ one has $N_{\text{rej}} = 134$ with $N_{\text{cmb}}^{\text{out}} = 14$ and $N_\perp^{\text{out}} = 120$, and $14+120 = 134$.

We evaluate Pebble under an *imbalanced* discrete-Gaussian ExpandMask, sampling the ℓ- and k-components of the masking nonce with widths σ_y and σ'_y, respectively, and applying the corresponding Bernoulli repetition factor M.

For each ML-DSA security level and threshold pair $(t,n) \in \{(2,2),(2,3),(3,3)\}$, we run 200 independent signing sessions and report the empirical per-outer-attempt success probability p_{outer}. Across the suite, p_{outer} ranges from 0.29 to 0.51, with the strongest configurations being ML-DSA-87 at $(3,3)$ ($p_{\text{outer}} = 0.514$) and ML-DSA-44 at $(2,2)$ ($p_{\text{outer}} = 0.476$). Failures are dominated by local Gaussian rejection (Bernoulli / missing contributions) rather than γ_2 combine checks, indicating that the chosen (σ_y, σ'_y, M) keep correctness checks largely inactive while rejection sampling remains the main cost.

6.2 Threshold ML-DSA with Associated Data

The construction with associated data follows the same constraints for rejection sampling parameters. In addition the binding constraint from Lemma 12 requires

$$\sqrt{2e}\cdot\left(\frac{\sqrt{2\pi e}\cdot(1+2\cdot 0.53\cdot\sigma_{\mathsf{pre}}\sqrt{T})}{\sigma_{\mathsf{ad}}}\right)^{kn/2} \le 2^{-2\lambda}.$$

ML-DSA-DG	(t,n)	k	σ_y	σ'_y	M	p_{outer}	N_{start}	N_{rej}	N_{bern}	N_{ℓ_2}	N_{skip}	N_z	N_{γ_2}	N_ω	N_z^{out}	N_ω^{out}	$N_\perp^{\text{out}}$
44	$(2,2)$	2	50000	12000	1.86	0.4762	840	440	797	0	1198	2	0	40	2	30	408
44	$(2,3)$	3	50000	12000	2.43	0.4098	976	576	1704	0	2396	4	0	44	4	34	538
44	$(3,3)$	4	50000	12000	1.95	0.3257	1842	1242	3591	0	6363	120	0	147	105	126	1011
65	$(2,2)$	3	200000	30000	2.00	0.3610	1108	708	1686	0	2500	0	0	332	0	252	456
65	$(2,3)$	5	200000	30000	2.68	0.3082	1298	898	4096	0	5640	2	0	350	0	276	622
65	$(3,3)$	9	165000	20000	2.64	0.2907	2064	1464	11618	0	17580	15	0	192	15	162	1287
87	$(2,2)$	3	200000	30000	1.57	0.4082	980	580	1089	0	1780	2	0	560	2	346	232
87	$(2,3)$	4	200000	30000	1.88	0.3899	1026	626	1870	0	2914	12	0	570	8	344	274
87	$(3,3)$	6	165000	20000	1.86	0.5141	1167	567	3206	0	5853	12	0	279	6	195	366

Table 3: Pebble with [CdPE+26] heuristic

For all parameter sets (k, n, λ) of MLDSA-44, MLDSA-65, MLDSA-87 this constraint holds for $\sigma_{\mathsf{ad}} \geq 5.86 + 5.34 \cdot \sigma_{\mathsf{pre}}\sqrt{T}$. We set σ_{ad} exactly equal to this lower bound. The lower bound on σ_{ad} affects the correctness upper bounds as

$$\sqrt{\kappa \cdot (T \cdot \sigma_{\mathsf{pre}}^2 + \sigma_{\mathsf{ad}}^2)} \leq \gamma_1 - \beta$$
$$\sqrt{\kappa \cdot (T \cdot \sigma_{\mathsf{pre}}^2 + (5.86 + 5.34 \cdot \sigma_{\mathsf{pre}}\sqrt{T})^2)} \leq \gamma_1 - \beta$$
$$\sqrt{\kappa \cdot T \cdot \sigma_{\mathsf{pre}}^2 (1 + (5.35)^2)} \leq \sigma_{\mathsf{pre}} \cdot 5.36 \cdot \sqrt{\kappa \cdot T} \leq \gamma_1 - \beta$$

We state full parameter choices in Table 4. We denote (N, T) the total number of participants and the signing threshold. We state the results for both the singular value heuristic and the heuristic used in [CdPE+26]. For all instantiations we take $\varepsilon = 2^{-64}$, $\mathsf{Q}_{\mathsf{Sign}} = 2^{64}$.

(T,N)	$\|c \cdot \mathbf{s}\| \leq$	σ_{pre}	σ_{ad}	M
(2,2)	585	$2000 \cdot 6$	$15110 \cdot 6$	3.217
(2,3)	850	$2000 \cdot 6$	$15110 \cdot 6$	5.488
(3,3)	585	$1633 \cdot 6$	$15110 \cdot 6$	4.192
(2,2)	1400	$6500 \cdot 4$	$49094 \cdot 4$	3.638
(2,3)	1884	$6500 \cdot 4$	$49094 \cdot 4$	5.709
(3,3)	1400	$5307 \cdot 4$	$49092 \cdot 4$	4.876
(2,2)	860	$6500 \cdot 4$	$49094 \cdot 4$	2.206
(2,3)	1290	$6500 \cdot 4$	$49094 \cdot 4$	3.285
(3,3)	860	$5307 \cdot 4$	$49092 \cdot 4$	2.638

(a) [DKL+18] heuristic.

(T,N)	$\|c \cdot \mathbf{s}\| \leq$	σ_{pre}	σ_{ad}	M
(2,2)	425	$2000 \cdot 6$	$15110 \cdot 6$	2.34
(2,3)	600	$2000 \cdot 6$	$15110 \cdot 6$	3.32
(3,3)	425	$1633 \cdot 6$	$15110 \cdot 6$	2.83
(2,2)	1116	$6500 \cdot 4$	$49094 \cdot 6$	2.795
(2,3)	1578	$6500 \cdot 4$	$49094 \cdot 6$	4.292
(3,3)	1116	$5307 \cdot 4$	$49092 \cdot 6$	3.528
(2,2)	721	$6500 \cdot 4$	$49094 \cdot 6$	1.94
(2,3)	1019	$6500 \cdot 4$	$49094 \cdot 6$	2.555
(3,3)	721	$5307 \cdot 4$	$49092 \cdot 6$	2.253

(b) [CdPE+26] heuristic.

Table 4: Pebble-ADA parameters.

We evaluate Pebble-ADA under the Gaussian ExpandMask / ExpandAD parameterization of Table 3, reporting empirical signing success for ML-DSA-44/65/87 and threshold pairs $(t, n) \in \{(2,2), (2,3), (3,3)\}$.

Each configuration uses $k = 150$ parallel repetitions and is measured over 200 independent sessions; we report the per-outer-attempt success probability p_{outer}.

Singular-value heuristic (Table 4(a)). With the larger norm bound on $\|c \cdot \mathbf{s}\|$ and the corresponding Bernoulli factors M, success rates vary substantially across the suite: p_{outer} ranges from 0.105 (ML-DSA-65, $(3,3)$) to 0.985 (ML-DSA-44, $(2,2)$). ML-DSA-44 remains strong at $(2,2)$ and $(2,3)$ (0.985 and 0.749), but degrades at $(3,3)$ (0.466). For ML-DSA-65 and ML-DSA-87, $(2,2)$ achieves $p_{\text{outer}} \approx 0.63$, while $(2,3)$ and especially $(3,3)$ drop sharply (0.341–0.362 and 0.105–0.281, respectively). In the low-success regimes, outer failures are dominated by missing contributions ($N_{\perp}^{\text{out}}$), consistent with heavy Bernoulli rejection under the larger M.

Mithril heuristic (Table 4(b)). Replacing the singular-value $\|c \cdot \mathbf{s}\|$ with the Mithril heuristic yields strictly smaller M at the same $(\sigma_{\mathsf{pre}}, \sigma_{\text{ad}})$ and improves success across all configurations. Here p_{outer} ranges from 0.232 to 1.000: ML-DSA-44 reaches near-ideal rates (1.000, 0.990, 0.889 for $(2,2)$, $(2,3)$, $(3,3)$), while ML-DSA-65/87 improve to 0.800/0.761 at $(2,2)$, 0.512/0.571 at $(2,3)$, and 0.232/0.376 at $(3,3)$. As in Table 4(a), residual failures concentrate in Bernoulli / combine-skip events rather than γ_2 checks. Overall, both heuristics keep Pebble-ADA practical for small thresholds, with the Mithril parameterization providing a clear gain in p_{outer}; $(3,3)$ at higher security levels remains the most expensive setting and is the main target for further (σ, M) tuning.

ML-DSA-AD	(t,n)	k	σ_{pre}	σ_{ad}	M	p_{outer}	N_{start}	N_{rej}	N_{bern}	N_{ℓ_2}	N_{skip}	N_z	N_{γ_2}	N_ω	$N_{\text{cmb}}^{\text{out}}$	$N_{\perp}^{\text{out}}$
44	$(2,2)$	150	$2000 \cdot 6$	$15110 \cdot 6$	3.217	0.9852	406	6	42128	0	12136	398	0	466	0	6
44	$(2,3)$	150	$2000 \cdot 6$	$15110 \cdot 6$	5.488	0.7491	534	134	65501	0	41722	480	0	602	14	120
44	$(3,3)$	150	$1633 \cdot 6$	$15110 \cdot 6$	4.192	0.4662	1287	687	147369	0	144504	564	0	765	175	512
65	$(2,2)$	150	$6500 \cdot 4$	$49094 \cdot 4$	3.638	0.6270	638	238	69222	0	55040	0	0	4140	3	235
65	$(2,3)$	150	$6500 \cdot 4$	$49094 \cdot 4$	5.709	0.3413	1172	772	144986	0	141480	0	0	4106	33	739
65	$(3,3)$	150	$5307 \cdot 4$	$49092 \cdot 4$	4.876	0.1048	5727	5127	682820	0	807408	0	0	6519	651	4476
87	$(2,2)$	150	$6500 \cdot 4$	$49094 \cdot 4$	2.206	0.6309	634	234	52219	0	47930	0	0	12020	5	229
87	$(2,3)$	150	$6500 \cdot 4$	$49094 \cdot 4$	3.285	0.3617	1106	706	115538	0	120484	0	0	11960	14	692
87	$(3,3)$	150	$5307 \cdot 4$	$49092 \cdot 4$	2.638	0.2813	2133	1533	198333	0	257235	0	0	14460	89	1444

Table 5: Pebble-ADA with [DKL+18] heuristic

ML-DSA-AD	(t,n)	k	σ_{pre}	σ_{ad}	M	p_{outer}	N_{start}	N_{rej}	N_{bern}	N_{ℓ_2}	N_{skip}	N_z	N_{γ_2}	N_ω	$N_{\text{cmb}}^{\text{out}}$	$N_{\perp}^{\text{out}}$
44	$(2,2)$	150	$2000 \cdot 6$	$15110 \cdot 6$	2.34	1.0000	400	0	34112	0	5606	378	0	434	0	0
44	$(2,3)$	150	$2000 \cdot 6$	$15110 \cdot 6$	3.32	0.9901	404	4	42275	0	11436	330	0	348	0	4
44	$(3,3)$	150	$1633 \cdot 6$	$15110 \cdot 6$	2.83	0.8889	675	75	65295	0	38169	540	0	669	13	62
65	$(2,2)$	150	$6500 \cdot 4$	$49094 \cdot 4$	2.795	0.8000	500	100	48050	0	32368	0	0	4402	0	100
65	$(2,3)$	150	$6500 \cdot 4$	$49094 \cdot 4$	4.292	0.5115	782	382	90072	0	79260	0	0	4186	3	379
65	$(3,3)$	150	$5307 \cdot 4$	$49092 \cdot 4$	3.528	0.2320	2586	1986	277905	0	334749	0	0	7083	154	1832
87	$(2,2)$	150	$6500 \cdot 4$	$49094 \cdot 4$	1.940	0.7605	526	126	38214	0	32704	0	0	11394	0	126
87	$(2,3)$	150	$6500 \cdot 4$	$49094 \cdot 4$	2.555	0.5714	700	300	63874	0	60004	0	0	10370	2	298
87	$(3,3)$	150	$5307 \cdot 4$	$49092 \cdot 4$	2.253	0.3759	1596	996	133017	0	172272	0	0	15870	33	963

Table 6: Pebble-ADA with [CdPE+26] heuristic

7 Evaluation

7.1 Implementation Details

All experiments were implemented in Rust and executed in a local Apple Silicon machine with an `Apple M2` CPU (8 logical cores) and `8 GB` RAM, running `macOS 15.6.1` (Darwin `24.6.0`, `arm64`).

Each signer is instantiated as an independent asynchronous task (one task per party) and all parties run on localhost within the same process space. Hence, (T, N) configurations are emulated by spawning N logical parties and activating T signers according to the protocol configuration.

Protocol communication is realized through an in-memory relay abstraction used by the test harness. In particular, each party is connected to a local message coordinator that provides round-based broadcast and message collection semantics. Messages are tagged by protocol round and attempt index.

Parties are executed concurrently using `tokio`-based async scheduling. The benchmarked signing time therefore captures protocol computation and local relay synchronization costs under concurrent execution on one host, rather than distributed WAN/LAN deployment costs. This setup is intended to isolate cryptographic/protocol overhead and message-volume effects from external networking effects. For each (T, N) setup and ML-DSA level, tests run : threshold signing T-MLDSA-Sign-AD, T-MLDSA-Verify-AD on the produced proof π and signature.

We are not benchmarking threshold ECSDA/EdDSA constructions, as the extra cost to compute the proof and verify is negligible. In more details the signer is doing a single extra hash, point multiplication and a modular addition for the computation of the proof and the tweak of the signature. The verifier as well performs a hash computation to compute the tweak, a point addition, and a point multiplication to verify the proof.

7.2 Discrete Gaussian Sampling

To instantiate Gaussian sampling in our implementation, we use a Karney [Kar16] discrete Gaussian sampler, rather than relying on floating-vector sampling followed by projection/rescaling. The main motivation is algorithmic alignment with the signing design in which nonce-related vectors are sampled directly from a discrete Gaussian over polynomial coefficients, with deterministic seed expansion. From an engineering perspective, the Karney-style approach provides a clean interface for transcript-bound sampling (via XOF-derived coins)

In Karney's exact discrete-Gaussian sampler, the algorithm first samples an auxiliary nonnegative integer $k \geq 0$ from a *right auxiliary distribution* proportional to $\exp(-k^2/2)$. This distribution is obtained via a two-stage construction based on Bernoulli trials with success probability $\exp(-1/2)$: the initial run length yields a geometric factor, while an additional acceptance correction using $k(k-1)$ Bernoulli trials adjusts the tail so that the resulting k has the required $\exp(-k^2/2)$ weight. Given k, the sampler then performs *randomized rounding* of the target real value $k\sigma$ to an integer. It chooses a random sign, anchors at $\lceil k\sigma \rceil$, and samples a uniform offset inside the rounding cell; proposals with invalid fractional position are rejected. Finally, the sampler applies an accept/reject correction with probability $\exp\big(-\frac{1}{2}x(2k+x)\big)$ to produce an integer $z \in \mathbb{Z}$ distributed close to the centered discrete Gaussian $D_{\mathbb{Z},0,\sigma}$.

7.3 Implementation performance

We benchmarked our Rust implementation of threshold DKG, DSG-AD signing, and Verify-AD on all configurations (ML-DSA-44/65/87 $\times$ $(t, n) \in \{(2,2), (2,3), (3,3)\}$). Each signing data point is the mean wall-clock time over three successful trials; DKG is measured once per configuration. Sign time includes end-to-end MPC execution, proof π assembly, and any outer rejection-sampling retries. Verify-AD time measures the joint check of signature $\sigma = (\tilde{c}, z, h)$ and proof $\pi = (w^{(pr)}, t_0)$.

Timing. DKG completes in under 5 ms in all configurations and is negligible compared to signing. Verify-AD is also lightweight: 0.58–1.24 ms across all setups, roughly three orders of magnitude faster than signing even in the fastest case. Signing dominates overall latency and exhibits strong

Level	(t, n)	DKG (ms)	Sign (ms)	Verify-AD (ms)	$\lvert\sigma\rvert$ (B)	$\lvert\pi\rvert$ (B)
ML-DSA-44	(2,2)	4.60	287.8	0.58	2420	4608
ML-DSA-65	(2,2)	3.31	464.2	0.83	3309	6912
ML-DSA-87	(2,2)	4.58	1493.6	1.24	4627	9216
ML-DSA-44	(2,3)	2.09	596.2	0.67	2420	4608
ML-DSA-65	(2,3)	2.19	2692.9	0.89	3309	6912
ML-DSA-87	(2,3)	2.78	1228.4	1.15	4627	9216
ML-DSA-44	(3,3)	1.76	2705.4	0.77	2420	4608
ML-DSA-65	(3,3)	3.54	7723.4	1.04	3309	6912
ML-DSA-87	(3,3)	2.26	4317.8	1.22	4627	9216

Table 7: DSG-AD and DKG wall-clock times (mean over 3 successful sign trials; DKG measured once per config). Sign time includes proof π assembly and outer rejection retries. Verify-AD checks signature $\sigma = (\tilde{c}, z, h)$ and proof $\pi = (w^{(pr)}, t_0)$.

dependence on both the security level and the threshold. For $(2, 2)$, mean sign time grows from 288 ms (ML-DSA-44) to 464 ms (65) and 1.49 s (87), reflecting the larger matrix dimensions and Gaussian sampling cost at higher levels. Increasing the number of online parties to $(2, 3)$ raises communication and combine work; ML-DSA-65 reaches 2.69 s, while 44 and 87 remain below 1.3 s in our run. The three-party online setting $(3, 3)$ is the hardest: ML-DSA-65 averages 7.72 s and ML-DSA-44 2.71 s, consistent with lower per-attempt success probability and hence more outer restarts, whereas ML-DSA-87 completes in 4.32 s. These numbers should be read as implementation-level figures (including rejection sampling), not as ideal single-attempt protocol cost.

Level	(t, n)	DKG	R1	R2	R3	R4	DSG	R1	R2	R3
ML-DSA-44	(2,2)	10.8	2.4	0.2	0.1	8.1	2738.1	0.1	1200.1	1537.9
ML-DSA-65	(2,2)	14.8	2.4	0.2	0.1	12.1	4031.8	0.1	1800.1	2231.6
ML-DSA-87	(2,2)	18.8	2.4	0.2	0.1	16.1	5438.1	0.1	2400.1	3037.9
ML-DSA-44	(2,3)	19.7	3.7	0.3	3.7	12.1	5476.1	0.3	2400.1	3075.7
ML-DSA-65	(2,3)	25.7	3.7	0.3	3.7	18.1	8063.6	0.3	3600.1	4463.2
ML-DSA-87	(2,3)	31.7	3.7	0.3	3.7	24.1	10876.1	0.3	4800.1	6075.7
ML-DSA-44	(3,3)	16.3	3.7	0.3	0.2	12.1	28749.7	1.4	12600.7	16147.6
ML-DSA-65	(3,3)	22.3	3.7	0.3	0.2	18.1	24190.9	0.8	10800.4	13389.7
ML-DSA-87	(3,3)	28.3	3.7	0.3	0.2	24.1	16314.2	0.4	7200.2	9113.6

Table 8: DSG-AD and DKG communication volume (outbound bytes summed over all parties; DSG-AD from last successful trial). Round columns are per-round totals in KB.

Communication. Protocol bandwidth is dominated by DSG-AD. Threshold key generation totals 11–32 KB outbound across all parties; round R4 (packed secret shares) carries most of the DKG volume. By contrast, a successful DSG-AD run transfers 2.7–28.8 MB, depending on (t, n) and the parameter set. The bulk lies in signing rounds R2 and R3, which broadcast material for $k = 50$ parallel repetitions per outer attempt (commitment openings, $\mathbf{w}_j$ lists, and combined $(\mathbf{z}, w^{(pr)})$ payloads). Round R3 is consistently the largest contributor, as it carries the packed $\mathbf{z}$ vectors together with $w^{(pr)}$ for every parallel iteration. Communication therefore scales approximately linearly in k and in the ML-DSA dimension (via $\mathbf{z}$ and $w^{(pr)}$ sizes), which explains the jump

from $\approx 2.7\,\mathrm{MB}$ at $(2,2)/44$ to $\approx 28.8\,\mathrm{MB}$ at $(3,3)/44$ even though only two parties sign online. At fixed (t,n), higher security levels increase DSG volume in proportion to signature and proof size ($|\sigma| \in \{2420, 3309, 4627\}\,\mathrm{B}$ and $|\pi| \in \{4608, 6912, 9216\}\,\mathrm{B}$ for levels 44/65/87).

Verify-AD adds minimal overhead once (σ, π) are available. Reducing k or improving rejection parameters would lower communication substantially at the cost of success probability per outer attempt; our measurements isolate this trade-off and motivate parameter tuning per security level and threshold.

References

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

[Bat08] Necdet Batir. Inequalities for the gamma function. *Archiv der Mathematik*, 91(6), 2008. 3.2

[BdCE+25] Alexander Bienstock, Leo de Castro, Daniel Escudero, Antigoni Polychroniadou, and Akira Takahashi. Quorus: Efficient, scalable threshold ML-DSA signatures from MPC. Cryptology ePrint Archive, Paper 2025/1163, 2025. 1

[BKL+25] Cecilia Boschini, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Akira Takahashi, and Mehdi Tibouchi. Ringtail: Practical two-round threshold signatures from learning with errors. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *2025 IEEE Symposium on Security and Privacy*, pages 149–164. IEEE Computer Society Press, May 2025. 1

[BL25] Katharina Boudgoust and Oleksandra Lapiha. Leftover hash lemma(s) over cyclotomic rings. Cryptology ePrint Archive, Report 2025/1080, 2025. 1.1, 2, 3.2

[CdPE+26] Sofía Celi, Rafaël del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. Efficient threshold ML-DSA. Cryptology ePrint Archive, Paper 2026/013, 2026. 1, 1, 1, 1, 8, 4, 4, 22, 27, 6.1, 2b, 3, 6.2, 4b, 6

[DFPS22] Julien Devevey, Omar Fawzi, Alain Passelègue, and Damien Stehlé. On rejection sampling in Lyubashevsky's signature scheme. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 34–64. Springer, Cham, December 2022. 1, 7, 16

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018. 6.1, 2a, 4a, 5

[DKM+24] Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248. Springer, Cham, May 2024. 1, 1

[DLL+17] Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS – dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Paper 2017/633, 2017. Version 20170627:201152; https://eprint.iacr.org/archive/2017/633/20170627:201152. 1, 1.1, 2.5, 2.5, 2.5

[DOTT22] Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. *Journal of Cryptology*, 35(2):14, April 2022. 1, 4

[dPKPR24] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. Raccoon: A masking-friendly signature proven in the probing model. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 409–444. Springer, Cham, August 2024. 1

[dPN25] Rafael del Pino and Guilhem Niot. Finally! a compact lattice-based threshold signature. In Tibor Jager and Jiaxin Pan, editors, *Public-Key Cryptography – PKC 2025*, pages 169–199, Cham, 2025. Springer Nature Switzerland. 1

[EKT24] Thomas Espitau, Shuichi Katsumata, and Kaoru Takemure. Two-round threshold signature from algebraic one-more learning with errors. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 387–424. Springer, Cham, August 2024. 1

[GKS24] Kamil Doruk Gür, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice-based signatures from threshold homomorphic encryption. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*, pages 266–300. Springer, Cham, June 2024. 1

[JLWG25] Haoxiang Jin, Feng-Hao Liu, Zhedong Wang, and Dawu Gu. Discrete gaussians modulo sub-lattices: New leftover hash lemmas for discrete gaussians. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part II*, volume 15675 of *LNCS*, pages 301–330. Springer, Cham, May 2025. 1.1

[Kar16] Charles F. F. Karney. Sampling exactly from the normal distribution. *ACM Transactions on Mathematical Software (TOMS)*, 42(1):3:1–3:14, 2016. 7.2

[KLLS26] Yashvanth Kondi, Oleksandra Lapiha, Iraklis Leontiadis, and Tanushri Sen. Threshold signing with associated data authentication, 2026. Manuscript. 1, 2.4, 3.1

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Cham, April / May 2018. 9, 15, 17

[KLSS23] Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580. Springer, Cham, August 2023. 1.1, 3

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013. 1.1, 3

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009. 1

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Berlin, Heidelberg, April 2012. 6, 8, 2.5

[NIS24] NIST. Module-lattice-based digital signature standard. Federal Information Processing Standards Publication NIST FIPS 204, U.S. Department of Commerce, Gaithersburg, MD, 2024. 2.5, 2.5, 2.5, 1

[Pei08] Chris Peikert. Limits on the hardness of lattice problems in lp norms. *computational complexity*, 17(2):300–351, 2008. 5

[Pei10] Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Berlin, Heidelberg, August 2010. 4

[PN25] Rafaël Del Pino and Guilhem Niot. Finally! A compact lattice-based threshold signature. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part III*, volume 15676 of *LNCS*, pages 169–199. Springer, Cham, May 2025. 2