Statistical Inference from Noisy Randomness Leakage for ML-DSA Attacks

Niklas Paskarbeit [1], Carsten Schubert [2], Marian Margraf [1], and Jean-Pierre Seifert [2]

[1] Freie Universität Berlin, Berlin, Germany
niklas.mueller@fu-berlin.de, marian.margraf@fu-berlin.de
[2] Technische Universität Berlin, Berlin, Germany
carsten.gm.schubert@tu-berlin.de, jean-pierre.seifert@tu-berlin.de

Abstract. ML-DSA is a NIST post-quantum signature standard whose security argument rests on rejection sampling making released signatures independent of the secret key. Liu *et al.* [1] and Damm *et al.* [2] showed that a single leaked bit of the masking randomness per signature breaks this guarantee, making ML-DSA subkeys recoverable from a number of so-called informative relations, and the resulting attacks were sharpened by Schubert *et al.* [3] and Bashiri *et al.* [4]. All of them treat every leaked bit as equally trustworthy.

We show that they need not. After the j-independence transformation introduced by Damm *et al.* [2], the values extracted from a clean and from a flipped bit are exact complements in absolute value, so the observable relation $|\tilde{z}|$ follows a two-component mixture whose components we determine in closed form for all leakage regimes. This yields, first, a method-of-moments estimator for the bit-error rate p that requires only the informative relations an attack already collects and that, unlike the estimator of Schubert *et al.* [3], also works in the low-leakage regime, down to leakage index 4 for ML-DSA-44 and ML-DSA-87 and 5 for ML-DSA-65, the same indices at which the attack itself remains feasible. However lower leakage indices come with a lower accuracy, at 25,000 informative relations the mean absolute error for the lowest cases is about 0.022 to 0.040 compared to values below 0.002 in the high-leakage regime. The noise estimator is also adjustable to estimate asymmetric noise rates. Second, it yields a posterior probability, for each individual relation, that its leaked bit is correct. Thresholding this posterior gives an $O(\alpha)$ preprocessing step that corrects relations classified as noisy and that any downstream attack can use unchanged, where α is the amount of informative relations.

Applied to the attack of Schubert *et al.* [3], the preprocessing reduces the number of informative relations required for key recovery by about 20% to 44% across all three parameter sets, leakage indices 6 to 9, and error rates 20% and 40%. Applied to the attack of Bashiri *et al.* [4] for ML-DSA-44 at leakage index 8, it improves key recovery across the noise range we tested: at $p = 0.45$ from 1,900,000 informative relations, at $p = 0.40$ from 450,000 informative relations, at $p = 0.20$ from 40,000 informative relations, a single fixed threshold chosen in hindsight succeeds

for 20, 22, and 23 of 30 seeds against 16, 18, and 19 of 30 without the preprocessing, and a parallel search over ten thresholds succeeds for 23, 25, and 29 of 30.

Keywords: Lattice-based Cryptography · Post-Quantum Cryptography · Side-Channel Attacks · Randomness Leakage Attacks · ML-DSA · CRYSTALS-Dilithium

1 Introduction

ML-DSA[3] [5], standardized by NIST as the primary post-quantum signature scheme, derives its security from the hardness of Module-LWE and from rejection sampling in the Fiat–Shamir-with-aborts paradigm [6, 7]: every released signature is statistically independent of the secret key, provided the masking randomness y stays hidden. Liu *et al.* [1] showed that this condition is fragile: a single leaked bit of the masking randomness y per signature already suffices in principle for key recovery. Damm *et al.* [2] turned the observation into a practical regression attack covering all three parameter sets. In a major revision of their work [8], they extended the analysis to *noisy* leakage, where the leaked bit is flipped independently with probability p.

The cost of these attacks has since fallen sharply. Schubert *et al.* [3] replaced the regression step by a multi-tier hill-climbing search over candidate subkeys, and Bashiri *et al.* [4] built a stochastic model of the leakage process, derived bounds on the number of informative relations needed for the key to be determined, and attained them with integer- and nonlinear-programming attacks. Between those works and Damm *et al.* [2], the required number of leaky signatures has dropped by one to two orders of magnitude, and key recovery is now reported at error rates approaching $p = \frac{1}{2}$. The attacks work for leaked bit indices from $j = 4$ for ML-DSA-44 and ML-DSA-87 and $j = 5$ for ML-DSA-65.

What none of these attacks do is distinguish between the leaked bits they are given. Every informative relation enters the objective function with the same weight, whether its bit is likely correct or likely flipped. Yet the relations do carry that information: an observable $\overline{z}$ extracted with a flipped bit is displaced by exactly 2^j ([8], Theorem 9). After the j-independence transformation transforms $\overline{z}$ into $\tilde{z}$, the absolutes of the clean and flipped values of $\tilde{z}$ always add up to the same number. $|\tilde{z}|$ is therefore drawn from a two-component mixture whose components are known in closed form, and each individual relation can be assigned a posterior probability of carrying a correct bit. Exploiting this is the subject of this work.

A second gap is that noise estimation has so far been available only in the high-leakage regime, via the identifiably corrupted relations that the extraction discards ([3], Section 5.2); in the low-leakage regime no relation is discarded and the mechanism is unavailable. Bashiri *et al.* [4] assume p known when sizing their attacks. An attacker facing a real side channel usually knows neither.

Our contributions.

[3] Formerly CRYSTALS-Dilithium.

– We determine the exact distribution of the transformed value $\tilde{z}^{(c)}$ in both leakage regimes, and give $\mathbb{E}[|\tilde{z}^{(c)}|]$ in closed form (Theorems 2 and 3) as a function of the public parameters and the leakage index alone. We verify both against simulation.
– From the complement identity we derive a method-of-moments estimator for the noise rate p (Theorem 1) that is unbiased, consistent up to a key-dependent bias, and has explicit variance. Unlike the estimator of [3] it uses only the informative relations the attack already retains, and it applies in the low-leakage regime, down to $j = 4$ for ML-DSA-44 and ML-DSA-87 and $j = 5$ for ML-DSA-65. We also derive a variant for the estimator to recover not only an overall noise rate but also individual noise rates for bits that are initially 0 and bits that are initially 1.
– We refine the estimator into a *per-relation* posterior probability that a given leaked bit is correct in Theorem 4, and turn it into a preprocessing step that corrects relations classified as noisy (Algorithm 1) and takes linear time in the amount of informative relations. The preprocessing is attack-agnostic: it consumes and produces informative relations, so any downstream attack can use it unchanged.
– We evaluate the preprocessing in a limited empirical setting to show its practical relevance. In combination with the attack of Schubert *et al.* [3], it reduces the number of required informative relations by 20% to 44% across our chosen parameter sets, leakage indices and noise rates. In a different format on the attack of Bashiri *et al.* [4] in the setting of their Remark 67, expanded by two additional possible noise rates, we managed to demonstrate that here also a recovery is possible when the attack alone does not suffice. We showed recovery rates increased by 21% to 25% for a single probability-threshold chosen in hindsight and an increase of 38.9% to 52.6% for a sweep over ten different probability-thresholds.

1.1 Outline

Section 1.2 fixes notation. Section 2 reviews the ML-DSA scheme and the leakage-based attacks of Liu *et al.* [1], Damm *et al.* [2], Schubert *et al.* [3] and Bashiri *et al.* [4]. Section 3 develops the noise estimator that works for all leakage indices. Section 4 evaluates the individual probabilities of leaked bits to be false. These probabilities are then used to build a preprocessing algorithm, which alters leaked bits based on the probability of them being wrong. The respective threshold is given as an input parameter to the algorithm. Section 5 reports experimental results for the noise estimator as well as for the attack scheme of Schubert *et al.* [3] with our preprocessing and suggests an option for an enhanced attack combining a regular attack of Bashiri *et al.* [4] and the presented preprocessing. Finally Section 6 discusses implications and open directions.

1.2 Preliminaries

We denote matrices with uppercase bold letter (e.g. $\mathbf{A}$) and vectors with lowercase bold letters (e.g. $\mathbf{x}$).

For sampling we denote by $x \xleftarrow{\$} D$ the sampling according to a distribution D. When we write a finite set S in place of a specific distribution we sample from the uniform distribution on that set.

By $[x]$ we denote the set $\{-x, \ldots, x\}$. By $R_{|S}$ we denote all ring elements of R where each coefficient lies within the set S.

For an integer a and a positive modulus m, the centered reduction $[a]_m$ denotes the unique representative of $a \bmod m$ in the half-open interval $[-m/2,\ m/2)$. Furthermore, by $|\cdot|_M$ we denote the centered reduction modulo M followed by yielding the absolute value.

Throughout, we work with extracted informative relations $(c, \tilde{z})$ in the sense of [3], and write $\delta(x^*) := [\langle c, x^* \rangle - \tilde{z}]_M$ for the centered residual of a candidate x^*, where the centered reduction modulo $M = 2^{j+1}$ is applied in the low-leakage regime ($2^{j-1} \leq \beta$) and the residual is taken over $\mathbb{Z}$ in the high-leakage regime ($2^{j-1} > \beta$), as in Algorithm 1 of [3].

We set $\sum_{i=a}^{b} f(i) = 0$ if $b < a$.

2 Attack Setting

Our preprocessing aims to improve the attacks of Schubert *et al.* [3] and Bashiri *et al.* [4] that are based on the attacks of Liu *et al.* [1] and Damm *et al.* [2]. We therefore use the same setting and will present this setting in this section.

The goal of all these attacks is to recover a (partial) secret ML-DSA key using leakage information.

2.1 ML-DSA

In this subsection we follow the explanations of Schubert *et al.* [3] for ML-DSA.

ML-DSA is a Modular-Lattice-based Digital Signature Algorithm following the Fiat-Shamir-with-aborts paradigm that was introduced by Lyubashevsky [6, 7]. The Fiat-Shamir transform [9] is combined with rejection sampling to ensure that no information about the secret key is revealed by any released signature. The security relies thereby on the hardness of the Module Learning with Errors (Module-LWE) problem [7]. The scheme was proposed by Ducas *et al.* [10] and subsequently standardized by the NIST [5] as part of their "PQC Standardization process". For a detailed description of the algorithms we refer to the standard [5]. In this chapter we focus on the most relevant parts to the attacks we want to improve.

Generally, we note that all algebraic operations take place in the polynomial ring $R \coloneqq \mathbb{Z}_q[x]/(x^n + 1)$. Elements of R can alternatively be viewed as n-dimensional integer vectors where each entry is the corresponding coefficient of the polynomial. During this work we will use that view.

Key generation. The signer samples a public matrix $\mathbf{A} \xleftarrow{\$} R^{k \times \ell}$ and two secret vectors $\mathbf{s}_1 \xleftarrow{\$} R^{\ell}_{|[\eta]}$, $\mathbf{s}_2 \xleftarrow{\$} R^{k}_{|[\eta]}$. $(\mathbf{A},\ \mathbf{t} \coloneqq \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$ is the public key.

Signing. In order to sign a message, the signer:

1. samples masking randomness $\mathbf{y} \xleftarrow{\$} R^{\ell}_{|[\gamma_1 - 1]}$,
2. calculates[4] a challenge polynomial $\mathbf{c} \in R$ with exactly τ coefficients equal to ± 1 and all other coefficients zero,
3. computes the signature vector $\mathbf{z} \coloneqq \mathbf{y} + \mathbf{c}\mathbf{s}_1$.
4. if $\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta$ release the signature $(\mathbf{z}, \mathbf{h}[5], \mathbf{c})$ else go back to step 1.

Role of rejection sampling. The rejection condition ensures that every coefficient z of the released signature vector $\mathbf{z}$ is uniformly distributed on $[\gamma_1 - \beta]$. Since this is independent of the secret key, no information about the secret key is leaked.[6]

We note that the attack operates at the level of individual polynomial coefficients and therefore the k-th coefficient of the polynomial product $\mathbf{c} \cdot \mathbf{x}$ in R can be written as an inner product $\langle \mathbf{c}^{(k)}, \mathbf{x} \rangle$, where $\mathbf{c}^{(k)} \in \{-1, 0, 1\}^n$ is the k-th row of the negacyclic matrix associated with $\mathbf{c}$ with exactly τ entries that are nonzero. Since we will only attack a partial key we will, for simplicity, write $\mathbf{c}$ for the relevant row vector and $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$ for a single coefficient of the signature vector following [2, 3].

Parameter sets. NIST lists three ML-DSA parameter sets in their published standard [5]. We list the for this work relevant parameters in Table 1.

Parameter	**ML-DSA-44**	**ML-DSA-65**	**ML-DSA-87**
n (ring dimension)	256	256	256
q (modulus)	8380417	8380417	8380417
(k, ℓ) (dimensions of $\mathbf{A}$)	$(4, 4)$	$(6, 5)$	$(8, 7)$
η (secret key position bound)	2	4	2
τ (# of ± 1's in $\mathbf{c}$)	39	49	60
$\beta = \tau \cdot \eta$ (norm bound)	78	196	120

Table 1: Excerpt from the ML-DSA parameter sets [5].

2.2 Attack ideas of Liu *et al.* [1] and Damm *et al.* [8]

While a signature leaks no information during an unblemished signing process, Liu *et al.* [1] discovered that with side-channel information leakage can be reintroduced and exploited by an attacker. In this subsection we show the ideas of Liu *et al.* [1] and Damm *et al.* [8]. The latter is a direct improvement and simplification of the former. Since for this work we are interested in the case where we only get a noisy

[4] Using $\mathbf{y}, \mathbf{A}$ as well as the message.
[5] $\mathbf{h}$ is a hint that allows the verifier to verify that the correct c was used.
[6] FIPS 204 accepts signatures only with coefficients within the open interval $(-\gamma_1 + \beta, \gamma_1 - \beta)$. We follow [2, 3] and use the closed interval $[\gamma_1 - \beta]$ throughout, due to the difference of a single integer value being negligible.

side-channel[7], we will focus on formulations including that. For readers interested in the full details of the prior attacks we refer to their respective papers.

Firstly we introduce the mathematical problem of Noisy Leaky-Signature-LWE.

Definition 1 (Noisy Leaky-Signature-LWE, based on [3] Definition 3). *Let* $(\mathbf{A}, \mathbf{t}) \in R^{k\times\ell} \times R^k$ *be an ML-DSA public key with corresponding secret key* $(\mathbf{s}_1, \mathbf{s}_2) \in R^{\ell}_{|[\eta]} \times R^{k}_{|[\eta]}$. *Let* $\mathbf{x} \in \{-\eta, \dots, \eta\}^n$ *be the coefficient vector of one of the* ℓ *polynomials in* $\mathbf{s_1}$. *Let* y *be the coefficient of the randomness corresponding to* x, *while* $y_j^{(c)}$ *denotes the* j*-th bit of* y. *Let* $p \in [0, \frac{1}{2})$ *be an error rate and* $y_j = y_j^{(c)} \oplus \zeta$ *with* $\zeta \sim \mathrm{Ber}(p)$. *Given* $(\mathbf{A}, \mathbf{t})$ *and arbitrarily many tuples* $(\mathbf{c}, z, y_j)$, *the attacker's goal is then to find* $\mathbf{x}$.

While we use $y_j^{(c)}$ to denote the correct bit, we denote the false bit as $y_j^{(f)} \coloneqq y_j^{(c)} \oplus 1$.

Remark 1. The distribution of z is uniform on $[\gamma_1 - \beta]$ due to the rejection sampling. As noted by Bashiri *et al.* [4] the distribution of $S \coloneqq \langle \mathbf{c}, \mathbf{x} \rangle$ lies in $[\beta]$ and is the sum of τ random variables that are uniformly distributed on $[\eta]$.

Definition 2 (Leakage Regimes). *We call a leakage index* j *part of the low-leakage regime if* $2^{j-1} \leq \beta$ *and part of the high-leakage regime if* $2^{j-1} > \beta$. *And we define* $\beta_{\text{eff}} \coloneqq \min(2^{j-1}, \beta)$.

Liu *et al.* [1] introduced the core mechanic of these attacks. The leaked bit y_j is used to calculate

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}$$

where $[y]_{2^j}$ is a random value distributed on $[2^{j-1}]$. While looking similar to the released signature the main difference is the missing rejection sampling. It is to be noted that in the low-leakage regime this equation holds only with a centered reduction $\mod M$ where $M = 4\beta_{\text{eff}}$. Liu *et al.* [1] showed that least square linear regression can be used to find the secret key.

Damm *et al.* [2] noted that, while in the low-leakage regime all such relations hold information, in the high-leakage regime only relations that fulfill $2^{j-1} + \beta \geq |\overline{z}| \geq 2^{j-1} - \beta$ hold information.

Definition 3 (Informative Relations, based on [3] Definition 4). *In the low-leakage regime all relations are* informative *relations.*

In the high-leakage regime a relation is called informative *if* $2^{j-1} + \beta \geq |\overline{z}| \geq 2^{j-1} - \beta$.

In order to only use the informative relations Damm *et al.* [2] propose a "j-independence transformation" for the high-leakage regime: All non-informative relations are discarded. All informative relations are transformed via:

$$\tilde{z} = \begin{cases} \overline{z} - 2^{j-1} + \beta, & \text{if } \overline{z} \geq 2^{j-1} - \beta, \\ \overline{z} + 2^{j-1} - \beta & \text{if } \overline{z} \leq -(2^{j-1} - \beta) \end{cases} \tag{1}$$

[7] Every leaked bit from the side-channel is only true with probability $1 - p$.

This transformation leads to $\tilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \tilde{y}$ where $\tilde{y} \xleftarrow{\$} [\beta]$. In order to unify notation: for the low-leakage case we define $\tilde{z} = \overline{z}$. Then we receive $\tilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \tilde{y}$ where $\tilde{y} \xleftarrow{\$} [\beta_{\text{eff}}]$ independent of the leakage regime.

Damm *et al.* [2] then use a linear regression as well to recover the secret key. In their major revision [8] they extend this attack to the noisy case.

2.3 Attack of Schubert *et al.* [3]

The attack of Schubert *et al.* [3] uses the same underlying mechanics and just replaces the linear regression with a hill-climbing algorithm.

In order to create their hill-climbing algorithm they need to create some kind of measure to evaluate a key candidate. They note that each informative relation can be used to evaluate a subkey $\mathbf{x}^*$ by calculating

$$\delta \coloneqq \langle \mathbf{c}, \mathbf{x}^* \rangle - \tilde{z}$$

and checking if $\delta \in [\beta_{\text{eff}}]$. For the true key, this always evaluates as true, while other keys have a chance of evaluating as false.
They start with a key candidate provided by linear regression. Since they use far fewer informative relations, the initial key candidate is still far away from the true key. They then evaluate if changes to a few positions would reduce the measure of their key candidate and replace their candidate with the better one in that case. This is repeated until no improvement is found for a set number of iterations. With this base approach and a number of further optimizations they manage to reduce the number of needed informative relations by a factor of up to 68 and also reduce the lowest possible leakage position to 4 for ML-DSA-44 and ML-DSA-87 and 5 for ML-DSA-65.

Unrelated to the attack Schubert *et al.* [3] note that in the high-leakage regime relations with $|\overline{z}| > 2^{j-1} + \beta$ are identifiably noise-corrupted, while relations with $|\overline{z}| < 2^{j-1} - \beta$ are identifiable correct. Both are non-informative and therefore not usable for key-recovery. However Schubert *et al.* [3] note that they still can be used to estimate the noise rate in the high-leakage case via

$$\hat{p} \coloneqq \frac{[\text{Number of identifiable noise corrupted relations}]}{[\text{Number of non-informative relations}]}.$$

2.4 Attack of Bashiri *et al.* [4]

Concurrently with Schubert *et al.* [3], Bashiri *et al.* [4] approach the same underlying problem from a more foundational, information-theoretic and probability-theoretic angle. Rather than starting at a concrete algorithm, they first build a stochastic model of the leakage process and derive from it both a rigorous characterization of an optimal attacker and concrete bounds on the number of informative relations required for key recovery, before turning to practical attacks. We adopt their results below, but write j for their leakage index λ and y_j for their observed (possibly noisy) bit b, to match the notation introduced above. (See Definition 1.)

Definition 4 (Candidate and Most Likely Keys). *Given informative relations* $(\mathbf{c}^{(i)}, z^{(i)}, y_j^{(i)})_{i=1}^{n_{\inf}}$*, the set of* candidate keys X_{cand} *consists of all* $\mathbf{x}^* \in X$ *consistent with every relation, and the set of* most likely keys X_{ml} *consists of those* $\mathbf{x}^* \in X$ *minimizing the number of violated relations.*

Bashiri *et al.* [4] show that returning an element of X_{ml} is an optimal decision rule in the sense of maximizing the attacker's overall success probability. This provides a model-based justification for the same count-based scoring principle that Schubert *et al.* [3] arrive at constructively via their monotonicity result (Lemma 1 and Corollary 1 there).

Definition 5 (Per-Relation Mutual Information). *Let* $S \coloneqq \langle \mathbf{c}, \mathbf{x} \rangle$ *and, for an observed relation* $(\mathbf{c}, z, y_j)$*, define*

$$p_{z,y_j} \coloneqq \Pr\big[y_j = \mathrm{bit}_j(z - S)\big], \qquad q_{z,y_j} \coloneqq (1-p)\, p_{z,y_j} + p \cdot p_{z,1-y_j},$$

where $\mathrm{bit}_j(\cdot)$ *extracts the* j*-th bit (two's complement) and* p *is the bit-error probability from Definition 1. Bashiri* et al. *[4] define the mutual information that the relation carries about the key as*

$$I(z, y_j) \coloneqq -\log_2(q_{z,y_j}) + (1-p)\log_2(1-p)\,\frac{p_{z,y_j}}{q_{z,y_j}} + p\log_2(p)\,\frac{p_{z,1-y_j}}{q_{z,y_j}}.$$

$I(z, y_j) > 0$ exactly when $z \bmod^{\pm} 2^j \in [-\beta, \beta)$, recovering the notion of an informative relation. Unlike the binary informative/non-informative split above, however, $I(z, y_j)$ varies continuously with z: relations near the center of the informative window carry strictly more information than those near its edge.

Building on $I(z, y_j)$, Bashiri *et al.* [4] derive closed-form lower and upper bounds on the expected number of candidate keys as a function of the number of informative relations $n_{\inf}$, using only the second and fourth moments of key-difference vectors, and from these extract concrete thresholds $n_{\inf,\text{lower}}$ and $n_{\inf,\text{upper}}$ at which a unique key becomes determined. E.g., for ML-DSA-44 for $p = 0$ in the high-leakage regime, $n_{\inf,\text{lower}} = 8\,774$ and $n_{\inf,\text{upper}} = 12\,841$. These serve as a theoretical yardstick against which concrete attacks, including that of Schubert *et al.* [3], can be measured for near-optimality.

ILP and NLP attacks. For exact leakage, Bashiri *et al.* [4] formulate key recovery as an integer feasibility problem over the 256 secret coefficients, solvable with Gurobi's feasibility-pump heuristic. For noisy leakage they add per-relation slack variables and minimize their sum, though this variant proves intractable for that solver even when given the correct solution as a hint. To handle noise in practice they instead relax the same problem to a continuous, differentiable objective, optimized via Limited-Memory-Broyden–Fletcher–Goldfarb–Shanno algorithm, followed by a combinatorial best-first search that refines the rounded continuous solution. This nonlinear-programming (NLP) attack is reported to improve on the previous attacks of Damm *et al.* [8] and Schubert *et al.* [3] and to tolerate higher bit-error probabilities. However the extent of this improvement is unclear as Bashiri *et al.* [4] note that improvements in the code of Schubert *et al.* [3] might lead to a narrower gap between the two approaches.

Remark 2 (Noise-rate estimation). Bashiri *et al.* [4] assume p is known in order to calculate the number of needed informative relations. They note that the amount of needed informative relations is roughly proportional to $1/\left(1-2\sqrt{p(1-p)}\right)$.

They also do not use their calculated information value of relations for their attack besides removing non-informative relations.

2.5 Further related work

While the previous mentioned related works are directly expanded in this paper, noisy side-channel information has been treated in other cryptanalytic settings. Dachman-Soled *et al.* [11] incorporate noisy *hints* obtained from side channels into lattice-reduction attacks on LWE-based schemes more broadly. Hermelink *et al.* [12] generalize such hints into *distribution hints* and build a greedy solver based on these. Other side-channel vectors against ML-DSA specifically, such as the signature-correction attack of Islam *et al.* [13], target different intermediate values than the masking-randomness bit leakage studied here.

3 Noise Structure and Estimator

Schubert *et al.* [3] observed that in the high-leakage regime we can estimate the noise rate by comparing the non-informative relations. In this section we present a way to use the informative relations in the low-leakage regime to estimate the noise in that case as well. The fundamental structural observation is that the extracted values $\tilde{z}^{(c)}$ (clean) and $\tilde{z}^{(f)}$ (noise-flipped) are *complements* in absolute value after centered reduction modulo M. This holds regardless of the secret key.

Proposition 1 (Complement identity). *Let $(c, \overline{z}^{(c)})$ be any informative relation extracted with the correct bit, and let $\overline{z}^{(f)}$ be the value extracted from the same signature with the flipped bit. We have the point-wise identity*

$$|\overline{z}^{(c)}| + |\overline{z}^{(f)}| \;=\; 2^j. \tag{2}$$

Proof. Theorem 9 of [8] gives, over $\mathbb{Z}$,

$$\overline{z}^{(f)} = \overline{z}^{(c)} - \operatorname{sign}(\overline{z}^{(c)}) \cdot 2^j.$$

Since $\overline{z}^{(c)} \in (-2^j, 2^j]$, we have $|\overline{z}^{(c)}| \le 2^j$.
If $\overline{z}^{(c)} > 0$: $\overline{z}^{(f)} = \overline{z}^{(c)} - 2^j < 0$, so $|\overline{z}^{(f)}| = 2^j - \overline{z}^{(c)} = 2^j - |\overline{z}^{(c)}|$.
If $\overline{z}^{(c)} < 0$: $\overline{z}^{(f)} = \overline{z}^{(c)} + 2^j > 0$, so $|\overline{z}^{(f)}| = 2^j + \overline{z}^{(c)} = 2^j - |\overline{z}^{(c)}|$.
In both cases (2) holds. For $\overline{z}^{(c)} = 0$ the convention $\operatorname{sign}(0) = +1$ gives $\overline{z}^{(f)} = -2^j$ and the identity holds as well. □

As Damm *et al.* [2] we want to focus on the informative relations only and therefore look at the consequences for $\tilde{z}$. In the low-leakage regime we already have $\tilde{z} = \overline{z}$. We next take a look at the high-leakage regime.

Corollary 1. *In the high-leakage regime it holds that:*

$$|\tilde{z}^{(c)}| + |\tilde{z}^{(f)}| \;=\; 2\beta.$$

Proof. The proof is analogous to the proof of Proposition 1 by using Corollary 2 of [3] which states:

$$\tilde{z}^{(f)} = \tilde{z}^{(c)} - \mathrm{sign}(\tilde{z}^{(c)}) \cdot 2\beta.$$

□

Corollary 2. *By using the notation of* $\beta_{\text{eff}} = \min\left(\beta,\ 2^{j-1}\right)$ *we can unify Proposition 1 and Corollary 1 to:*

$$|\tilde{z}^{(c)}| + |\tilde{z}^{(f)}| \;=\; 2\beta_{\text{eff}}. \tag{3}$$

This proposition holds for every individual relation and requires no knowledge of the secret key x, the challenge vector c, or the randomness y. It is a purely algebraic consequence of the structure of the bit-flip perturbation.

The complement identity (3) immediately yields a method-of-moments estimator for the noise rate p from the observable values $|\tilde{z}|$ alone.

We define $\mu_c := \mathbb{E}[|\tilde{z}^{(c)}|]$, where the expectation is over the joint distribution of the challenge $\mathbf{c}$, the randomness y, and the true key $\mathbf{x}$ (uniform on X). By Corollary 2, the flipped mean satisfies $\mu_f := \mathbb{E}[|\tilde{z}^{(f)}|] = 2\beta_{\text{eff}} - \mu_c$.

The observable mean under noise rate p is therefore

$$\mu(p) := \mathbb{E}[|\tilde{z}|] = (1-p)\,\mu_c + p\,(2\beta_{\text{eff}} - \mu_c) = \mu_c + p\,(2\beta_{\text{eff}} - 2\mu_c). \tag{4}$$

Inverting (4) yields the estimator.

Theorem 1 (Method-of-moments noise-rate estimator). *Let* $\tilde{z}_1, \ldots, \tilde{z}_\alpha$ *be informative extracted values drawn under noise rate* $p \in [0, \frac{1}{2})$*; conditioned on* $\mathbf{x}$ *they are i.i.d. Write* $\mu_c(\mathbf{x}) := \mathbb{E}\big[|\tilde{z}^{(c)}| \,\big|\, \mathbf{x}\big]$*, so that* $\mu_c = \mathbb{E}_{\mathbf{x}}[\mu_c(\mathbf{x})]$*. Define the empirical mean*

$$\hat{\mu} \;:=\; \frac{1}{\alpha}\sum_{t=1}^{\alpha} |\tilde{z}_t|,$$

and $\Delta := 2\beta_{\text{eff}} - 2\mu_c = \mu_f - \mu_c$*. If* $\Delta > 0$*, then the method-of-moments estimator*

$$\hat{p} \;:=\; \frac{\hat{\mu} - \mu_c}{\Delta}. \tag{5}$$

fulfills:

1. (Unbiased) $\mathbb{E}[\hat{p}] = p$ *for every* $p \in [0, \frac{1}{2})$*, where the expectation is taken over the uniform distribution of possible secret keys as well.*
2. (Consistent up to key-dependence) *Fix a secret subkey* $\mathbf{x}$*. Conditioned on that particular true key* $\mathbf{x}$*,*

$$\hat{p} \;\xrightarrow{a.s.}\; p + \frac{\mu_c(\mathbf{x}) - \mu_c}{\Delta} \qquad \text{as } \alpha \to \infty.$$

3. (Variance) $\mathrm{Var}(\hat{p} \mid \mathbf{x}) = \dfrac{\mathrm{Var}(|\tilde{z}^{(c)}| \mid \mathbf{x}) + p(1-p)\Delta^2}{\alpha\,\Delta^2}$.

Proof. Let $W_t := |\tilde{z}_t|$. Conditioned on $\mathbf{x}$, each W_t follows the same mixture distribution: with probability $1-p$ it equals $|\tilde{z}_t^{(c)}|$ (clean), and with probability p it equals $|\tilde{z}_t^{(f)}| = 2\beta_{\text{eff}} - |\tilde{z}_t^{(c)}|$ (flipped, by Corollary 2). Since the $\tilde{z}_t$ are i.i.d. given $\mathbf{x}$, so are the W_t.

(1) Unbiasedness. The mean of the mixture distribution is

$$\begin{aligned}\mathbb{E}[W_t] &= (1-p)\,\mathbb{E}[|\tilde{z}^{(c)}|] + p\,\mathbb{E}[|\tilde{z}^{(f)}|] \\ &= (1-p)\,\mu_c + p\,(2\beta_{\text{eff}} - \mu_c) \qquad \text{(Corollary 2)} \\ &= \mu_c + p\,\Delta. \end{aligned} \tag{6}$$

By linearity of expectation,

$$\mathbb{E}[\hat{\mu}] = \frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbb{E}[W_t] = \mu_c + p\,\Delta.$$

Substituting into (5),

$$\mathbb{E}[\hat{p}] = \frac{\mathbb{E}[\hat{\mu}] - \mu_c}{\Delta} = \frac{(\mu_c + p\,\Delta) - \mu_c}{\Delta} = p.$$

Hence $\hat{p}$ is unbiased in the sense of (1).

(2) Consistency. Since $W_t = |\tilde{z}_t| \in [0, 2\beta_{\text{eff}}]$ is bounded, in particular $\mathbb{E}[|W_t|] < \infty$. Conditioned on $\mathbf{x}$ the W_t are i.i.d. with mean $\mu_c(\mathbf{x}) + p\Delta$, so by the Law of Large Numbers we have

$$\hat{\mu} = \frac{1}{\alpha}\sum_{t=1}^{\alpha} W_t \xrightarrow{a.s.} \mu_c(\mathbf{x}) + p\,\Delta \quad \text{as } \alpha \to \infty.$$

Since $\hat{p} = g(\hat{\mu})$ with $g(t) := (t - \mu_c)/\Delta$ is a continuous function and $\Delta > 0$, the continuous mapping theorem gives

$$\hat{p} = g(\hat{\mu}) \xrightarrow{a.s.} g\big(\mu_c(\mathbf{x}) + p\,\Delta\big) = p + \frac{\mu_c(\mathbf{x}) - \mu_c}{\Delta}.$$

(3) Variance. All expectations in this part are conditional on $\mathbf{x}$. We compute $\mathrm{Var}(W_t)$ using the law of total variance. Let $E_t \in \{0, 1\}$ indicate whether relation t is noise-flipped ($E_t = 1$ with probability p, independent of $|\tilde{z}_t^{(c)}|$). Then $W_t = (1 - E_t)\,|\tilde{z}_t^{(c)}| + E_t\,(2\beta_{\text{eff}} - |\tilde{z}_t^{(c)}|)$. We can calculate:

$$\begin{aligned}\mathbb{E}[W_t^2] &= (1-p)\,\mathbb{E}[|\tilde{z}_t^{(c)}|^2] + p\,\mathbb{E}[(2\beta_{\text{eff}} - |\tilde{z}_t^{(c)}|)^2] \\ &= (1-p)\,\mathbb{E}[|\tilde{z}_t^{(c)}|^2] + p\big(4\beta_{\text{eff}}^2 - 4\beta_{\text{eff}}\,\mu_c + \mathbb{E}[|\tilde{z}_t^{(c)}|^2]\big) \\ &= \mathbb{E}[|\tilde{z}_t^{(c)}|^2] + 4p\beta_{\text{eff}}(\beta_{\text{eff}} - \mu_c). \end{aligned} \tag{7}$$

Using $\mathbb{E}[W_t]^2 = (\mu_c + p\Delta)^2$ from (6) and $\Delta = 2(\beta_{\text{eff}} - \mu_c)$:

$$\begin{aligned}
\text{Var}(W_t) &= \mathbb{E}[W_t^2] - \mathbb{E}[W_t]^2 \\
&= \mathbb{E}[|\tilde{z}_t^{(c)}|^2] + 4p\beta_{\text{eff}}(\beta_{\text{eff}} - \mu_c) - \big(\mu_c + p\Delta\big)^2 \\
&= \big(\mathbb{E}[|\tilde{z}_t^{(c)}|^2] - \mu_c^2\big) + 4p(\beta_{\text{eff}} - \mu_c)(\beta_{\text{eff}} - \mu_c) - p^2\Delta^2 \\
&= \text{Var}(|\tilde{z}^{(c)}|) + p(1-p)\Delta^2, \qquad (8)
\end{aligned}$$

Since $\hat{p} = \frac{\hat{\mu} - \mu_c}{\Delta}$ is a linear function of $\hat{\mu}$, and the W_t are i.i.d. given $\mathbf{x}$,

$$\text{Var}(\hat{p}) = \frac{\text{Var}(\hat{\mu})}{\Delta^2} = \frac{\text{Var}(W_t)}{\alpha\,\Delta^2} = \frac{\text{Var}(|\tilde{z}^{(c)}|) + p(1-p)\Delta^2}{\alpha\,\Delta^2}.$$

The decomposition has a natural interpretation: the second term $p(1-p)/\alpha$ is the variance of a Bernoulli estimator with success probability p, reflecting the discrete component of noise; the first term captures the additional uncertainty from the distribution of $|\tilde{z}^{(c)}|$. □

Remark 3 (Residual key-dependence). The bias term in Theorem 1 (point 2) does not decay with α: averaged over keys it vanishes, but for the single fixed key an attacker actually faces it persists. Its size is governed by the spread of $\mu_c(\mathbf{x})$ around μ_c. Empirically this spread is small compared to Δ in the high-leakage regime. In the low-leakage regime, where Δ is one to two orders of magnitude smaller, the same absolute spread translates into a correspondingly larger bias leading to worse results for our estimator. Empirical results for these values can be found in Table 2.

We will now calculate μ_c in order to be able to apply Theorem 1.

Proposition 2 (Law of the transformed value, high-leakage regime). *Let $2^{j-1} > \beta$ and condition on $S := \langle \mathbf{c}, \mathbf{x} \rangle = s$. Conditioned on the relation being informative, the transformed value satisfies*

$$\Pr\big[\tilde{z}^{(c)} = t \;\big|\; S = s\big] \;=\; \frac{\mathbb{1}_{\{-\beta,\ldots,\beta-1\}}(t-s) \;+\; \mathbb{1}_{\{0\}}(t)}{2\beta+1}. \qquad (9)$$

Equivalently, $\tilde{z}^{(c)} = s + \tilde{y}$ where $\tilde{y}$ is uniform on the multiset $\{-\beta, \ldots, \beta-1\} \uplus \{-s\}$ of cardinality $2\beta + 1$.

Proof. Write $h := 2^{j-1}$. The extraction yields $\overline{z} = s + [y]_{2^j}$ with $[y]_{2^j}$ uniform on $\{-h, \ldots, h-1\}$, and the relation is informative iff $|\overline{z}| \geq h - \beta$.

Positive branch. $\overline{z} \geq h - \beta$ holds iff $[y]_{2^j} \geq h - \beta - s$, i.e. for $(h-1) - (h - \beta - s) + 1 = \beta + s$ values of $[y]_{2^j}$. On this branch (1) gives $\tilde{z} = \overline{z} - h + \beta = s + \tilde{y}$ with $\tilde{y} = [y]_{2^j} - h + \beta$ ranging over $\{-s, \ldots, \beta - 1\}$.

Negative branch. $\overline{z} \leq -(h - \beta)$ holds iff $[y]_{2^j} \leq -h + \beta - s$, i.e. for $(-h + \beta - s) - (-h) + 1 = \beta - s + 1$ values, and $\tilde{z} = \overline{z} + h - \beta = s + \tilde{y}$ with $\tilde{y} = [y]_{2^j} + h - \beta$ ranging over $\{-\beta, \ldots, -s\}$.

Parameter set	Regime	j	$\mathrm{sd}(\mu_c(\mathbf{x}))$	$\mathrm{sd}\big((\mu_c(\mathbf{x})-\mu_c)/\Delta\big)$
ML-DSA-44	low	4	0.072	$4.0 \cdot 10^{-2}$
	low	5	0.108	$9.5 \cdot 10^{-3}$
	low	6	0.062	$2.1 \cdot 10^{-3}$
	low	7	0.031	$5.0 \cdot 10^{-4}$
	high	≥ 8	0.026	$3.3 \cdot 10^{-4}$
ML-DSA-65	low	5	0.146	$4.3 \cdot 10^{-2}$
	low	6	0.232	$1.0 \cdot 10^{-2}$
	low	7	0.136	$2.3 \cdot 10^{-3}$
	low	8	0.068	$5.4 \cdot 10^{-4}$
	high	≥ 9	0.046	$2.3 \cdot 10^{-4}$
ML-DSA-87	low	4	0.051	$6.3 \cdot 10^{-2}$
	low	5	0.140	$1.5 \cdot 10^{-2}$
	low	6	0.097	$3.4 \cdot 10^{-3}$
	low	7	0.049	$7.9 \cdot 10^{-4}$
	high	≥ 8	0.026	$2.2 \cdot 10^{-4}$

Table 2: Standard deviation of the key-dependent quantity $\mu_c(\mathbf{x})$ over uniformly random secret keys, and the resulting standard deviation of the limiting bias $(\mu_c(\mathbf{x}) - \mu_c)/\Delta$ of $\hat{p}$. Rows are obtained by evaluating $\mu_c(\mathbf{x})$ exactly for 400 random keys.

The two shifts are chosen so that each branch maps its *inner* edge to zero: $\overline{z} = h - \beta \mapsto 0$ and $\overline{z} = -(h - \beta) \mapsto 0$. Both inner edges are informative, so the map $\overline{z} \mapsto \tilde{z}$ is injective on each branch but two-to-one at $\tilde{z} = 0$, i.e. at $\tilde{y} = -s$. For $s > -\beta$ the two ranges together cover $\{-\beta, \ldots, \beta - 1\}$ and the total count is $(\beta + s) + (\beta - s + 1) = 2\beta + 1$, independent of s. For $s = -\beta$ the positive branch is empty and the negative branch alone covers $\{-\beta, \ldots, \beta\}$, again $2\beta + 1$ outcomes, but each attained once. □

Remark 4. Proposition 2 refines Theorem 3 of [2] (stated in Equation (1)), which describes $\tilde{y}$ as uniform on $[\beta]$ and independent of $\langle \mathbf{c}, \mathbf{x} \rangle$. The exact law is supported on the multiset $\{-\beta, \ldots, \beta - 1\} \uplus \{-s\}$. The distinction is immaterial for the attacks of Damm *et al.* [2] and Schubert *et al.* [3], which use only the bound $|\tilde{y}| \leq \beta_{\mathrm{eff}}$, but it shifts $\mathbb{E}[|\tilde{z}^{(c)}|]$ by $\beta/(2\beta + 1)$ and therefore matters for a moment-based estimator.

Lemma 1 (Conditional expectation). *Let $s \in \mathbb{Z}$ with $|s| \leq \beta_{\mathrm{eff}}$ and set*

$$N := \begin{cases} 2\beta_{\mathrm{eff}} + 1, & \textit{high-leakage regime } (2^{j-1} > \beta), \\ 2\beta_{\mathrm{eff}}, & \textit{low-leakage regime } (2^{j-1} \leq \beta). \end{cases}$$

Then in both regimes

$$\mathbb{E}_{\tilde{y}}\big[|s + \tilde{y}|_M\big] = \frac{1}{N} \sum_{k=-\beta_{\mathrm{eff}}}^{\beta_{\mathrm{eff}}-1} |s + k| = \frac{\beta_{\mathrm{eff}}^2 + s^2 - s}{N} \tag{10}$$

holds. However in the low-leakage scenario $|s| > \beta_{\text{eff}}$ *is possible and in that case we end up with*

$$\mathbb{E}_{\tilde{y}}\big[\,|[s+\tilde{y}]_M|\,\big] \;=\; \frac{1}{2\beta_{\text{eff}}} \sum_{k=-\beta_{\text{eff}}}^{\beta_{\text{eff}}-1} |s+k|_M. \tag{11}$$

Proof. In the low-leakage regime $\tilde{y} = [y]_{2^j}$ is uniform on the $2\beta_{\text{eff}}$ values $\{-\beta_{\text{eff}}, \ldots, \beta_{\text{eff}}-1\}$. In the high-leakage regime Proposition 2 gives $2\beta_{\text{eff}}+1$ equiprobable outcomes, namely the same $2\beta_{\text{eff}}$ values of $\tilde{y}$ together with a second copy of $\tilde{y} = -s$; the duplicated outcome contributes $|s+(-s)| = 0$ to the sum and therefore changes only the value of N, not the summation range. This yields the first equality in (10) in both regimes.

For the second equality in (10), the values $s+k$ for $k \in \{-\beta_{\text{eff}}, \ldots, \beta_{\text{eff}}-1\}$ change sign at $k = -s$. Splitting the sum accordingly and substituting $i = -s-k$ (negative part) and $i = s+k$ (positive part):

$$\begin{aligned}\sum_{k=-\beta_{\text{eff}}}^{\beta_{\text{eff}}-1} |s+k| &= \underbrace{\sum_{k=-\beta_{\text{eff}}}^{-s-1} (-s-k)}_{=\,\sum_{i=1}^{\beta_{\text{eff}}-s} i} + 0 \;+\; \underbrace{\sum_{k=-s+1}^{\beta_{\text{eff}}-1} (s+k)}_{=\,\sum_{i=1}^{\beta_{\text{eff}}+s-1} i} \\ &= \frac{(\beta_{\text{eff}}-s)(\beta_{\text{eff}}-s+1)}{2} + \frac{(\beta_{\text{eff}}+s)(\beta_{\text{eff}}+s-1)}{2} \;=\; \beta_{\text{eff}}^2 + s^2 - s.\end{aligned}$$

Note that for the case of $s = -\beta_{\text{eff}}$ we have $\sum_{i=1}^{-1} i = 0$ and all cases are handled by the first sum. Further note that due to $s+k \in \{-2\beta_{\text{eff}}, \ldots, 2\beta_{\text{eff}}-1\}$ no centered reduction happens here.
The case of $|s| > \beta_{\text{eff}}$ can occur in the low-leakage regime, where we have to be careful since $\tilde{z}$ is calculated with a centered reduction modulo $M = 4\beta_{\text{eff}}$. Evaluating the reduced summands directly yields (11). □

Theorem 2. *In the high-leakage regime it holds that*

$$\mu_c \;=\; \frac{\beta_{\text{eff}}^2 + \mathrm{Var}(S)}{2\beta_{\text{eff}}+1} \;=\; \frac{\beta_{\text{eff}}^2 + \dfrac{(\eta+1)\cdot\beta_{\text{eff}}}{3}}{2\beta_{\text{eff}}+1} \;=\; \frac{\beta_{\text{eff}}(3\beta_{\text{eff}}+\eta+1)}{6\beta_{\text{eff}}+3}. \tag{12}$$

Parameter set	η	τ	β	μ_c (exact)	μ_c	μ_f	Δ
ML-DSA-44	2	39	78	$\frac{78\cdot 237}{471}$	39.25	116.75	77.50
ML-DSA-65	4	49	196	$\frac{196\cdot 593}{1179}$	98.58	293.42	194.84
ML-DSA-87	2	60	120	$\frac{120\cdot 363}{723}$	60.25	179.75	119.50

Table 3: Exact values of μ_c, $\mu_f = 2\beta - \mu_c$, and $\Delta = \mu_f - \mu_c$, computed via (12) and (13) in the high-leakage regime. All values are rational; the decimal expansions are rounded to two places.

Proof. Applying the law of total expectation, Lemma 1 with $N = 2\beta_{\text{eff}} + 1$, and $\mathbb{E}[S] = 0$:

$$\begin{aligned}\mu_c = \mathbb{E}_S\big[\mathbb{E}_{\tilde{y}}\big[|S+\tilde{y}|\big] \,\big|\, S\big] &= \mathbb{E}_S\left[\frac{\beta_{\text{eff}}^2 + S^2 - S}{2\beta_{\text{eff}}+1}\right] \\ &= \frac{\beta_{\text{eff}}^2 + \mathbb{E}[S^2] - \mathbb{E}[S]}{2\beta_{\text{eff}}+1} \\ &= \frac{\beta_{\text{eff}}^2 + \mathbb{E}[S^2] - \mathbb{E}[S]^2}{2\beta_{\text{eff}}+1} \\ &= \frac{\beta_{\text{eff}}^2 + \mathrm{Var}(S)}{2\beta_{\text{eff}}+1}.\end{aligned}$$

Lemma 18(b) of [4] gives $\mathrm{Var}(S) = \eta\tau(\eta+1)/3 = \beta(\eta+1)/3$ and yields the theorem. □

Corollary 3. *Consequently in the high-leakage regime*

$$\Delta \;=\; 2\beta_{\text{eff}} - 2\mu_c \;=\; \frac{2\beta_{\text{eff}}\,(3\beta_{\text{eff}} - \eta + 2)}{6\beta_{\text{eff}} + 3}. \tag{13}$$

Remark 5. Note that (12) depends only on the public parameters β and η. In the high-leakage regime the formula is again independent of j. This also leads to $\mu_c = \frac{\beta(3\beta+\eta+1)}{6\beta+3} \approx \frac{\beta}{2}$. Numerical values for the three ML-DSA parameter sets in the high-leakage regime are given in Table 3.

Theorem 3. *In the low-leakage regime it holds that*

$$\begin{aligned}\mu_c \;=\; &\sum_{s=-\beta_{\text{eff}}}^{\beta_{\text{eff}}} \phi(s)\cdot\frac{\beta_{\text{eff}}^2 + s^2 - s}{2\beta_{\text{eff}}} \\ &+ \sum_{s=\beta_{\text{eff}}+1}^{\beta} \phi(s)\cdot\frac{1}{2\beta_{\text{eff}}}\sum_{k=-\beta_{\text{eff}}}^{\beta_{\text{eff}}-1} |s+k|_M \\ &+ \sum_{s=-\beta}^{-\beta_{\text{eff}}-1} \phi(s)\cdot\frac{1}{2\beta_{\text{eff}}}\sum_{k=-\beta_{\text{eff}}}^{\beta_{\text{eff}}-1} |s+k|_M,\end{aligned} \tag{14}$$

with $\phi(s) := \Pr(S = s) = \xi(s)/(2\eta+1)^\tau$ *with* $\xi(s)$ *given by Lemma 18(e) of [4]:*

$$\xi(s) = \begin{cases} \sum_{i=0}^{\lfloor (s+\beta)/(2\eta+1) \rfloor} (-1)^i \binom{\tau}{i} \binom{s+\beta-(2\eta+1)i+\tau-1}{\tau-1}, & s \in [\beta], \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Analogously to Bashiri *et al.* [4] (Lemma 18) we define $\xi(s) := \#\{(u_1, \ldots, u_\tau) \mid u_i \in [\eta], \sum_{i=1}^{\tau} u_i = s\}$ and $\phi(s) := \frac{\xi(s)}{(2\eta+1)^\tau} = \mathbb{P}(S = s)$. So we get:

$$\begin{aligned} \mu_c &= \sum_{s=-\beta}^{\beta} \mathbb{P}(S = s) \cdot \mathbb{E}_{\tilde{y}}\left[|s + \tilde{y}|_M\right] \\ &= \sum_{s=-\beta_{\text{eff}}}^{\beta_{\text{eff}}} \phi(s) \cdot \mathbb{E}_{\tilde{y}}\left[|s + \tilde{y}|_M\right] + \sum_{s=\beta_{\text{eff}}+1}^{\beta} \phi(s) \cdot \mathbb{E}_{\tilde{y}}\left[|s + \tilde{y}|_M\right] \\ &\quad + \sum_{s=-\beta}^{-\beta_{\text{eff}}-1} \phi(s) \cdot \mathbb{E}_{\tilde{y}}\left[|s + \tilde{y}|_M\right] \end{aligned}$$

Applying Lemma 1 with $N = 2\beta_{\text{eff}}$ to the first sum and (11) to the remaining two, together with Lemma 18(e) of [4], yields the formula. □

Remark 6. We note that while unwieldy the formula from Theorem 3 only uses public parameters and the leakage index j and can therefore be calculated beforehand. The results are shown in Table 4.
This table also suggests why the algorithms of Schubert *et al.* [3] and Bashiri *et al.* [4] fail at index 3 for ML-DSA-44 and ML-DSA-87 and at index 4 for ML-DSA-65: The values $\tilde{z}$ are nearly-uniformly distributed even in the noise-free case.

Remark 7 (Numerical verification). We verified (12) and Theorem 3 against direct simulation of the extraction and of the transformation (1). The empirical means agree with the closed forms to within Monte Carlo error in both regimes.

Remark 8 (Practical use). An attacker collects α informative signatures, computes the empirical mean $\hat{\mu}$ of the extracted values $|\tilde{z}_t|$, looks up the precomputed μ_c for the target parameter set and leakage index (e.g. Table 4), and applies Theorem 1 to calculate

$$\hat{p} = \frac{\hat{\mu} - \mu_c}{\Delta}$$

and estimate the error rate p depending on the setting.
The estimate $\hat{p}$ can then be used to:

1. Predict the number of signatures required for key recovery, e.g. via the formula of Bashiri *et al.* [4] (their Remark 54) or via tables provided in [8, 3, 4]
2. Invert all leaked bits in the case of error rates above 0.5.
3. Initialize the preprocessing algorithm of Section 4.

Parameter set	Regime	j	β	β_{eff}	μ_c	μ_f	Δ
ML-DSA-44	low	3	78	4	4.00	4.00	0.01
	low	4	78	8	7.09	8.91	1.81
	low	5	78	16	10.34	21.66	11.31
	low	6	78	32	17.22	46.78	29.56
	low	7	78	64	32.61	95.39	62.78
	high	≥ 8	78	78	39.25	116.75	77.50
ML-DSA-65	low	4	196	8	7.99	8.01	0.01
	low	5	196	16	14.31	17.69	3.39
	low	6	196	32	20.88	43.12	22.23
	low	7	196	64	34.55	93.45	58.90
	low	8	196	128	65.28	190.72	125.45
	high	≥ 9	196	196	98.58	293.42	194.84
ML-DSA-87	low	3	120	4	4.00	4.00	0.00
	low	4	120	8	7.60	8.40	0.80
	low	5	120	16	11.38	20.62	9.25
	low	6	120	32	17.87	46.13	28.25
	low	7	120	64	32.94	95.06	62.13
	high	≥ 8	120	120	60.25	179.75	119.50

Table 4: Exact values of $\mu_c = \mathbb{E}[|\tilde{z}^{(c)}|]$, $\mu_f = 2\beta_{\text{eff}} - \mu_c$, and $\Delta = \mu_f - \mu_c$ for all ML-DSA parameter sets across both leakage regimes. In the high-leakage regime ($2^{j-1} > \beta$), $\beta_{\text{eff}} = \beta$, the residual is taken over $\mathbb{Z}$ and the values are independent of j; they are computed via the closed form of Theorem 2. In the low-leakage regime ($2^{j-1} \leq \beta$), $\beta_{\text{eff}} = 2^{j-1}$, the centered reduction $[\cdot]_M$ modulo $M = 2^{j+1}$ is applied to $\tilde{z}^{(c)}$, and μ_c is computed via Theorem 3. Values are rounded to two decimal places. The complement identity $\mu_c + \mu_f = 2\beta_{\text{eff}}$ holds exactly in both regimes.

3.1 Asymmetric Noise

During the rest of this work we only care for the overall noise rate and in the implementations that is reflected as a noise rate that is the same for all bits. In practice side-channel leakage often has two different noise rates for bits that are equal to 0 or 1. In this section we want to take a look at our noise-estimator in combination with such asymmetric noise rates.

We know that the $y_i^{(c)}$ are equally distributed between 0 and 1. Therefore

$$\Pr(y_i^{(c)} = 0) = \frac{1}{2} = \Pr(y_i^{(c)} = 1).$$

If instead of a single noise rate p we assume two different noise rates p_0 and p_1 for two different observed bit flip probabilities, as typical in many side-channel

settings in practice, we define

$$p_0 \coloneqq \Pr(y_i = 1 \mid y_i^{(c)} = 0),$$
$$p_1 \coloneqq \Pr(y_i = 0 \mid y_i^{(c)} = 1).$$

Combining this we get

$$\begin{aligned}\Pr(y_i = 1) &= \Pr(y_i = 1 \mid y_i^{(c)} = 1) \cdot \Pr(y_i^{(c)} = 1) + \Pr(y_i = 1 \mid y_i^{(c)} = 0) \cdot \Pr(y_i^{(c)} = 0) \\ &= (1 - p_1) \cdot \frac{1}{2} + p_0 \cdot \frac{1}{2} = \frac{1 - p_1 + p_0}{2}\end{aligned}$$

and

$$\begin{aligned}p &\coloneqq \Pr(y_i \neq y_i^{(c)}) \\ &= \Pr(y_i = 0 \mid y_i^{(c)} = 1) \cdot \Pr(y_i^{(c)} = 1) + \Pr(y_i = 1 \mid y_i^{(c)} = 0) \cdot \Pr(y_i^{(c)} = 0) \\ &= \frac{p_1}{2} + \frac{p_0}{2} = \frac{p_0 + p_1}{2}.\end{aligned}$$

Proposition 3. *We define* $\hat{q}_1 = \frac{\#\{y_i \mid y_i = 1\}}{\#\{y_i\}}$. *And let* $\hat{p}$ *be the overall noise estimation. Then*

$$\hat{p}_1 = \frac{1}{2} + \hat{p} - \hat{q}_1$$

estimates the noise rate p_1 *and*

$$\hat{p}_0 = 2\hat{p} - \hat{p}_1$$

estimates p_0.

Proof. Using the above derived formulas we know

$$2p = p_0 + p_1 \Leftrightarrow p_0 = 2p - p_1 \tag{15}$$
$$2q_1 = 1 - p_1 + p_0 \tag{16}$$

Substituting (15) into (16) we receive

$$\begin{aligned}2q_1 &= 1 - p_1 + 2p - p_1 \\ \Leftrightarrow p_1 &= \frac{1}{2} + p - q_1\end{aligned}$$

Replacing the true values p and q_1 with their estimators $\hat{p}$ and $\hat{q}_1$ we receive the first estimator and (15) gives the second by replacing p and p_1 with their estimators $\hat{p}$ and $\hat{p}_1$. □

4 Informative Relation Noise Probability

In order to evaluate the relations further we look at the probability that a given value carries a falsely leaked bit. We begin with the probability for $|\tilde{z}^{(c)}|$ to lie above a given threshold.

Lemma 2 (Tail probability of $|\tilde{z}^{(c)}|$, high-leakage regime). *Let $2^{j-1} > \beta$ (high-leakage regime). For any integer k with $1 \le k \le 2\beta$,*

$$\Pr\big(|\tilde{z}^{(c)}| \ge k\big) \;=\; \frac{1}{2\beta+1} \sum_{s=-\beta}^{\beta} \phi(s)\, N_{\mathrm{high}}(s,k), \tag{17}$$

where $N_{\mathrm{high}}(s,k) = \max(0,\ s+\beta-k) + \max(0,\ \beta-s-k+1)$ and $\phi(s)$ as above.

Proof. By Proposition 2, conditioned on $S = s$ the value $\tilde{z}^{(c)}$ is uniform on the $2\beta+1$ outcomes consisting of $\{s-\beta, \dots, s+\beta-1\}$ together with one additional outcome $\tilde{z}^{(c)} = 0$. For $k \ge 1$ the additional outcome does not satisfy $|\tilde{z}| \ge k$, so it is enough to count the $t \in \{s-\beta, \dots, s+\beta-1\}$ with $|t| \ge k$. These are

$$\#\{t \ge k\} = \max(0,\ (s+\beta-1) - k + 1) = \max(0,\ s+\beta-k)$$

and

$$\#\{t \le -k\} = \max(0,\ (-k) - (s-\beta) + 1) = \max(0,\ \beta-s-k+1),$$

which gives $N_{\mathrm{high}}(s,k)$. Averaging over S with weights $\phi(s)$ yields (17). □

Lemma 3 (Tail probability of $|\tilde{z}^{(c)}|$, low-leakage regime). *Let $2^{j-1} \le \beta$ (low-leakage regime), $\beta_{\mathrm{eff}} = 2^{j-1}$, $M = 2^{j+1} = 4\beta_{\mathrm{eff}}$. The extracted value satisfies $\tilde{z}^{(c)} = [S + \tilde{y}]_M$ with $\tilde{y} \xleftarrow{\$} \{-\beta_{\mathrm{eff}}, \dots, \beta_{\mathrm{eff}} - 1\}$. For any integer k with $1 \le k \le 2\beta_{\mathrm{eff}}$,*

$$\Pr\big(|\tilde{z}^{(c)}| \ge k\big) \;=\; \frac{1}{2\beta_{\mathrm{eff}}} \sum_{s=-\beta}^{\beta} \phi(s) \cdot N_{\mathrm{low}}(s,k), \tag{18}$$

where

$$N_{\mathrm{low}}(s,k) \;=\; \begin{cases} \max(0,\ s+\beta_{\mathrm{eff}}-k) + \max(0,\ \beta_{\mathrm{eff}}-s-k+1) & \text{if } |s| \le \beta_{\mathrm{eff}}, \\ \#\big\{\tilde{y} \in \{-\beta_{\mathrm{eff}}, \dots, \beta_{\mathrm{eff}}-1\} : |[s+\tilde{y}]_M| \ge k\big\} & \text{if } |s| > \beta_{\mathrm{eff}}. \end{cases} \tag{19}$$

Proof. Conditioning on $S = s$ and using the independence of S and $\tilde{y}$:

$$\Pr\big(|\tilde{z}^{(c)}| \ge k\big) = \sum_{s=-\beta}^{\beta} \phi(s) \cdot \Pr\left(|[s+\tilde{y}]_M| \ge k\right) = \frac{1}{2\beta_{\mathrm{eff}}} \sum_{s=-\beta}^{\beta} \phi(s) \cdot N_{\mathrm{low}}(s,k).$$

First we establish the closed form of $N_{\mathrm{low}}(s,k)$ for $|s| \le \beta_{\mathrm{eff}}$. Since $\tilde{y} \in \{-\beta_{\mathrm{eff}}, \dots, \beta_{\mathrm{eff}} - 1\}$ and $|s| \le \beta_{\mathrm{eff}}$, the raw sum satisfies $s+\tilde{y} \in [-2\beta_{\mathrm{eff}},\ 2\beta_{\mathrm{eff}}-1] \subset [-M/2,\ M/2)$, so the centered reduction is the identity: $[s+\tilde{y}]_M = s+\tilde{y}$. Therefore $N_{\mathrm{low}}(s,k) = \#\{\tilde{y} : s+\tilde{y} \ge k\} + \#\{\tilde{y} : s+\tilde{y} \le -k\}$. The two counts are, respectively, $\max(0,\ \beta_{\mathrm{eff}} - 1 - (k-s) + 1) = \max(0,\ s+\beta_{\mathrm{eff}}-k)$ and $\max(0,\ (-k-s) - (-\beta_{\mathrm{eff}}) + 1) = \max(0,\ \beta_{\mathrm{eff}}-s-k+1)$, giving (19).

For $|s| > \beta_{\mathrm{eff}}$ the values $s+\tilde{y}$ can exceed $M/2$ in magnitude, wrapping under $[\cdot]_M$; no closed form for $N_{\mathrm{low}}(s,k)$ is available in general and it is evaluated directly from the definition by counting the values of $\tilde{y}$ fulfilling the requirement. □

Parameter set	j	Regime	β_{eff}	$T = 1.5\beta_{\text{eff}}$	$T = 1.6\beta_{\text{eff}}$	$T = 1.7\beta_{\text{eff}}$	$T = 1.8\beta_{\text{eff}}$	$T = 1.9\beta_{\text{eff}}$
ML-DSA-44	$j = 4$	low	8	0.2126	0.1624	0.1144	0.0680	0.0225
	$j = 5$	low	16	0.0605	0.0389	0.0232	0.0169	$6.7 \cdot 10^{-3}$
	$j = 6$	low	32	$4.35 \cdot 10^{-3}$	$1.27 \cdot 10^{-3}$	$4.46 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$	$3.77 \cdot 10^{-5}$
	$j = 7$	low	64	$4.75 \cdot 10^{-6}$	$1.03 \cdot 10^{-7}$	$1.96 \cdot 10^{-9}$	$7.33 \cdot 10^{-12}$	$2.15 \cdot 10^{-14}$
	$j \geq 8$	high	78	$4.57 \cdot 10^{-8}$	$3.64 \cdot 10^{-10}$	$3.71 \cdot 10^{-13}$	$5.14 \cdot 10^{-17}$	$2.53 \cdot 10^{-22}$
ML-DSA-65	$j = 4$	low	8	0.2808	0.2183	0.1560	0.0936	0.0312
	$j = 5$	low	16	0.2041	0.1537	0.1051	0.0814	0.0347
	$j = 6$	low	32	0.0607	0.0392	0.0268	0.0169	$8.59 \cdot 10^{-3}$
	$j = 7$	low	64	$4.6 \cdot 10^{-3}$	$1.64 \cdot 10^{-3}$	$6.08 \cdot 10^{-4}$	$1.67 \cdot 10^{-4}$	$4.71 \cdot 10^{-5}$
	$j = 8$	low	128	$6.5 \cdot 10^{-6}$	$2.37 \cdot 10^{-7}$	$4.6 \cdot 10^{-9}$	$4.4 \cdot 10^{-11}$	$1.86 \cdot 10^{-13}$
	$j \geq 9$	high	196	$1.30 \cdot 10^{-10}$	$4.80 \cdot 10^{-14}$	$1.11 \cdot 10^{-18}$	$1.67 \cdot 10^{-24}$	$2.62 \cdot 10^{-33}$
ML-DSA-87	$j = 4$	low	8	0.2509	0.1938	0.1377	0.0824	0.0274
	$j = 5$	low	16	0.0976	0.0682	0.0438	0.0331	0.0137
	$j = 6$	low	32	0.0121	$5.09 \cdot 10^{-3}$	$2.47 \cdot 10^{-3}$	$1.11 \cdot 10^{-3}$	$4.27 \cdot 10^{-4}$
	$j = 7$	low	64	$9.41 \cdot 10^{-5}$	$8.08 \cdot 10^{-6}$	$7 \cdot 10^{-7}$	$2.61 \cdot 10^{-8}$	$1.04 \cdot 10^{-9}$
	$j \geq 8$	high	120	$9.54 \cdot 10^{-11}$	$2.71 \cdot 10^{-14}$	$8.84 \cdot 10^{-19}$	$1.44 \cdot 10^{-24}$	$1.66 \cdot 10^{-32}$

Table 5: Rates at which a value $|\tilde{z}^{(c)}|$ is larger than the threshold T, i.e. $\Pr(|\tilde{z}^{(c)}| \geq T)$, for clean relations across both leakage regimes and all ML-DSA parameter sets. High-leakage rows use Lemma 2, low-leakage rows use Lemma 3.

It follows that for α correct informative relations, the chance of at least one exceeding a threshold T is

$$\Pr\big(\max_t |\tilde{z}_t^{(c)}| > T\big) = 1 - \big(1 - \Pr(|\tilde{z}^{(c)}| > T)\big)^{\alpha}.$$

Remark 9. As seen in Table 5 high values for $|\tilde{z}^{(c)}|$ are extremely unlikely, especially in the high-leakage regime. We can therefore classify all values above a certain threshold as noise-corrupted side-channel information and flip the leaked bit. For example, with a million clean relations at leakage index 9 for ML-DSA-65, flipping all relations with $|\tilde{z}| > 1.5\beta_{\text{eff}}$ leaves a 0.013% chance of having flipped a single correct relation.

As a direct consequence of Corollary 2 of [3] we get

$$\tilde{z}^{(c)} = \tilde{z}^{(f)} - \text{sign}(\tilde{z}^{(f)}) \cdot 2\beta_{\text{eff}}.$$

By calculating $\tilde{z} - \text{sign}(\tilde{z}) \cdot 2\beta_{\text{eff}}$ we can therefore compute the value we would have obtained had we extracted $\tilde{z}$ with the flipped leaked bit. This allows us to insert Algorithm 1 as an intermediate step between the extraction of $\tilde{z}$ and the use of the relations to find the correct subkey.

Algorithm 1 touches each relation once and performs a comparison and at most one addition, so it runs in $O(\alpha)$ time and negligible space relative to any of the downstream attacks. It consumes and produces informative relations in exactly the format used by [2, 3], so it can be inserted in front of any of these

Algorithm 1: ML-DSA leakage processing

Input: α extracted informative relations $(\tilde{z}_t)_{t=1,\dots,\alpha}$, effective bound β_{eff}, threshold T

Output: α preprocessed informative relations $(\tilde{z}'_t)_{t=1,\dots,\alpha}$

```
for t = 1, ..., α do
    if |z̃_t| > T then
        z̃'_t ← z̃_t − sign(z̃_t) · 2β_eff
    else
        z̃'_t ← z̃_t
return (z̃'_t)_{t=1,...,α}
```

attacks without modifying them. For attacks in a different format like [4], we can calculate $\tilde{z}$ and switch the leaked bit if necessary with little effort.

We can use Table 5[8] in combination with the noise rate[9] to estimate a threshold T that allows us to correct $\tilde{z}^{(f)}$ without incorrectly changing $\tilde{z}^{(c)}$.

4.1 Per-relation probabilities

In addition to the general estimate we want to derive a probability for each individual relation, so that we can decide at which probability level we prefer to flip the incoming bit.

Lemma 4. *In the high-leakage regime it holds for* $1 \le k \le 2\beta$ *that*

$$\Pr\big(|\tilde{z}^{(c)}| = k\big) = \frac{1}{2\beta+1}\left(\sum_{s=k-\beta+1}^{k+\beta}\phi(s) \;+\; \sum_{s=k-\beta}^{k+\beta-1}\phi(s)\right),$$

while $\Pr(\tilde{z}^{(c)} = 0) = \frac{2-\phi(\beta)}{2\beta+1}$.

Proof. By Proposition 2 and for $k \ge 1$,

$$\begin{aligned}\Pr\big(|\tilde{z}^{(c)}| = k\big) &= \Pr\big(\tilde{z}^{(c)} = k\big) + \Pr\big(\tilde{z}^{(c)} = -k\big)\\ &= \sum_{s=-\beta}^{\beta}\phi(s)\,\frac{\mathbb{1}_{\{-\beta,\dots,\beta-1\}}(k-s) + \mathbb{1}_{\{-\beta,\dots,\beta-1\}}(-k-s)}{2\beta+1},\end{aligned}$$

since the duplicated outcome $\tilde{z} = 0$ does not contribute for $k \ge 1$. The first indicator is non-zero exactly for $k-\beta+1 \le s \le k+\beta$ and the second exactly for $-k-\beta+1 \le s \le -k+\beta$; using the symmetry $\phi(s) = \phi(-s)$ the second range may be rewritten as $k-\beta \le s \le k+\beta-1$.

For $k = 0$, Proposition 2 gives $\Pr(\tilde{z}^{(c)} = 0 \mid S = s) = \big(\mathbb{1}_{\{-\beta,\dots,\beta-1\}}(-s) + 1\big)/(2\beta+1)$. The indicator vanishes precisely for $s = -\beta$, where $-s = \beta \notin \{-\beta,\dots,\beta-1\}$. The additional multiset-element is not a duplicate of an existing element in that one case. This yields $\Pr(\tilde{z}^{(c)} = 0) = \big(2-\phi(\beta)\big)/(2\beta+1)$. □

[8] Lemma 2 or Lemma 3 might also be used for the same purpose.

[9] Either known or calculated via the noise estimator described in Section 3.

Lemma 5. *In the low-leakage regime it holds for $k \in [1, 2\beta_{\text{eff}} - 1]$ that*

$$\begin{aligned}&\Pr\big(|\tilde{z}^{(c)}| = k\big)\\ =&\frac{1}{2\beta_{\text{eff}}} \sum_{s=-\beta}^{\beta} \phi(s) \left(\mathbb{1}_{\{-\beta_{\text{eff}},\ldots,\beta_{\text{eff}}-1\}}([k-s]_M) + \mathbb{1}_{\{-\beta_{\text{eff}},\ldots,\beta_{\text{eff}}-1\}}([-k-s]_M)\right).\end{aligned}$$

Proof. This can be seen by

$$\begin{aligned}\Pr\big(|\tilde{z}^{(c)}| = k\big) &= \Pr\big([\tilde{z}^{(c)}]_M = k\big) + \Pr\big([\tilde{z}^{(c)}]_M = -k\big)\\ &= \Pr\big([S+\tilde{y}]_M = k\big) + \Pr\big([S+\tilde{y}]_M = -k\big)\\ &= \sum_{s=-\beta}^{\beta} \phi(s) \cdot \Pr\big(\tilde{y} = [k-s]_M\big) + \sum_{s=-\beta}^{\beta} \phi(s) \cdot \Pr\big(\tilde{y} = [-k-s]_M\big)\\ &= \frac{1}{2\beta_{\text{eff}}} \sum_{s=-\beta}^{\beta} \phi(s) \Big(\mathbb{1}_{\{-\beta_{\text{eff}},\ldots,\beta_{\text{eff}}-1\}}([k-s]_M)\\ &\quad + \mathbb{1}_{\{-\beta_{\text{eff}},\ldots,\beta_{\text{eff}}-1\}}([-k-s]_M)\Big),\end{aligned}$$

where we used that $\tilde{y}$ is uniform on the $2\beta_{\text{eff}}$ values $\{-\beta_{\text{eff}}, \ldots, \beta_{\text{eff}} - 1\}$. □

Theorem 4. *It holds that*

$$Pr\big(\tilde{z} = \tilde{z}^{(c)} \big| |\tilde{z}| = k\big) = \frac{\Pr(|\tilde{z}^{(c)}| = k) \cdot (1-p)}{\Pr(|\tilde{z}^{(c)}| = k)\,(1-p) + \Pr(|\tilde{z}^{(c)}| = 2\beta_{\text{eff}} - k)\,p}. \tag{20}$$

Proof. Lemma 4 and Lemma 5 provide the evaluation for correct leaked bits only. For incorrect bits, Corollary 2 gives:

$$\Pr\big(|\tilde{z}^{(f)}| \le k\big) = \Pr\big(2\beta_{\text{eff}} - |\tilde{z}^{(c)}| \le k\big) = \Pr\big(|\tilde{z}^{(c)}| \ge 2\beta_{\text{eff}} - k\big)$$

$$\Pr\big(|\tilde{z}^{(f)}| = k\big) = \Pr\big(2\beta_{\text{eff}} - |\tilde{z}^{(c)}| = k\big) = \Pr\big(|\tilde{z}^{(c)}| = 2\beta_{\text{eff}} - k\big). \tag{21}$$

Since we know that every observation $\tilde{z}$ stems from a correctly or incorrectly leaked bit, we obtain via Bayes' Theorem

$$\begin{aligned}\Pr\big(\tilde{z} = \tilde{z}^{(c)} \big| |\tilde{z}| = k\big) &= \frac{\Pr(|\tilde{z}^{(c)}| = k) \cdot (1-p)}{\Pr(|\tilde{z}^{(c)}| = k)\,(1-p) + \Pr(|\tilde{z}^{(f)}| = k)\,p}\\ &\overset{(21)}{=} \frac{\Pr(|\tilde{z}^{(c)}| = k) \cdot (1-p)}{\Pr(|\tilde{z}^{(c)}| = k)\,(1-p) + \Pr(|\tilde{z}^{(c)}| = 2\beta_{\text{eff}} - k)\,p}\end{aligned}$$

□

We can therefore assign to each relation a probability of its leaked bit being correct (or incorrect), since the right-hand side of (20) depends only on known parameters and on the noise rate p, which we can estimate via the noise estimator of Section 3. To define a *threshold* rule based on Theorem 4 we first establish monotonicity in k:

Lemma 6 (The distribution of $|\tilde{z}^{(c)}|$ is non-increasing in the high-leakage regime). *Write $A(k) := \Pr(|\tilde{z}^{(c)}| = k)$. In the high-leakage regime we have, for every k with $1 \le k \le 2\beta - 1$,*

$$A(k) - A(k+1) \;=\; \frac{\phi(k-\beta) + \phi(k-\beta+1)}{2\beta+1} \;\ge\; 0,$$

so A is non-increasing on $\{1, \ldots, 2\beta\}$.

Proof. By Lemma 4 we can write $A(k) = \big(F(k) + G(k)\big)/(2\beta+1)$ with

$$F(k) = \sum_{s=k-\beta+1}^{k+\beta} \phi(s), \qquad G(k) = \sum_{s=k-\beta}^{k+\beta-1} \phi(s).$$

Both are sums over a window of fixed width 2β. Replacing k by $k+1$ shifts each window one step to the right, which removes its leftmost term and adds a new rightmost one:

$$F(k{+}1) = F(k) - \phi(k{-}\beta{+}1) + \phi(k{+}\beta{+}1), \qquad G(k{+}1) = G(k) - \phi(k{-}\beta) + \phi(k{+}\beta).$$

For $k \ge 1$ we have $k+\beta+1 > \beta$ and $k+\beta > \beta$, so by definition of $\xi(s)$ both newly added terms are zero. Hence $F(k) - F(k+1) = \phi(k-\beta+1)$ and $G(k) - G(k+1) = \phi(k-\beta)$, and both are non-negative because ϕ is a probability mass function. Adding the two and dividing by $2\beta+1$ gives the claim. □

Remark 10. In the low-leakage regime the values $\tilde{z}$ wrap around modulo M, so the sliding-window argument above does not apply directly. We verified numerically that A is non-increasing for all three ML-DSA parameter sets and all leakage indices $j \ge 3$; since A is a finite table computable in advance from Lemma 5, this can be checked for any concrete parameter choice before mounting the attack.

Lemma 7 (Monotonicity of the posterior). *Let A be non-increasing on $\{1, \ldots, 2\beta_{\text{eff}}\}$ and let $1 \le k \le k' \le 2\beta_{\text{eff}} - 1$ be two observable values, i.e. values for which the denominator of (20) is non-zero. Then*

$$\Pr\big(\tilde{z} = \tilde{z}^{(c)} \,\big|\, |\tilde{z}| = k\big) \;\ge\; \Pr\big(\tilde{z} = \tilde{z}^{(c)} \,\big|\, |\tilde{z}| = k'\big).$$

Proof. For $p = 0$ both sides equal 1, so assume $p > 0$ and abbreviate $a := A(k)(1-p)$, $b := A(2\beta_{\text{eff}} - k)\,p$ and likewise a', b' for k'. Then using (20) we get

$$\begin{aligned}
&\Pr\big(\tilde{z} = \tilde{z}^{(c)} \,\big|\, |\tilde{z}| = k\big) = \frac{a}{a+b} \ge \frac{a'}{a'+b'} = \Pr\big(\tilde{z} = \tilde{z}^{(c)} \,\big|\, |\tilde{z}| = k'\big) \\
&\Leftrightarrow a(a'+b') \ge a'(a+b) \\
&\Leftrightarrow ab' \ge a'b \\
&\Leftrightarrow A(k)\,A(2\beta_{\text{eff}} - k') \;\ge\; A(k')\,A(2\beta_{\text{eff}} - k).
\end{aligned}$$

Since $k \le k'$ and A is non-increasing, $A(k) \ge A(k')$. Since also $2\beta_{\text{eff}} - k' \le 2\beta_{\text{eff}} - k$, the same monotonicity gives $A(2\beta_{\text{eff}} - k') \ge A(2\beta_{\text{eff}} - k)$. Multiplying the two inequalities between non-negative numbers yields the claim. □

We can then calculate a threshold T as

$$T \;=\; \min\left\{T' \in \mathbb{N} \;\middle|\; \Pr\left(\tilde{z} = \tilde{z}^{(c)} \,\middle|\, |\tilde{z}| = T'\right) < t\right\} \tag{22}$$

with t being the probability-threshold we would like to accept.[10] By Lemmas 6 and 7 every relation with $|\tilde{z}| \geq T$ then also falls below that level, so thresholding on $|\tilde{z}|$ is equivalent to thresholding on the posterior.

5 Results

This section presents the experimental evaluation of our presented ideas. Section 5.1 covers the estimation of low-leakage noise levels; Section 5.2 evaluates the effects of preprocessing on the existing algorithms. Our Python reference implementation, experiment scripts, and all raw data described in this section are made publicly available at `https://github.com/niotarho/mldsa-statistical-inference`.

5.1 Noise Estimator

In order to test our estimator we generated a set of informative relations with a random noise rate $p \in [0, 1]$.[11] Based on these our noise estimator tried to guess the noise rate p. We evaluated the estimator on sets of $1000, 5000, 25000$ informative relations as well as for ML-DSA-44, ML-DSA-65, and ML-DSA-87 for the leakage indices 4 to 9 on 1000 runs each.

As expected based on Table 4 the higher the leakage index and therefore the higher the separator Δ the better are the predictions of our noise estimator (see Table 6). Especially for ML-DSA-65 and leakage index 4 the results were unusable.[12] As expected more informative relations give a closer estimate of the true noise rate. However even at just $1,000$ informative relations we already have reasonable estimates in most settings. If an even more accurate guess is needed there would be the option of combining this noise estimator with the noise estimator suggested by Schubert *et al.* [3].

Asymmetric Noise Estimator In order to test our asymmetric noise estimator (Section 3.1) we used the same setup as for the symmetric version we use throughout the rest of this paper.

The results (see Table 7) show that while the accuracy is slightly below the symmetric estimator – which is expected due to the estimation of q_1 introducing additional uncertainty – the results are still usable and within tolerable margins.

[10] A probability threshold of 0.01 would mean that we are 99% sure that a relation of that value is false.

[11] Definition 1 requires $p \in [0, \frac{1}{2})$ but the noise estimator applies unchanged for $p \geq 0.5$ and even enables via Remark 8 a recovery for noise rates above 0.5 by flipping all leaked bits.

[12] The same would be expected for all lower leakage indices across all parameter sets.

		ML-DSA-44		ML-DSA-65		ML-DSA-87	
j	α	**MAE**	**STD-E**	**MAE**	**STD-E**	**MAE**	**STD-E**
4	1000	0.065866	0.082100	9.072461	11.470815	0.157629	0.194784
	5000	0.033760	0.042791	4.069648	5.106786	0.071761	0.089943
5	1000	0.017357	0.021735	0.071778	0.090665	0.021678	0.027641
	5000	0.008610	0.010953	0.035265	0.044622	0.011852	0.014977
6	1000	0.009618	0.011986	0.018565	0.023189	0.010996	0.013769
	5000	0.004411	0.005521	0.008625	0.011044	0.004926	0.006271
7	1000	0.008106	0.010145	0.010002	0.012527	0.007888	0.009968
	5000	0.003492	0.004363	0.004466	0.005590	0.003686	0.004567
8	1000	0.007579	0.009479	0.007702	0.009630	0.007276	0.009002
	5000	0.003578	0.004419	0.003442	0.004357	0.003164	0.003981
9	1000	0.007742	0.009589	0.007707	0.009532	0.007610	0.009562
	5000	0.003573	0.004451	0.003292	0.004151	0.003293	0.004160

Table 6: Noise-Estimator Results for ML-DSA-44, ML-DSA-65, and ML-DSA-87. Showing the leakage index j, the number of used informative relations α, as well as the Mean-Absolute-Error (MAE) and Standard-Error (STD-E) over 1,000 runs for each parameter set.

5.2 Simple preprocessing

We decided to use our preprocessing in combination with the attack of Schubert *et al.* [3] due to their code and setup being easily available and reproducible. Our goal was to compare the attack with and without preprocessing. We therefore elected the attack we could most easily and accurately replicate.

In order to be able to compare our work with Schubert *et al.* [3] we use the exact same settings: In their github-repository we found the commands used to generate their results. We therefore used the same commands and only inserted a preprocessing step to adjust the $\tilde{z}$ as described in Section 4. The used threshold is calculated with our noise estimator to estimate the noise level and subsequently calculating which changes are 99% or 99.99% correct and apply only those.[13] In other words, we applied a probability-threshold of 0.01 and 0.0001 respectively as these are the chances we are willing to take on a flip being harmful to our algorithm as per (22).

Similar to Schubert *et al.* [3] we ran 5 experiments in each setting and considered it a success if at least 4 key recoveries were successful, and left everything else untouched as well. We didn't alter any other code to ensure a direct comparability of using the preprocessing with not using it. To give an

[13] 99.99% was used for the high-leakage case and the low-leakage case closest to the high-leakage case while 99% was used for all other cases. As based on Table 5 these would still result in a meaningful number of flipped informative relations.

ML-DSA	j	MAE($\hat{p}$)	STD-E($\hat{p}$)	MAE($\hat{p}_0$)	STD-E($\hat{p}_0$)	MAE($\hat{p}_1$)	STD-E($\hat{p}_1$)
44	4	0.031738	0.039771	0.032057	0.040027	0.031878	0.040021
	5	0.007599	0.009535	0.008090	0.010259	0.008282	0.010383
	6	0.004349	0.005464	0.005507	0.006898	0.005431	0.006933
	7	0.003432	0.004282	0.004581	0.005818	0.004561	0.005748
	8	0.003390	0.004225	0.004486	0.005773	0.004515	0.005839
	9	0.003560	0.004459	0.004741	0.005975	0.004736	0.006018
65	4	4.069487	5.104979	4.069466	5.104906	4.069652	5.105075
	5	0.032225	0.040437	0.032614	0.040749	0.032375	0.040560
	6	0.007921	0.010001	0.008552	0.010714	0.008724	0.010994
	7	0.004403	0.005482	0.005372	0.006732	0.005485	0.006975
	8	0.003525	0.004432	0.004740	0.006072	0.004703	0.006052
	9	0.003428	0.004290	0.004439	0.005833	0.004586	0.005935
87	4	0.069172	0.085860	0.069238	0.086045	0.069234	0.086021
	5	0.010850	0.013441	0.011245	0.014051	0.011123	0.013831
	6	0.004939	0.006141	0.006016	0.007628	0.005733	0.007223
	7	0.003676	0.004585	0.004737	0.006109	0.004690	0.006023
	8	0.003400	0.004313	0.004553	0.005968	0.004652	0.005931
	9	0.003294	0.004075	0.004680	0.005929	0.004542	0.005755

Table 7: Noise-Estimator Results for asymmetric noise. Showing the ML-DSA parameter set $\{44, 65, 87\}$, leakage index j, for $1,000$ runs with 5000 informative relations each the Mean-Absolute-Error (MAE) and Standard-Error (STD-E) for the overall noise rate p as well as the specific noise rates p_0 and p_1. The full results as for the symmetric estimator can be found in our github repository.

overview we looked at three parameter sets at 20% and 40% noise rate and across leakage indices $6, 7, 8$ for ML-DSA-44 and ML-DSA-87 as well as $7, 8, 9$ for ML-DSA-65. We chose this setup to ensure comparability with [3].

In these settings, see Table 8, we see a reduction of $20\% - 44\%$ in needed informative relations across our tested parameter sets with the same success criterion as Schubert *et al.* [3]. This might be increased further if one finds a way to determine a threshold based on the input parameter without having to guess a random threshold. Also, all but one scenario[14] reported an improvement of at least 30%.

During our tests we sometimes saw that the algorithm ends up with a candidate within Hamming distance 1 of the true subkey. While this case might be further exploitable by an attacker, we decided to still count it as failure due to Schubert *et al.* [3] not discussing it either, and we wanted to ensure strict comparability.

[14] ML-DSA-44 Leakage Index 6 with $p = 0.4$ reported only 20% reduction in needed informative relations in order to be a success.

			Parameter set		
Leakage j	**Preprocessed?**	**Noise rate**	ML-DSA-44	ML-DSA-65	ML-DSA-87
6	No	20%	50,000	–	50,000
	Yes	20%	35,000	–	–
	Advantage	20%	30%	–	–
	No	40%	500,000	–	500,000
	Yes	40%	400,000	–	300,000
	Advantage	40%	20%	–	40%
7	No	20%	100,000	110,000	90,000
	Yes	20%	70,000	70,000	60,000
	Advantage	20%	30%	36%	33%
	No	40%	1,200,000	1,100,000	900,000
	Yes	40%	700,000	650,000	500,000
	Advantage	40%	41%	40%	44%
8	No	20%	120,000	200,000	170,000
	Yes	20%	80,000	125,000	105,000
	Advantage	20%	33%	37%	38%
	No	40%	1,400,000	2,000,000	1,700,000
	Yes	40%	850,000	1,250,000	1,050,000
	Advantage	40%	39%	37%	38%
9	No	20%	–	320,000	–
	Yes	20%	–	200,000	–
	Advantage	20%	–	37%	–
	No	40%	–	3,100,000	–
	Yes	40%	–	1,950,000	–
	Advantage	40%	–	37%	–

Table 8: Leakage information required with preprocessing and without preprocessing. For each parameter set and leakage index j, we report the minimum number of informative relations at which at least 4 of 5 keys were recovered. There are no numbers given for ML-DSA-44 and ML-DSA-87 for leakage index 9 since that index is already covered by leakage index 8 due to both belonging to the high-leakage regime nd therefore no changes are expected. The rows without preprocessing represent the work of Schubert *et al.* [3]. In the advantage rows we noted the amount of fewer signatures needed in comparison. Note that there is no number given for the ML-DSA-87 Leakage Index 6 at 20% noise rate, this is due to the fact that our algorithm did not find any informative relations that contained a false leaked bit with a probability-threshold of 0.01. At a probability-threshold of 0.1 it achieved a success at 35,000 informative relations.

5.3 Parallel Preprocessing

Currently we don't know the ideal (probability-)threshold. So one idea would be to run multiple attacks with different (probability-)thresholds. All of these θ experiments would be independent of each other and can therefore be run in

parallel. Even if they can only be run sequentially they would only add a factor of $\theta \leq \beta_{\text{eff}}$ to the duration of the attack. While a single attack might not retrieve the correct subkey if any of the θ parallel attacks recovers the true subkey checking would only add another factor of θ to the duration that again can be parallelized.

To test our preprocessing with another setup we investigated the attack of Bashiri *et al.* [4] as well. Since Bashiri *et al.* [4] use a different formulation compared to Damm *et al.* [8] and Schubert *et al.* [3] as well as this work, we implemented our work as a step between signature generation and attack of Bashiri *et al.* [4] (Remark 67) and focused on the presented case in that remark only (ML-DSA-44, leakage index 8, Noise Rate 45%). We reduced the number of available informative relations to 1,900,000 in order to show the potential of this attack form. We also added two more noise rates: 40% at 450,000 informative relations as well as 20% at 40,000 informative relations. The amount of informative relations is in all cases close to the theoretical estimate of Bashiri *et al.* [4].

	0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
ML-DSA-44 at 45% noise rate with 1,900,000 inf. rel.											
recovered	16	0	7	14	17	17	20	17	16	16	19
gained	–	0	1	1	2	3	4	3	1	0	3
lost	–	16	10	3	1	2	0	2	1	0	0
ML-DSA-44 at 40% noise rate with 450,000 inf. rel.											
recovered	18	0	17	22	18	15	17	18	19	18	18
gained	–	0	3	6	1	0	1	1	1	0	0
lost	–	18	2	2	1	3	2	1	0	0	0
ML-DSA-44 at 20% noise rate with 40,000 inf. rel.											
recovered	19	9	18	21	20	23	22	18	22	22	20
gained	–	0	3	5	3	5	4	1	3	3	1
lost	–	10	4	3	2	1	1	2	0	0	0

Table 9: Success for key recovery with varying probability-thresholds in the preprocessing step. It is shown for how many out of 30 seeds the attack was successful and how many were gained/lost compared to the baseline attack (0).

As shown in Table 9, for the first 30 tested seeds the implementation of Bashiri *et al.* [4] is able to retrieve the correct key in 16/18/19 of them. By using different probability-thresholds we were able to increase the amount of successful recoveries overall to 23/25/29. The best probability-threshold did vary for these three experiments. We see an increase in recovery-rate of 21% to 25% for a single probability-threshold chosen in hindsight and an increase of 38.9% to 52.6% for a parallel processing and subsequent testing of all θ key-candidates.

6 Conclusion

Our noise estimator presents a simple option to calculate the noise rate of an ML-DSA side-channel from the informative relations an attack already collects. A higher leakage index separates the clean and the noise-flipped distributions of $|\tilde{z}|$ more sharply and therefore yields more accurate estimates, but our method is, to our knowledge, the first to work in the low-leakage regime at all, where no relation is discarded and the mechanism of Schubert *et al.* [3] is unavailable. Besides the informative relation it only needs access to the public parameters. A variant of our noise estimator can be used to recover separate rates for 0-bits and 1-bits, a common occurrence in practical side-channel leakage.

The preprocessing algorithm built on top of it offers an attacker several options. As shown in Section 5.2, it reduces the number of informative relations needed for key recovery by about 20% to 44% across the parameter sets, leakage indices and noise rates we tested. The algorithm itself is attack-agnostic: it consumes and produces informative relations in the format used by Damm *et al.* [8] and Schubert *et al.* [3] and runs in time linear in their number, so it can be placed in front of an existing attack without modifying it. However how the algorithm responds to our changes varies. We observe that the attack of Bashiri *et al.* [4] appears to benefit less from our preprocessing than that of Schubert *et al.* [3], although the experiments measure different quantities[15] and are therefore not directly comparable. A plausible explanation is the different way the two attacks absorb incorrect information. Schubert *et al.* [3] score a candidate by an unweighted count of violated relations, so a single incorrectly flipped relation shifts every candidate's score by at most one. Bashiri *et al.* [4], by contrast, optimize a continuous objective to which every informative relation contributes, so an incorrect flip can influence the search trajectory rather than merely perturb a score.

Alternatively, an attacker might, as in Section 5.3, use a parallel approach and use several thresholds on different machines at once. An attacker could run every reasonable threshold[16] at once, maximizing the chance of recovering the key. However this comes at the price of calculating the full key for each candidate independently. These β_{eff} runs are mutually independent and thus embarrassingly parallel.

Our findings reinforce the importance of safeguarding against side-channel leakage for ML-DSA signatures.

6.1 Open Problems

Determining a threshold. The biggest open problem is how to find the best threshold or at least a good threshold for a single shot for key recovery. A first

[15] A reduction in needed informative relations to archive a "success" vs. the recovery rate at a fixed number of informative relations.

[16] $T \in \{\beta_{\text{eff}} + 1, \ldots, 2\beta_{\text{eff}}\}$

idea of flipping all leaked bits that are more likely incorrect than correct[17] and therefore maximizing the amount of correctly leaked bits, leads to a threshold T way too low to yield any meaningful results. In the setting of Schubert *et al.* [3] the attack did not return any meaningful results at all. Table 9 shows that in the setting of Bashiri *et al.* [4] even a probability-threshold of 0.1 already recovers no keys and even 0.01 falls behind in recovery rate compared to the attack without preprocessing.

This is probably due to the fact that higher values of $|\tilde{z}|$ also contain more information about the true key x. Flipping those therefore diminishes the information contained in the informative relations. A model about the information contained in a single informative relation, extending the work of Bashiri *et al.* [4], might prove to be a tool to estimate more accurately when a flip does more good by increasing the chance to be correct compared to the information lost by the flip.

Another, empirical approach would be to characterize which thresholds succeed how often. In our experiments the most successful probability-threshold varied. Determining which thresholds are broadly effective could be a valuable first step to decrease the number of needed threshold trials.

Unified algorithm. A further direction would be to create a new algorithm that incorporates the possibility of a wrong information directly, rather than consuming the per-relation probabilities only in a separate preprocessing step. This could also align well with the excess metric approach of Schubert *et al.* [3] since there would be a balance between the information being more likely to be false and the excess it provides for wrong key candidates. Another way would be to incorporate them as distribution hints in the way of Hermelink *et al.* [12].

Broader experimental validation. Our evaluation in Section 5.2 compares preprocessed and non-preprocessed key recovery across three parameter sets, four leakage indices, and two noise rates, but relies on five runs per setting, the same protocol as Schubert *et al.* [3], which we adopted for direct comparability, but one that limits the statistical confidence attainable for the reported reduction. Likewise, our evaluation against the attack of Bashiri *et al.* [4] in Section 5.3 covers a single (parameter set, leakage index, noise rate) configuration. A more thorough evaluation, more seeds per setting to obtain confidence intervals on the reduction in required informative relations, and coverage of additional parameter/index/noise-rate combinations for the Bashiri *et al.* [4] comparison, would strengthen confidence in the generality of both results.

Relatedly, our results characterize only the point at which key recovery becomes reliable in the sense of Schubert *et al.* [3] (at least 4 of 5 seeds succeeding), which reflects a typical, near-worst-case behavior. We know little about the other end of the curve: the smallest number of informative relations at which any seed first succeeds, and how success probability grows between that point and the reliable-recovery threshold reported in Table 8. Characterizing this onset behavior rather than only the point of near-certain recovery would give a clearer

[17] Represented by a probability threshold of 0.5.

picture of the preprocessing's effect and could also inform the threshold-selection problem discussed above, since early-success cases may correspond to particularly favorable thresholds. Additionally such an evaluation could also show more about the applicability and success probabilities of the previous works of Schubert *et al.* [3] and Bashiri *et al.* [4] themselves.

Broader noise model. Our model assumes that the noise of the side-channel is independent and constant. Testing our estimator under a broader model, e.g. dependent noise or time-varying noise would be another angle for future work. The probabilities of single informative relations being correct could also be updated with such a new model. Such an enhanced noise model could equally be incorporated into the models built by Schubert *et al.* [3], Damm *et al.* [8], and Bashiri *et al.* [4] to ensure the attacks continue to work. Also our own preprocessing step can be enhanced by updating the probabilities with the individual noise rates instead of the overall noise rate.

AI Use

We thank Maximilian J. Kramer, Jonas Thietke, Nicolai Kraus and Alexander May for insightful discussions. We used Anthropic's Claude AI tools, especially Claude Code and the Chat function with their Opus 5.0 and Sonnet 4.6 / 5.0 models, to generate code used in our experiments and review our texts, including enhancing suggestions, in order to improve the overall quality and readability. Carsten Schubert is funded by the "Deutsche Forschungsgemeinschaft" (German Research Foundation, DFG) program SPP2514, project number 563143113, as well as by the PQ-CCA project of the German Federal Ministry of Research, Technology and Space ("BMFTR"). Niklas Paskarbeit is also funded by the BMFTR's PQ-CCA project.

References

1. Liu, Y., Zhou, Y., Sun, S., Wang, T., Zhang, R., Ming, J.: On the Security of Lattice-Based Fiat-Shamir Signatures in the Presence of Randomness Leakage. IEEE Transactions on Information Forensics and Security **16**, 1868–1879 (2021). https://doi.org/10.1109/TIFS.2020.3045904
2. Damm, S., Kraus, N., May, A., Nowakowski, J., Thietke, J.: One Bit to Rule Them All – Imperfect Randomness Harms Lattice Signatures. In: Jager, T., Pan, J. (eds.) Public-Key Cryptography – PKC 2025, pp. 284–316. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-91820-9_10
3. Schubert, C., Paskarbeit, N., Seifert, J.-P., Margraf, M.: Descent into Broken Trust: Uncovering ML-DSA Subkeys with Scarce Leakage and Local Optimization. In: Selected Areas in Cryptography – SAC 2026 – To appear. Preproceedings version, Ottawa, Canada (2026). https://sacworkshop.org/SAC26/preproceedings/SAC2026_2-final46.pdf (visited on 08/16/2026)
4. Bashiri, K., Geuenich, J., Mittmann, J.: Exploiting Noisy Single-Bit Leakage in ML-DSA, Cryptology ePrint Archive, Paper 2026/580 (2026). https://eprint.iacr.org/2026/580 (visited on 08/16/2026).
5. National Institute of Standards and Technology, Module-Lattice-Based Digital Signature Standard. Tech. rep. Federal Information Processing Standard (FIPS) 204, U.S. Department of Commerce (2024). https://doi.org/10.6028/NIST.FIPS.204
6. Lyubashevsky, V.: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: Matsui, M. (ed.) Advances in Cryptology – ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_35
7. Lyubashevsky, V.: Lattice Signatures without Trapdoors. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_43
8. Damm, S., Kraus, N., May, A., Nowakowski, J., Thietke, J.: Less Than a Bit to Rule Them All – Key Recovery from Randomness Leakage in ML-DSA, Cryptology ePrint Archive, Paper 2025/820 (2026). https://eprint.iacr.org/2025/820 (visited on 08/16/2026).
9. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
10. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (2018). https://doi.org/10.13154/tches.v2018.i1.238-268. https://tches.iacr.org/index.php/TCHES/article/view/839
11. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with Side Information: Attacks and Concrete Security Estimation. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020, pp. 329–358. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_12
12. Hermelink, J., Streit, S., Mårtensson, E., Petri, R.: A Generic Framework for Side-Channel Attacks Against LWE-Based Cryptosystems. In: Advances in Cryptology – EUROCRYPT 2025: 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4–8, 2025, Pro-

ceedings, Part VIII, pp. 3–32. Springer-Verlag, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-3-031-91101-9_1
13. Islam, S., Mus, K., Singh, R., Schaumont, P., Sunar, B.: Signature Correction Attack on Dilithium Signature Scheme. In: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), pp. 647–663 (2022). https://doi.org/10.1109/EuroSP53844.2022.00046

A Appendix

A.1 Monotonicity Verification

Table 10 shows the numeric evaluation mentioned in Remark 10.

j	regime	β_{eff}	max increase	verdict
ML-DSA-44 ($\eta = 2$, $\tau = 39$, $\beta = 78$)				
3	low	4	-6.72×10^{-5}	OK
4	low	8	-9.91×10^{-4}	OK
5	low	16	-3.40×10^{-4}	OK
6	low	32	-2.42×10^{-6}	OK
7	low	64	-1.09×10^{-16}	OK
8	high	78	0.00×10^{0}	OK
9	high	78	0.00×10^{0}	OK
ML-DSA-65 ($\eta = 4$, $\tau = 49$, $\beta = 196$)				
3	low	4	-1.38×10^{-14}	OK
4	low	8	-6.98×10^{-6}	OK
5	low	16	-1.17×10^{-4}	OK
6	low	32	-4.26×10^{-5}	OK
7	low	64	-3.70×10^{-7}	OK
8	low	128	-1.83×10^{-16}	OK
9	high	196	0.00×10^{0}	OK
ML-DSA-87 ($\eta = 2$, $\tau = 60$, $\beta = 120$)				
3	low	4	-2.35×10^{-6}	OK
4	low	8	-4.38×10^{-4}	OK
5	low	16	-3.12×10^{-4}	OK
6	low	32	-1.30×10^{-5}	OK
7	low	64	-1.73×10^{-11}	OK
8	high	120	0.00×10^{0}	OK
9	high	120	0.00×10^{0}	OK

Table 10: Numerical monotonicity verification of A across ML-DSA parameter sets. All parameter sets report a non-increase of A on $[1, 2\beta_{\text{eff}} - 1]$.

A.2 Noise Estimator

In Table 11, Table 12, and Table 13 we display more information about the accuracy of our noise estimator with the different parameter sets. Also we report the numbers for 25,000 informative relations.

j	α	**MAE**	**STD-E**	**Q50 AE**	**Q90 AE**	**Q95 AE**
4	1000	0.065866	0.082100	0.056722	0.136530	0.157596
	5000	0.033760	0.042791	0.028563	0.068878	0.084832
	25000	0.021968	0.029122	0.016377	0.046341	0.063102
5	1000	0.017357	0.021735	0.014532	0.035596	0.043388
	5000	0.008610	0.010953	0.007247	0.017829	0.021199
	25000	0.005373	0.006920	0.004317	0.011026	0.013469
6	1000	0.009618	0.011986	0.008207	0.019634	0.023288
	5000	0.004411	0.005521	0.003853	0.009017	0.010655
	25000	0.002122	0.002656	0.001774	0.004434	0.005290
7	1000	0.008106	0.010145	0.006956	0.016309	0.019651
	5000	0.003492	0.004363	0.002911	0.007178	0.008280
	25000	0.001528	0.001908	0.001293	0.003080	0.003800
8	1000	0.007579	0.009479	0.006239	0.015467	0.018826
	5000	0.003578	0.004419	0.003022	0.007189	0.008481
	25000	0.001537	0.001917	0.001314	0.003221	0.003801
9	1000	0.007742	0.009589	0.006637	0.015598	0.018504
	5000	0.003573	0.004451	0.002970	0.007419	0.008511
	25000	0.001512	0.001907	0.001241	0.003055	0.00376

Table 11: Noise-Estimator Results for ML-DSA-44. Showing the Leakage index j, the number of used informative relations α as well as for $1,000$ runs the Mean-Absolute-Error (MAE), the Standard-Error (STD-E), the Quantile 50, Quantile 90 and Quantile 95 absolute errors (QXX AE)

A.3 Parallel Attack

Table 14, Table 15, and Table 16 show for each seed and each probability-threshold exactly if the attack was successful. They are therefore more detailed depictions of Table 9.

j	α	**MAE**	**STD-E**	**Q50 AE**	**Q90 AE**	**Q95 AE**
4	1000	9.072461	11.470815	7.455973	18.992841	22.818210
	5000	4.069648	5.106786	3.474138	8.336991	9.823776
	25000	1.841400	2.307446	1.623389	3.740263	4.420320
5	1000	0.071778	0.090665	0.062678	0.153351	0.176095
	5000	0.035265	0.044622	0.028885	0.070918	0.088398
	25000	0.024386	0.032040	0.018707	0.054135	0.064342
6	1000	0.018565	0.023189	0.015688	0.038266	0.044970
	5000	0.008625	0.011044	0.007032	0.018608	0.022374
	25000	0.005207	0.006988	0.003975	0.010854	0.014719
7	1000	0.010002	0.012527	0.008400	0.021377	0.024434
	5000	0.004466	0.005590	0.003730	0.009525	0.010763
	25000	0.002202	0.002785	0.001808	0.004584	0.005334
8	1000	0.007702	0.009630	0.006539	0.015739	0.018753
	5000	0.003442	0.004357	0.002774	0.007076	0.008571
	25000	0.001575	0.001953	0.001332	0.003325	0.003839
9	1000	0.007707	0.009532	0.006487	0.015938	0.018482
	5000	0.003292	0.004151	0.002725	0.006958	0.008208
	25000	0.001493	0.001877	0.001223	0.003082	0.003582

Table 12: Noise-Estimator Results for ML-DSA-65. Showing the Leakage index j, the number of used informative relations α as well as for $1,000$ runs the Mean-Absolute-Error (MAE), the Standard-Error (STD-E), the Quantile 50, Quantile 90 and Quantile 95 absolute errors (QXX AE)

j	α	**MAE**	**STD-E**	**Q50 AE**	**Q90 AE**	**Q95 AE**
4	1000	0.157629	0.194784	0.138889	0.317079	0.376455
	5000	0.071761	0.089943	0.061578	0.147777	0.175485
	25000	0.039945	0.050643	0.033097	0.080145	0.099949
5	1000	0.021678	0.027641	0.017902	0.045378	0.054395
	5000	0.011852	0.014977	0.010019	0.025252	0.029735
	25000	0.007827	0.010285	0.006134	0.016536	0.020740
6	1000	0.010996	0.013769	0.009245	0.022673	0.026226
	5000	0.004926	0.006271	0.004037	0.010469	0.012433
	25000	0.002615	0.003334	0.002162	0.005295	0.006605
7	1000	0.007888	0.009968	0.006273	0.016995	0.019927
	5000	0.003686	0.004567	0.003338	0.007543	0.008798
	25000	0.001691	0.002119	0.001413	0.003549	0.004172
8	1000	0.007276	0.009002	0.006352	0.014725	0.017060
	5000	0.003164	0.003981	0.002708	0.006628	0.007875
	25000	0.001467	0.001839	0.001234	0.003037	0.003574
9	1000	0.007610	0.009562	0.006542	0.015782	0.018827
	5000	0.003293	0.004160	0.002732	0.006925	0.008356
	25000	0.001491	0.001849	0.001290	0.002937	0.003614

Table 13: Noise-Estimator Results for ML-DSA-87. Showing the Leakage index j, the number of used informative relations α as well as for $1,000$ runs the Mean-Absolute-Error (MAE), the Standard-Error (STD-E), the Quantile 50, Quantile 90 and Quantile 95 absolute errors (QXX AE)

seed	0.0	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7	1e-8	1e-9	1e-10	Breakable?
0	0	0	0	0	0	0	0	1	0	0	1	Yes
1	0	0	0	0	0	0	1	0	0	0	0	Yes
2	1	0	0	0	1	1	1	0	1	1	1	Yes
3	0	0	0	1	1	1	1	0	0	0	0	Yes
4	0	0	0	0	0	1	0	1	0	0	1	Yes
5	1	0	0	0	1	1	1	1	1	1	1	Yes
6	1	0	0	1	1	1	1	1	1	1	1	Yes
7	1	0	0	1	0	1	1	0	1	1	1	Yes
8	0	0	0	0	0	0	0	0	0	0	0	No
9	0	0	0	0	0	0	0	0	0	0	0	No
10	0	0	1	0	0	0	1	0	0	0	0	Yes
11	1	0	1	0	1	1	1	1	1	1	1	Yes
12	0	0	0	0	1	1	0	1	1	0	1	Yes
13	1	0	0	1	1	0	1	1	1	1	1	Yes
14	0	0	0	0	0	0	0	0	0	0	0	No
15	1	0	0	1	1	1	1	1	1	1	1	Yes
16	1	0	1	1	1	1	1	1	1	1	1	Yes
17	0	0	0	0	0	0	0	0	0	0	0	No
18	0	0	0	0	0	0	0	0	0	0	0	No
19	1	0	0	1	1	1	1	1	1	1	1	Yes
20	1	0	1	1	1	1	1	1	1	1	1	Yes
21	1	0	0	1	1	1	1	1	1	1	1	Yes
22	0	0	0	0	0	0	0	0	0	0	0	No
23	1	0	1	1	1	1	1	1	1	1	1	Yes
24	1	0	1	1	1	0	1	1	1	1	1	Yes
25	0	0	0	0	0	0	0	0	0	0	0	No
26	1	0	0	1	1	1	1	1	1	1	1	Yes
27	1	0	1	1	1	1	1	1	0	1	1	Yes
28	1	0	0	1	1	1	1	1	1	1	1	Yes
29	0	0	0	0	0	0	1	0	0	0	0	Yes
Total	**16**	**0**	**7**	**14**	**17**	**17**	**20**	**17**	**16**	**16**	**19**	**23**

Table 14: Success for key recovery with varying probability-thresholds in the preprocessing step for $p = 0.45$. The columns represent the different probability-thresholds. A probability-threshold of 0 skips the preprocessing step since all bits have some chance of being correct. The lines represent the seed used for the attack to ensure reproducibility. A 0 denotes that the correct key was not recovered at the specific seed with the denoted probability-threshold, while a 1 denotes a successful recovery. All seeds but 8, 9, 14, 17, 18, 22, and 25 are recovered for some probability-threshold.

seed	0.0	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7	1e-8	1e-9	1e-10	Breakable?
0	0	0	0	0	0	0	0	0	0	0	0	No
1	1	0	1	1	1	1	1	1	1	1	1	Yes
2	1	0	1	1	1	1	1	1	1	1	1	Yes
3	0	0	0	1	0	0	0	1	1	0	0	Yes
4	0	0	0	1	0	0	1	0	0	0	0	Yes
5	1	0	1	1	1	1	1	1	1	1	1	Yes
6	1	0	1	1	1	1	1	1	1	1	1	Yes
7	1	0	0	1	1	1	1	1	1	1	1	Yes
8	1	0	1	1	1	0	1	1	1	1	1	Yes
9	1	0	1	1	1	1	1	1	1	1	1	Yes
10	0	0	0	1	0	0	0	0	0	0	0	Yes
11	1	0	1	1	1	1	1	1	1	1	1	Yes
12	1	0	1	1	1	1	1	1	1	1	1	Yes
13	0	0	0	1	0	0	0	0	0	0	0	Yes
14	1	0	1	1	1	0	0	0	1	1	1	Yes
15	1	0	0	0	1	1	1	1	1	1	1	Yes
16	0	0	1	1	1	0	0	0	0	0	0	Yes
17	1	0	1	1	1	1	1	1	1	1	1	Yes
18	0	0	0	0	0	0	0	0	0	0	0	No
19	1	0	1	1	1	1	1	1	1	1	1	Yes
20	0	0	0	0	0	0	0	0	0	0	0	No
21	0	0	0	0	0	0	0	0	0	0	0	No
22	1	0	1	1	0	1	1	1	1	1	1	Yes
23	0	0	1	0	0	0	0	0	0	0	0	Yes
24	1	0	1	1	1	1	1	1	1	1	1	Yes
25	1	0	1	1	1	1	1	1	1	1	1	Yes
26	0	0	0	0	0	0	0	0	0	0	0	No
27	1	0	1	1	1	1	1	1	1	1	1	Yes
28	1	0	0	0	1	0	0	1	1	1	1	Yes
29	0	0	0	1	0	0	0	0	0	0	0	Yes
Total	18	0	17	22	18	15	17	18	**19**	**18**	**18**	**25**

Table 15: Success for key recovery with varying probability-thresholds in the preprocessing step for $p = 0.4$. The columns represent the different probability-thresholds. A probability-threshold of 0 skips the preprocessing step since all bits have some chance of being correct. The lines represent the seed used for the attack to ensure reproducibility. A 0 denotes that the correct key was not recovered at the specific seed with the denoted probability-threshold, while a 1 denotes a successful recovery. All seeds but 0, 18, 20, 21, and 26 are recovered for some probability-threshold.

seed	0.0	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7	1e-8	1e-9	1e-10	Breakable?
0	0	0	0	0	1	0	1	0	0	0	0	Yes
1	1	0	0	0	1	1	0	0	1	1	1	Yes
2	1	0	1	1	1	1	1	1	1	1	1	Yes
3	1	1	1	1	1	1	1	1	1	1	1	Yes
4	1	0	0	0	0	1	1	0	1	1	1	Yes
5	1	1	1	1	1	1	1	1	1	1	1	Yes
6	0	0	0	0	0	0	0	0	0	0	0	No
7	0	0	0	1	1	0	0	0	0	0	0	Yes
8	1	0	1	1	1	1	1	1	1	1	1	Yes
9	0	0	0	1	0	0	0	0	0	0	0	Yes
10	1	0	1	1	1	1	1	1	1	1	1	Yes
11	1	0	1	1	1	1	1	1	1	1	1	Yes
12	0	0	1	1	0	0	1	1	0	0	0	Yes
13	0	0	0	0	0	0	1	0	0	1	0	Yes
14	0	0	0	0	0	1	0	0	0	0	0	Yes
15	1	1	1	1	1	1	1	1	1	1	1	Yes
16	1	1	1	1	1	1	1	1	1	1	1	Yes
17	1	1	1	1	1	1	1	1	1	1	1	Yes
18	0	0	1	1	0	1	1	0	1	1	0	Yes
19	0	0	0	0	1	1	0	0	1	1	0	Yes
20	1	0	1	1	1	1	1	1	1	1	1	Yes
21	1	0	0	1	1	1	1	1	1	1	1	Yes
22	0	0	0	1	0	1	0	0	1	0	0	Yes
23	1	1	1	1	1	1	1	1	1	1	1	Yes
24	1	1	1	1	1	1	1	1	1	1	1	Yes
25	1	1	1	1	1	1	1	1	1	1	1	Yes
26	1	0	1	1	1	1	1	1	1	1	1	Yes
27	1	1	1	1	1	1	1	1	1	1	1	Yes
28	1	0	0	0	0	0	1	1	1	1	1	Yes
29	0	0	1	0	0	1	0	0	0	0	1	Yes
Total	**19**	**9**	**18**	**21**	**20**	**23**	**22**	**18**	**22**	**22**	**20**	**29**

Table 16: Success for key recovery with varying probability-thresholds in the preprocessing step for $p = 0.2$. The columns represent the different probability-thresholds. A probability-threshold of 0 skips the preprocessing step since all bits have some chance of being correct. The lines represent the seed used for the attack to ensure reproducibility. A 0 denotes that the correct key was not recovered at the specific seed with the denoted probability-threshold, while a 1 denotes a successful recovery. All seeds but 6 are recovered for some probability-threshold.