Descent into Broken Trust: Uncovering ML-DSA Subkeys with Scarce Leakage and Local Optimization

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

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

Abstract. ML-DSA (formerly CRYSTALS-Dilithium), the primary NIST post-quantum signature standard, relies on rejection sampling to ensure that released signatures are statistically independent of the secret key. Recent work by Liu et al. and Damm et al. showed that this protection breaks down as soon as an attacker obtains even a single bit of the generated masking randomness per signature, enabling key recovery via linear regression over the resulting noisy linear system. However, this regression approach requires the attacker to collect a large number of such leaky signatures—up to 2.4 million so-called informative relations for ML-DSA-65—thereby limiting the attack's practical applicability.

We dramatically reduce this cost by reformulating the key recovery as a constraint-satisfaction problem solvable by local optimization. Our approach rests on two contributions. First, we provide a verification routine that checks individual candidate subkeys only using the leakage relations collected. Second, building on theoretical insights from this verification method, we design a multi-tier hill-climbing algorithm that iteratively refines candidates by minimizing a scoring function.

In the exact leakage setting, our attack recovers ML-DSA subkeys from as few as 1 500 to 35 000 informative relations across all parameter sets and tested leakage bit indices, constituting a reduction by a factor of 37–68× over the previous state of the art and allowing a recovery for leakage bit indices as low as 4 for ML-DSA-44 and ML-DSA-87 and 5 for ML-DSA-65. We further extend the attack to a noisy leakage model, where each leaked bit is flipped with some probability p. We demonstrate that key recovery even remains practical at noise rates as high as $p = 45\%$, again with substantially less leakage information than prior work.

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

1 Introduction

ML-DSA [1], standardized by the National Institute of Standards and Technology (NIST) as the primary post-quantum digital signature scheme, derives its security from the hardness of the Module Learning with Errors problem. A central design element is rejection sampling in the Fiat-Shamir-with-aborts paradigm introduced by Lyubashevsky [2, 3]: Every released signature is statistically independent of the secret key, because the signer discards any output whose distribution would betray the signing randomness. This guarantee, however, rests on the assumption that the masking randomness $\mathbf{y}$ used during signing remains entirely hidden from an adversary.

Liu *et al.* [4] demonstrated that this assumption is more fragile than it appears: Access to a single leaked bit of $\mathbf{y}$ per signature suffices, in principle, to mount a key-recovery attack. Damm *et al.* [5] subsequently refined the approach into a practical regression-based attack applicable to all three ML-DSA parameter sets and across a wide range of leakage bit positions. Yet their method remains data-hungry: Depending on the parameter set, between 500 000 and 2 400 000 informative relations, and correspondingly many more raw signatures[3], are needed to reliably recover a single subkey at high leakage indices. This large signature requirement constitutes the main practical barrier to exploiting the attack, and closing the gap between the theoretical threat ("one bit per signature is enough") and a realistic attack scenario is the central motivation for our work.

We address this gap by replacing the linear-regression step with an iterative local-optimization procedure. Two ingredients make this possible. First, we develop a lightweight *verification routine* that decides, using only the informative relations pertaining to a single module component of the secret key, whether a candidate subkey is correct, without requiring knowledge of the remaining components or performing lattice reduction (Section 3). This routine yields a natural scoring function whose theoretical properties (monotonicity, convexity) we establish formally. Second, we build a multi-tier *hill-climbing algorithm* that minimizes this score over the coefficient space of the subkey, employing adaptive block sizes, lateral moves, and perturbation-based restarts to navigate the search landscape reliably (Section 4). In the exact leakage setting, these techniques reduce the number of required informative relations by a factor of 37–68× compared to the regression attack of Damm *et al.* [5], depending on the parameter set. We further extend the attack to the *noisy leakage model* that Damm *et al.* [6] introduced in their latest paper revision, i.e. where the leaked bits are independently flipped with some error probability p (Section 5). We analyze how noise affects all relevant leakage index regimes, show that our excess-based scoring function becomes unreliable and must be replaced by count-based scoring in the presence of noise, and demonstrate experimentally that the attack remains viable even at substantial noise rates—again requiring significantly fewer signatures than prior work.

[3] Depending on the leakage bit index. At low indices every signature is informative. For more information see Definition 2.

1.1 Related Work

Our work builds directly on Liu *et al.* [4] and Damm *et al.* [5] as described in Section 2. In this subsection we sketch the surrounding literature to provide a broader context for our contributions.

Passive side-channel attacks. Ulitzsch *et al.* [7] presented the first end-to-end key recovery attack on Dilithium. They used a profiling-based power analysis attack on the Cortex-M4 implementation to recover zero-coefficients of $\mathbf{y}$ in order to solve for $\mathbf{s}_1$ using linear algebra. Their attack therefore showed the possibility of recovering coefficients of $\mathbf{y}$. Steffen *et al.* [8] complemented this with hardware-focused work and Qiao *et al.* [9] showed that leakage based on the NTT algorithm can expose bits of $\mathbf{y}$ even in masked settings. Bronchain *et al.* [10] further demonstrated that noisy knowledge of the coefficients of $\mathbf{y}$ can be accumulated across multiple signatures via belief propagation, with the approach remaining effective even against shuffled implementations of Dilithium.

Fault attacks on ML-DSA. Fault attacks offer another possibility to acquire knowledge about the used randomness. Espitau *et al.* [11] created fault attacks on lattice based signature schemes and proposed countermeasures. Ulitzsch *et al.* [12], demonstrated that these loop-abort countermeasures can be defeated by injecting faults into the sampling of $\mathbf{y}$ that zero out individual coefficients, and that the resulting over-determined linear system can be solved via ILP to achieve key recovery. This attack was extended by Krahmer *et al.* [13] to target the NIST selected standard of randomized/hedged Dilithium.

Masked implementations. Migliore *et al.* [14] established the first efficient masked implementation of Dilithium and evaluated its side-channel resistance, building on the foundational probing-model security framework of Ishai, Sahai, and Wagner [15] and the higher-order masking techniques of Rivain and Prouff [16]. Their work was further refined by following papers such as [17, 18]. A survey on masking was done by Covic, Ganji, and Forte [19]. More recently Hermelink, Ning, and Petri [20] analyzed the security of masked ML-DSA, specifically targeting $\mathbf{y}$.

Concealed ILWE. Damm *et al.* [21] introduced a different leakage scenario: Instead of leaking single bits of $\mathbf{y}$, the leakage reveals whether a coefficient of $\mathbf{y}$ is zero. They validate their approach with a profiling attack on a masked Dilithium implementation where this classification leakage is present.

Generic belief-propagation frameworks for LWE-based side-channel attacks. A parallel line of work asks how to recover LWE secrets from side-channel information more generally. Hermelink *et al.* [22] introduced belief propagation(BP) as a tool for this purpose, demonstrating that BP consistently requires fewer hints than greedy solvers for recovering secrets from decryption-error inequalities in the ML-KEM context, at the cost of greater computational effort and occasional numerical stability issues. Building on this, Hermelink *et al.* [23] proposed a generic distribution-hint framework that unifies most previously defined hint types and formally proves that greedy algorithms used in prior attacks are computationally cheaper approximations of BP. We elaborate more on this in Section 7.

1.2 Outline

Section 1.3 fixes notation. Section 2 reviews the ML-DSA scheme and the leakage-based attacks of Liu *et al.* [4] and Damm *et al.* [5]. Section 3 develops the subkey verification routine. Section 4 presents the hill-climbing key-recovery algorithm. Section 5 extends the analysis and the algorithm to noisy leakage. Section 6 reports experimental results for both the exact and the noisy setting. Section 7 discusses implications and open directions.

1.3 Preliminaries

We denote vectors by bold lowercase letters (e.g. $\mathbf{v}$) and matrices by bold uppercase letters (e.g. $\mathbf{A}$). The standard inner product of two vectors of equal dimension is written $\langle \mathbf{a}, \mathbf{b} \rangle := \sum_i a_i b_i$, and $\|\mathbf{v}\|_\infty := \max_i |v_i|$ denotes the infinity norm.

For a finite set S, we write $x \xleftarrow{\$} S$ for uniformly random sampling from S, and $x \xleftarrow{\$} D$ for sampling from a distribution D. The notation $\xleftarrow{\$} S_m$ denotes uniform sampling from a set S restricted to elements with $\|\mathbf{a}\|_\infty \leq m$. We write $\mathrm{Ber}(p)$ for the Bernoulli distribution with success probability p.

We use standard interval notation for integers: $[a, b]$ denotes the integer interval $\{a, a+1, \ldots, b\}$ unless the real-valued interval is clear from context, $[n]$ is shorthand for $[1, n]$, and $[\pm n]$ abbreviates $[-n, n]$. The set $[\pm 1]^n_\tau \subset \{-1, 0, 1\}^n$ consists of all vectors with exactly τ nonzero entries. 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 $[-m/2,\ m/2)$.

Finally, $\mathbb{E}_X[\cdot]$ denotes the expectation taken over the randomness of the random variable X while all other random variables are held fixed, and $\mathrm{sign}(x)$ returns $+1$ if $x \geq 0$ and -1 if $x < 0$.

2 Attack Setting and Review of the Attack of Damm *et al.*

Our attack aims for a (partial) key recovery of an ML-DSA key using leakage information. We therefore use the same attack setting as Damm *et al.* [5]. In this section we review this setting and their attack.

2.1 ML-DSA

ML-DSA is a lattice-based digital signature scheme following the Fiat-Shamir-with-aborts paradigm introduced by Lyubashevsky [2, 3]. The central idea is to combine the Fiat-Shamir transform [24] with rejection sampling, ensuring that released signatures do not leak information about the secret key. The scheme was proposed by Ducas *et al.* [25] and subsequently standardized by the NIST [1] as part of their post-quantum cryptography standardization project. Its security relies on the hardness of the Module Learning with Errors (Module-LWE) problem [3].

Scheme overview. The full ML-DSA algorithms are given in Figure 1; we summarize the aspects relevant to the leakage attack below. All algebraic operations take place in the polynomial ring $R \coloneqq \mathbb{Z}_q[x]/(x^n+1)$, where an element of R can equivalently be viewed as an n-dimensional integer vector of its coefficients.

Key generation. The signer samples a public matrix $\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}$ and two secret vectors $\mathbf{s}_1 \xleftarrow{\$} R_\eta^\ell$, $\mathbf{s}_2 \xleftarrow{\$} R_\eta^k$, where every coefficient of $\mathbf{s}_1$ and $\mathbf{s}_2$ is bounded by η in absolute value. The public key is $(\mathbf{A},\ \mathbf{t} \coloneqq \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$.

Signing. To sign a message, the signer repeatedly:

1. samples masking randomness $\mathbf{y} \in R^\ell$ with coefficients drawn uniformly from $\{-(\gamma_1-1), \ldots, \gamma_1-1\}$,
2. derives a challenge polynomial $\mathbf{c} \in R$ with exactly τ coefficients equal to ± 1 and the rest zero,
3. computes the signature vector $\mathbf{z} \coloneqq \mathbf{y} + \mathbf{c}\mathbf{s}_1$.

The signature $(\mathbf{z}, \mathbf{h}, \mathbf{c})$ is released only if the rejection condition $\|\mathbf{z}\|_\infty \le \gamma_1 - \beta$ is satisfied, where $\beta = \eta \cdot \tau$ is an upper bound on $\|\mathbf{c}\mathbf{s}_1\|_\infty$. Otherwise the signer aborts and restarts with fresh randomness.

Role of rejection sampling. The rejection condition is the mechanism that protects the secret key: It ensures that every coefficient z of the released signature vector $\mathbf{z}$ is uniformly distributed on $[-(\gamma_1-\beta),\ \gamma_1-\beta]$, regardless of the secret key[4]. Since the attack operates at the level of individual polynomial coefficients, we note that 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 $\mathbf{c}$; it retains exactly τ nonzero entries. For notational simplicity, following [5], we henceforth 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, where $\mathbf{x}$ is the coefficient vector of one component polynomial of $\mathbf{s_1}$ and y the corresponding coefficient of $\mathbf{y}$. Without rejection, z would be the uniform law of y shifted by the secret-dependent $\langle \mathbf{c}, \mathbf{x}\rangle$; rejection retains only its flat central portion, making z seemingly independent of $\mathbf{x}$. As we show next, this guarantee breaks down when the attacker obtains additional leakage information about the masking randomness $\mathbf{y}$.

Parameter sets. NIST standardized three ML-DSA parameter sets [1], targeting security categories 2, 3, and 5. The parameters relevant to the leakage attack are listed in Table 1. In all variants, the ring dimension is $n = 256$.

2.2 Leakage-Based Attack by Liu *et al.* and Damm *et al.*

We now review the leakage-based attacks of Liu *et al.* [4] and Damm *et al.* [5] against ML-DSA, which form the basis of our study. Since Damm *et al.* [5] already improved upon the previous work, we primarily adapt their formulation. We start

[4] Strictly, FIPS 204 accepts signatures satisfying $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$ (open interval). Following [5], we use the closed interval $[-(\gamma_1-\beta),\ \gamma_1-\beta]$ throughout, as the difference of a single integer value is negligible.

ML-DSA (CRYSTALS-Dilithium)

$\texttt{setup}(1^\kappa) \to (pk, sk)$:

1. Sample $\mathbf{A} \xleftarrow{\$} R^{k\times\ell}$ with $R \coloneqq \mathbb{Z}_q[x]/(x^n+1)$
2. Sample two small secret vectors $\mathbf{s}_1 \xleftarrow{\$} R_\eta^\ell,\ \mathbf{s}_2 \xleftarrow{\$} R_\eta^k$
3. Compute $\mathbf{t} \coloneqq \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k$
4. Output: pk $= (\mathbf{A}, \mathbf{t})$, sk $= (\mathbf{s}_1, \mathbf{s}_2)$

$\texttt{sign}(m, sk, pk) \to (\mathbf{z}, \mathbf{h}, \mathbf{c})$:
Repeat until output:

1. Sample $\mathbf{y} \xleftarrow{\$} R_{\gamma_1}^\ell$
2. Compute $\mathbf{w} \coloneqq \mathbf{A}\mathbf{y} \in R_q^k$
3. Split $\mathbf{w}_1 = \text{HighBits}(\mathbf{w})$ and $\mathbf{w}_0 = \text{LowBits}(\mathbf{w})$
4. Compute $\mathbf{c} \coloneqq \mathcal{H}(m, \mathbf{w}_1)$
5. Compute $\mathbf{z} \coloneqq \mathbf{y} + \mathbf{c}\mathbf{s}_1$
6. If $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ restart
7. Compute $\mathbf{h} \coloneqq \text{MakeHint}(\mathbf{w}_0 - \mathbf{c}\mathbf{s}_2,\ \mathbf{w}_1)$
8. If $|\mathbf{h}| < \omega$: Output signature $(\mathbf{z}, \mathbf{h}, \mathbf{c})$

$\texttt{verify}(pk, m, (\mathbf{z}, \mathbf{h}, \mathbf{c})) \to$ accept or reject:

1. Check validity of $\mathbf{z}, \mathbf{h}, \mathbf{c}$
2. Compute $\mathbf{w}' \coloneqq \mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} \in R^k$
3. Recompute high bits using hint: $\mathbf{w}_1' \coloneqq \text{UseHint}(\mathbf{h}, \mathbf{w}')$
4. Compute $\mathbf{c}' = \mathcal{H}(m, \mathbf{w}_1')$
5. If $\mathbf{c}' = \mathbf{c}$ accept, else reject

Fig. 1: ML-DSA (Dilithium) as proposed by Ducas *et al.* [25] and standardized by NIST [1]. The MakeHint and UseHint functions serve only to compress the signature and are not relevant to the leakage attack; see [25] for details.

by formalizing the problem of recovering a secret key component from leakage information obtained during signing.

Definition 1 (Leaky-Signature-LWE Problem). *Let* $(\mathbf{A}, \mathbf{t}) \in R_q^{k\times\ell} \times R_q^k$ *be an ML-DSA public key with secret key* $(\mathbf{s}_1, \mathbf{s}_2)$ *satisfying* $\|\mathbf{s}_1\|_\infty, \|\mathbf{s}_2\|_\infty \leq \eta$. *Fix one entry of* $\mathbf{s_1}$ *with coefficient vector* $\mathbf{x} \in \{-\eta, \ldots, \eta\}^n$, *one coefficient position of it, and a bit index* j, *using the per-coefficient relation* $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$ *of Section 2.1. The attacker is given* $(\mathbf{A}, \mathbf{t})$ *and arbitrarily many tuples* $(\mathbf{c}, z, y_j)$; *one tuple per released signature, all at the fixed position with* y_j *as the* j*-th bit in the binary two's complement representation of* y. *Their goal is to recover* $\mathbf{x}$.

Note that $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \eta \cdot \tau = \beta$ since $\mathbf{c}$ has τ entries of ± 1 and $\|\mathbf{x}\|_\infty \leq \eta$. Technically, in order to subsequently rebuild the full key $\mathbf{s_1}$ based on $\mathbf{x}$, an attacker would additionally need to know the position of $\mathbf{x}$ within the secret

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
γ_1 (randomness range)	2^{17}	2^{19}	2^{19}
$\beta = \tau \cdot \eta$ (norm bound)	78	196	120
Security category	2	3	5

Table 1: ML-DSA parameter sets relevant to the leakage attack [1]. Parameters not directly used in the attack (d, γ_2, λ, ω, ζ) are omitted; see [1] for the complete specification.

key $\mathbf{s_1}$. However, for the purposes of our paper this information is irrelevant since we solely focus on recovering $\mathbf{x}$. Additionally, the success probability for reconstructing $\mathbf{s_1}$ from $\mathbf{x}$ (as mentioned below) does not depend on the position.

Remark 1. Definition 1 models the leakage information abstractly as access to a fixed bit y_j of the masking randomness. The formulation is thus agnostic to this information's physical origin (see Section 1.1 for recommended operational research).

From leakage to LWE relations. The key idea, introduced by Liu *et al.* [4], is to use the leaked bit y_j to extract information from the signature component z that would otherwise be hidden by the masking randomness. Specifically, using the bit y_j and the public value z, one computes a transformed value

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}, \tag{1}$$

where $[y]_{2^j}$ denotes the reduction of y modulo 2^j (centered). This yields an Integer LWE instance with error $[y]_{2^j}$, which is uniformly distributed over $[-2^{j-1}, 2^{j-1} - 1]$. We regard this relation extraction procedure as a black box and refer to Damm *et al.* [5], chapter 5, for the algorithmic details.

A crucial observation by Damm *et al.* [5] is that not all extracted relations are useful. Those with $|\overline{z}| < 2^{j-1} - \beta$ carry no information about $\mathbf{x}$, since the inner product $\langle \mathbf{c}, \mathbf{x} \rangle$ is completely masked by the error.

Definition 2 (Informative relation). *Following Damm* et al. *[5], a relation* $(\mathbf{c}, z, y_j)$ *is called* informative *if* $|\overline{z}| \geq 2^{j-1} - \beta$.

The j-independence transformation. Damm *et al.* [5] then introduce a transformation that, in the *high-leakage regime* ($2^{j-1} > \beta$), maps all informative relations to a canonical form with error independent of j. Specifically, they compute

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

and obtain $\tilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \tilde{y}$ for some error $\tilde{y}$ uniformly distributed over $[-\beta, \beta]$.

Theorem 1 (Theorem 3 in [5], restated). *If $2^{j-1} > \beta$ and $(\mathbf{c}, \overline{z}, y_j)$ is an informative relation, then the transformation (2) yields $\tilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \tilde{y}$ where $\tilde{y} \xleftarrow{\$} [\pm\beta]$.*

This result is meaningful as it makes the error size independent of the leakage index j when $2^{j-1} > \beta$. We further note that the relation extraction (1) itself requires a minimum leakage index: Damm *et al.* [5] formally require $j \geq \lceil \log_2(\beta) \rceil + 1$ for their extraction to be exact, but show that this is a worst-case condition and the attack succeeds in practice for all $j \geq 6$ across ML-DSA parameter sets, since $\langle \mathbf{c}, \mathbf{x} \rangle$ concentrates well below β by the central limit theorem. We refer to [5] for the detailed analysis.

Generalized error bound. Theorem 1 requires $2^{j-1} > \beta$ and thus only covers the high-leakage regime. In the *low-leakage regime* where $2^{j-1} \leq \beta$, the transformation (2) is not applicable. However, this situation is in fact *more favorable* for the attacker: The resulting LWE error $[y]_{2^j}$ is then uniform on $[-2^{j-1}, 2^{j-1} - 1]$, that is, a *smaller* interval than $[\pm\beta]$. Moreover, since $2^{j-1} \leq \beta$, the informativeness condition $|\overline{z}| \geq 2^{j-1} - \beta$ is trivially satisfied by all relations, meaning no leaked signature is discarded.

We can unify both regimes by defining the *effective error bound*

$$\beta_{\text{eff}} = \min\left(\beta,\ 2^{j-1}\right). \tag{3}$$

In both cases, the attacker obtains relations of the form $\tilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \tilde{y}$ with $\tilde{y}$ uniform on $[\pm\beta_{\text{eff}}]$: In the high-leakage regime via the transformation of Theorem 1, and in the low-leakage regime directly from the extraction (1). The smaller error is one of the factors that enable improved key recovery with fewer relations in the low-leakage regime, as we later exploit in our attack (Section 4).

Key recovery via linear regression. Given sufficiently many informative relations, the transformed system $\tilde{z}_i = \langle \mathbf{c}_i, \mathbf{x} \rangle + \tilde{y}_i$ is a noisy linear system that can be solved for $\mathbf{x}$ via ordinary least-squares regression followed by rounding to the nearest integer in $\{-\eta, \ldots, \eta\}$. While polynomial in complexity, this regression approach requires a large number of informative relations to overcome the masking randomness. Involving hundreds of thousands of informative relations for the ML-DSA parameter sets considered by Damm *et al.* [5], this constitutes a practical limitation.

From subkey to full key recovery. The procedure described recovers a single coefficient polynomial $\mathbf{x}$ of the secret key $\mathbf{s_1} \in R^{\ell}$. Damm *et al.* [5] describe in their Section 4.3 how to recover the full key $\mathbf{s}_1$ from $\mathbf{x}$ using the lattice reduction with side information framework of Dachman-Soled *et al.* [26, 27] and May and Nowakowski [28]. Since this step is unchanged in our work, we refer to the respective literature for details.

Limitations and motivation for our improvements. The regression-based attack of Damm *et al.* [5] is highly effective when sufficient leakage is available: It runs in

polynomial time, applies to all ML-DSA parameter sets, and handles all leakage bit positions $j \geq 6$. However, the number of informative relations required for reliable key recovery is substantial: Even for the easiest variant (ML-DSA 44), they report 0.5 million needed informative relations in the higher leakage index regimes. Moreover, the fraction of informative relations among all released signatures decreases exponentially with the leakage index j, since the non-informative range $[\pm(2^{j-1} - \beta)]$ covers a growing share of the support of z. As a result, the total number of signatures required grows exponentially in j even though the number of *informative* relations stays roughly constant [5].

Our work addresses this limitation by replacing the linear regression with an iterative optimization procedure. As we demonstrate in subsequent sections (see Section 6 in particular), this approach reduces the number of required informative relations by a factor of up to 68× while maintaining practical attack speeds.

3 ML-DSA Subkey Verification Routine

Before presenting the improved key-recovery algorithm itself, we develop a self-contained verification subroutine that plays a dual role in what follows: Applied in isolation, it lets the attacker decide correctness of a candidate subkey $\mathbf{x}'$ without assembling the entire secret key or invoking lattice reduction, thereby enabling them to mount more efficient attacks e.g. in the ML-DSA attack model of Liu *et al.* [4] where leakage bits are collected independently for all ℓ subkeys. Moreover, this routine also crystallizes the theoretical insights that will later drive the hill-climbing optimizer in Section 4. We therefore state the result in full generality.

Damm *et al.* [5] left the verification question open: In their pipeline, correctness is inferred only globally, once the full key $\mathbf{s_1}$ has been assembled and tested against the public key. We close this gap by observing that the informative relations that are already collected and transformed (cf. Section 2) can themselves be repurposed as a verification oracle for the corresponding subkey. The key insight, formalized in Theorem 2, is that a candidate $\mathbf{x}'$ satisfies all relations associated with the true subkey $\mathbf{x}$ if and only if $\mathbf{x}' = \mathbf{x}$, provided sufficiently many relations are available. This results in a lightweight and self-contained check, not requiring any knowledge of the remaining subkeys.

Theorem 2. *Given α random informative relations in a* Leaky-Signature-LWE *setting with $\ell > 1$, one can check in linear time whether a given candidate $\mathbf{x}^*$ is equal to $\mathbf{x}$ with a success probability of at least $1 - \left(1 - \left(2n\eta + \frac{n}{\tau}\right)^{-1}\right)^{\alpha}$, without having access to any additional information about $\mathbf{s_1}$.*

The proof of Theorem 2 requires some intermediate results, most notably, Algorithm 1 and two lemmas (Lemma 1 and Lemma 2). We start by developing the former, which in turn relies on the related main result of [5] :

Theorem 3 (Theorem 3 in [5], restated). *If $2^{j-1} > \beta$ and $(\mathbf{c}, \overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + y, y_j)$ is an* informative relation *(see Definition 2) in the* Leaky-Signature-

LWE problem, then it can be transformed into $(\mathbf{c}, \widetilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y})$ *where* $\widetilde{y} \xleftarrow{\$} [\pm\beta]$. *To do so, let* $\widetilde{z} = \overline{z} - 2^{j-1} + \beta$ *if* $\overline{z} > 2^{j-1} - \beta$, *and* $\widetilde{z} = \overline{z} + 2^{j-1} - \beta$ *otherwise.*

Using the relations produced by Theorem 3 – and their low-leakage analogues as described in Equation (3) – we construct a probabilistic verification routine for $\mathbf{x}$ with a one-sided error, depicted in Algorithm 1.

Algorithm 1: ML-DSA subkey verification step

Input: Preprocessed informative relation $(\mathbf{c}, \widetilde{z})$, effective bound β_{eff}, leakage index j, verification candidate $\mathbf{x}^*$
Output: *False* if $\mathbf{x}^* \neq \mathbf{x}$ with positive probability, *True* otherwise

1 $\delta \leftarrow \langle \mathbf{c}, \mathbf{x}^* \rangle - \widetilde{z}$
2 **if** $2^{j-1} \leq \beta$ **then** $\delta \leftarrow [\delta]_{2^{j+1}}$ `/* centered reduction mod` $M = 2^{j+1}$ `*/`
3 **if** $|\delta| > \beta_{\text{eff}}$ **then return** *False*
4 **return** *True*

Regime-dependent constraint checking. Algorithm 1 checks whether the residual δ between the candidate's inner product $\langle \mathbf{c}, \mathbf{x}^* \rangle$ and the preprocessed relation value $\widetilde{z}$ exceeds the effective error bound $\beta_{\text{eff}} = \min\left(\beta, 2^{j-1}\right)$ (cf. Equation (3)). In the *high-leakage regime* $(2^{j-1} > \beta)$, the j-independence transformation of Theorem 3 yields relations with error $\widetilde{y} \xleftarrow{\$} [\pm\beta]$ over the integers, so $\beta_{\text{eff}} = \beta$ and the residual $\delta = \langle \mathbf{c}, \mathbf{x}^* \rangle - \widetilde{z}$ requires no further processing. In the *low-leakage regime* $(2^{j-1} \leq \beta)$, the transformation is not applied and the error is the smaller quantity $[y]_{2^j} \in [-2^{j-1}, 2^{j-1} - 1]$, yielding $\beta_{\text{eff}} = 2^{j-1} < \beta$. However, since the relation extraction (1) in this regime involves a centered reduction modulo $M \coloneqq 2^{j+1}$, the relation $\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [\, y \,]_{2^j}$ holds modulo M rather than over the integers. The residual must therefore be centered-reduced modulo M before comparison, as reflected in Line 2 of Algorithm 1.

Correctness of Algorithm 1. If $\mathbf{x}^*$ equals the true key $\mathbf{x}$, then $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{c}, \mathbf{x} \rangle$ for all $\mathbf{c}$, so the residual (before or after any modular reduction) equals $\widetilde{y}$. Since $|\widetilde{y}| \leq \beta_{\text{eff}}$, the algorithm always returns *True*.

Given a false candidate $\mathbf{x}^* \neq \mathbf{x}$, the algorithm will still output *True* if $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{c}, \mathbf{x} \rangle$ (which happens when $\mathbf{c}$ places no nonzero entry on a differing position). However, when $\mathbf{x}^* \neq \mathbf{x}$, we have $\langle \mathbf{c}, \mathbf{x}^* \rangle \neq \langle \mathbf{c}, \mathbf{x} \rangle$ with positive probability, since $\mathbf{c}$ randomly *selects* and *signs* exactly τ positions. Additionally, for the algorithm to output *False*, $\widetilde{y}$ must satisfy $|\delta| > \beta_{\text{eff}}$. Since $\widetilde{y} = \pm\beta_{\text{eff}}$ with probability $\frac{1}{2\beta_{\text{eff}}+1}$ each, this happens with positive probability (at least $\frac{1}{2\beta_{\text{eff}}+1}$) whenever $\langle \mathbf{c}, \mathbf{x}^* \rangle \neq \langle \mathbf{c}, \mathbf{x} \rangle$. □

Monotonicity of detection. The following lemma establishes that correcting an erroneous position in a candidate subkey can only *decrease* the detection probability. We state and prove the result for the error bound β (i.e. the high-leakage

regime where $\beta_{\text{eff}} = \beta$), since in that regime the per-relation detection function is genuinely convex and the proof rests on a clean application of Jensen's inequality. Proposition 1 below extends the conclusion to the low-leakage regime where $\beta_{\text{eff}} < \beta$.

Lemma 1. *Let $\mathbf{x}^*$ be a candidate subkey that differs from the correct subkey $\mathbf{x}$. Let $\mathbf{x}^{**}$ be another candidate obtained from $\mathbf{x}^*$ by correcting one of those positions, i.e. by setting $x_i^{**} = x_i$ for some $i \in S \coloneqq \{i \mid x_i^* \neq x_i\}$ and having $x_k^{**} = x_k^*$ for all $k \neq i$. Then, when the verification of Algorithm 1 is performed with error bound β, the probability of receiving a False output for $\mathbf{x}^*$ is at least as large as for $\mathbf{x}^{**}$.*

Proof. Let $\mathbf{e}' \coloneqq \mathbf{x}^* - \mathbf{x}$ and write $e'_k \coloneqq x_k^* - x_k$ for its components. Define $\epsilon \coloneqq \langle \mathbf{c}, \mathbf{e}' \rangle = \langle \mathbf{c}, \mathbf{x}^* \rangle - \langle \mathbf{c}, \mathbf{x} \rangle$. From

$$\begin{aligned} & \langle \mathbf{c}, \mathbf{x}^* \rangle \notin [\widetilde{z} - \beta, \widetilde{z} + \beta] \\ \Leftrightarrow\, & \langle \mathbf{c}, \mathbf{x}^* \rangle \notin [\langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y} - \beta, \langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y} + \beta] \\ \Leftrightarrow\, & \langle \mathbf{c}, \mathbf{x}^* \rangle - \langle \mathbf{c}, \mathbf{x} \rangle - \widetilde{y} \notin [-\beta, \beta] \\ \Leftrightarrow\, & |\epsilon - \widetilde{y}| > \beta, \end{aligned}$$

we obtain the detection probability $p_{\text{detect}}(\mathbf{x}^*) \coloneqq \Pr[|\epsilon - \widetilde{y}| > \beta]$ with randomness drawn from $\mathbf{c}$ and $\widetilde{y}$. With $\widetilde{y} \xleftarrow{\$} [\pm\beta]$ the complementary probability is

$$\begin{aligned} p_{\text{miss}} =& \Pr[|\epsilon - \widetilde{y}| \leq \beta] = |\{v \in [-\beta, \beta] : |\epsilon - v| \leq \beta\}| \cdot \frac{1}{2\beta + 1} \\ =& |[-\beta, \beta] \cap [\epsilon - \beta, \epsilon + \beta]| \cdot \frac{1}{2\beta + 1} \end{aligned}$$

i.e. the overlap length of the intervals $[-\beta, \beta]$ and $[\epsilon - \beta, \epsilon + \beta]$, normalized by $\frac{1}{2\beta+1}$. Since the second interval is just the first one shifted by ϵ, the length of the overlap is $2\beta + 1 - |\epsilon|$.[5]

Therefore we get

$$p_{\text{miss}}(\mathbf{x}^*) = 1 - \frac{|\epsilon|}{2\beta + 1} \Rightarrow p_{\text{detect}}(\mathbf{x}^*) = \frac{|\epsilon|}{2\beta + 1} =: f(\epsilon).$$

Note that f is a convex function of ϵ on its entire domain $|\epsilon| \leq 2\beta$.

Since for any fixed $\mathbf{c}$ the detection probability (over the randomness of $\widetilde{y}$) is $f(\langle \mathbf{c}, \mathbf{e}' \rangle)$, we have

$$p_{\text{detect}}(\mathbf{x}^*) = \mathbb{E}_{\mathbf{c}}[f(\langle \mathbf{c}, \mathbf{e}' \rangle)]\,. \tag{4}$$

By the same reasoning, as $\mathbf{x}^{**} - \mathbf{x}$ agrees with $\mathbf{e}'$ everywhere except that position i is zeroed out,

$$p_{\text{detect}}(\mathbf{x}^{**}) = \mathbb{E}_{\mathbf{c}}[f(\epsilon_{-i})]\,, \qquad \text{where } \epsilon_{-i} \coloneqq \textstyle\sum_{k \neq i} c_k e'_k. \tag{5}$$

[5] Since each of the τ nonzero entries of $\mathbf{c}$ contributes at most 2η through $\mathbf{e}'$, we have $|\epsilon| \leq 2\tau\eta = 2\beta$, so the overlap is also always non-negative and, in particular, $p_{\text{detect}} \leq 2\beta/(2\beta + 1) < 1$.

To compare these two expectations, we note that $\epsilon_{-i} + c_i e'_i = \epsilon = \langle \mathbf{c}, \mathbf{e}' \rangle$. Now consider the number $t \coloneqq |\{k \neq i : c_k \neq 0\}|$ of nonzero entries already placed in $\mathbf{c}$ beyond c_i. Since $\mathbf{c}$ has exactly τ nonzero entries in total, there are two cases for the remaining randomness in c_i:

- **Case $t = \tau$:** Then $c_i = 0$ and therefore $f(\epsilon_{-i} + c_i e'_i) = f(\epsilon_{-i})$.
- **Case $t = \tau - 1$:** Then $c_i \in \{-1, +1\}$ each with probability $\frac{1}{2}$. By Jensen's inequality applied to the convex function f,

$$\mathbb{E}_{c_i}[f(\epsilon_{-i} + c_i e'_i)] = \frac{1}{2} f(\epsilon_{-i} + e'_i) + \frac{1}{2} f(\epsilon_{-i} - e'_i) \;\geq\; f(\epsilon_{-i}).$$

In both cases the contribution is at least $f(\epsilon_{-i})$. Averaging over all valid assignments of the remaining entries of $\mathbf{c}$ (i.e. taking the outer expectation) yields $p_{\text{detect}}(\mathbf{x}^*) \geq p_{\text{detect}}(\mathbf{x}^{**})$. □

We note that in all practically relevant settings (compare Table 1), the inequality of Lemma 1 is in fact strict:

Corollary 1 (Strict monotonicity). *In Lemma 1, we have the strict inequality $p_{\text{detect}}(\mathbf{x}^*) > p_{\text{detect}}(\mathbf{x}^{**})$ as long as $\tau \leq n - 2\eta - 1$ and either τ is odd, or one position g is already guessed correctly (i.e. $x^*_g = x_g$).*

Proof. Recall $t \coloneqq |\{k \neq i : c_k \neq 0\}| \in \{\tau - 1, \tau\}$ from the proof of Lemma 1: The contribution to $p_{\text{detect}}(\mathbf{x}^*) - p_{\text{detect}}(\mathbf{x}^{**})$ vanishes on $\{t = \tau\}$ and is non-negative on $\{t = \tau - 1\}$ by Jensen's inequality applied to the convex function $f(\epsilon) = |\epsilon|/(2\beta + 1)$. Since f is strictly convex at the origin and $e'_i \neq 0$, the inequality is strict on the sub-event $\{t = \tau - 1,\ \epsilon_{-i} = 0\}$. It therefore suffices to construct a single $\mathbf{c}$ with $c_i \neq 0$ and $\epsilon_{-i} = 0$: The corresponding elementary event has positive probability and contributes strictly, while every other configuration contributes non-negatively.

To this end, we call (j, h) with $j, h \neq i$ a *cancellation pair* if $|e'_j| = |e'_h|$. Such a pair yields $c_j e'_j + c_h e'_h = 0$ with $c_j = +1$, and $c_h = +1$ if $e'_h = 0$ and $c_h = -\frac{e'_j}{e'_h}$ otherwise. By partitioning $[n] \setminus \{i\}$ into the $2\eta + 1$ buckets indexed by $|e'_k| \in \{0, 1, \ldots, 2\eta\}$, we see that there are at least $(n-1) - (2\eta+1) = n - 2\eta - 2$ items that are part of cancellation pairs, since each bucket can contain at most one unmatched item. As we assumed $\tau \leq n - 2\eta - 1$, we can select and adjust the remaining $\tau - 1$ nonzero entries of $\mathbf{c}$ directly corresponding to those positions within cancellation pairs if τ is odd. For even τ, we first set $c_g = +1$ and then use the previous argument with the other $\tau - 2$ entries of $\mathbf{c}$. □

Proposition 1 (Extension to the low-leakage regime). *When $\beta_{\text{eff}} = 2^{j-1} < \beta$, any lower bound on the per-relation detection probability derived at Hamming distance $D = 1$ with error bound β is automatically a valid and conservative lower bound for the β_{eff}-verification as well.*

Proof. When $\beta_{\text{eff}} < \beta$, the per-relation detection function becomes $f_{\text{eff}}(\epsilon) = \min\left(1,\ \frac{|\epsilon|}{2\beta_{\text{eff}}+1}\right)$, which is *not convex* since $|\epsilon| > 2\beta_{\text{eff}} + 1$ is possible and therefore

not amenable to the Jensen argument of Lemma 1. Nonetheless, the conclusion that the worst-case detection is attained at Hamming distance 1 carries over to this regime.

Since $\beta_{\text{eff}} \leq \beta$, the β_{eff}-verification imposes a strictly tighter acceptance window, giving $p^{(\beta_{\text{eff}})}_{\text{detect}}(\mathbf{x}^*) \geq p^{(\beta)}_{\text{detect}}(\mathbf{x}^*)$ for every candidate $\mathbf{x}^*$. Combining this with the monotonicity established in Lemma 1 yields, for any $\mathbf{x}^* \neq \mathbf{x}$,

$$p^{(\beta_{\text{eff}})}_{\text{detect}}(\mathbf{x}^*) \;\geq\; p^{(\beta)}_{\text{detect}}(\mathbf{x}^*) \;\geq\; \min_{\mathbf{x}'} p^{(\beta)}_{\text{detect}}(\mathbf{x}'),$$

where the right-hand side is the minimum over all wrong candidates $\mathbf{x}'$ at Hamming distance $D = 1$ from the true key $\mathbf{x}$ (by Lemma 1). Hence the claimed bound follows. □

As a consequence of Lemma 1 (extended to arbitrary β_{eff} by Proposition 1), we thus know that the worst-case detection probability over all $\mathbf{x}^* \neq \mathbf{x}$ is lower-bounded by the detection probability at Hamming distance $D = 1$ from the true key $\mathbf{x}$. We use that insight to prove the following:

Lemma 2. *Given a random (valid) informative relation and input $\mathbf{x}^* \neq \mathbf{x}$, Algorithm 1 returns False with probability at least $\left(2n\eta + \frac{n}{\tau}\right)^{-1}$.*

Proof. Let D be the number of positions in which x and $\mathbf{x}^*$ differ, which is positive as $\mathbf{x}^* \neq \mathbf{x}$. Because of Lemma 1 and Proposition 1 we assume $D = 1$ without loss of generality.

If $\mathbf{x}$ and $\mathbf{x}^*$ diverge in exactly one position i, then $\langle \mathbf{c}, \mathbf{x} \rangle \neq \langle \mathbf{c}, \mathbf{x}^* \rangle$ if and only if $\mathbf{c}_i \neq 0$, which clearly happens with probability $\frac{\tau}{n}$. In those cases, let $\epsilon = \langle \mathbf{c}, \mathbf{x}^* \rangle - \langle \mathbf{c}, \mathbf{x} \rangle$. For the check in Line 2 of Algorithm 1 to succeed, we additionally need the independent random variable $\widetilde{y}$ to be small or large enough such that either $\epsilon > \widetilde{y} + \beta$ or $\epsilon < \widetilde{y} - \beta$ (depending on the sign of ϵ). The marginal probability for such a $\widetilde{y}$ is at least $\frac{1}{2\beta+1}$. Therefore, the overall probability of the algorithm correctly identifying any false candidate $\mathbf{x}^* \neq \mathbf{x}$ is at least $\frac{\tau}{n} \cdot \frac{1}{2\beta+1} = \frac{1}{2n\eta + \frac{n}{\tau}} = \left(2n\eta + \frac{n}{\tau}\right)^{-1}$. □

Remark 2 (Tightness in the low-leakage regime). The bound of Lemma 2 uses β (i.e. the worst case $\beta_{\text{eff}} = \beta$). In the low-leakage regime where $\beta_{\text{eff}} = 2^{j-1} < \beta$, the per-relation detection probability is at least $\frac{\tau}{n} \cdot \frac{1}{2\beta_{\text{eff}}+1} > \frac{\tau}{n} \cdot \frac{1}{2\beta+1}$, so the bound of Theorem 2 is conservative. In particular, fewer informative relations suffice for a given confidence level when $\beta_{\text{eff}} < \beta$.

Using Algorithm 1 and Lemma 2, we can now easily show Theorem 2.

Proof of Theorem 2. Since the random outcome of Algorithm 1 only depends on the randomness of the input informative relation, we can just repeat it on α many informative relations which are stochastically independent from each other (using the same β_{eff} and $\mathbf{x}^*$ in each call). If $\mathbf{x}^* = \mathbf{x}$, then each iteration will output *True*, as described. If $\mathbf{x}^* \neq \mathbf{x}$, the probability that every iteration still outputs *True* is at most $\left(1 - \left(2n\eta + \frac{n}{\tau}\right)^{-1}\right)^{\alpha}$ by application of the complementary probability of

Lemma 2. Thus, we can distinguish the two cases with a probability of at least $1 - \left(1 - \left(2n\eta + \frac{n}{\tau}\right)^{-1}\right)^{\alpha}$, finishing the proof. □

Wrapping up, we complementarily want to mention that if Algorithm 1 is run repeatedly to verify some candidate $\mathbf{x}^*$ with high probability, then the single iterations may of course also be processed in parallel, linked with logical *AND*-gates. As a verification routine, it may also make sense to pre-compute $\widetilde{z} \pm \beta$ for each input relation, such that only the scalar product $\langle \mathbf{c}, \mathbf{x}^* \rangle$ has to be computed and its two bound checks performed at each iteration.

4 An Improved ML-DSA Subkey Finder

The verification routine of Section 3 provides a binary test: Given a candidate $\mathbf{x}^*$, it detects $\mathbf{x}^* \neq \mathbf{x}$ with positive probability by checking whether $\langle \mathbf{c}, \mathbf{x}^* \rangle$ falls outside the interval $[\widetilde{z} - \beta_{\text{eff}}, \widetilde{z} + \beta_{\text{eff}}]$, where $\beta_{\text{eff}} = \min(\beta, 2^{j-1})$ is the regime-dependent error bound (cf. Equation (3)). A natural extension is to use this test not merely to *accept* or *reject* a candidate, but to *compare* candidates: A candidate that triggers more violations across a collection of informative relations is, on expectation, farther from the true key. This idea is justified by Lemma 1 and Corollary 1, which together establish that correcting any single wrong position in a candidate key *strictly reduces* its expected detection rate (in practically relevant cases). In this section, we formalize this comparison using a scoring function and build a respective hill-climbing algorithm[6] that iteratively improves a candidate position by position.

4.1 ML-DSA Subkey Evaluation

Count-based scoring. The most direct way to turn the verification routine into a scoring function is to run Algorithm 1 on α independent informative relations and count how many return *False*. The resulting count is a random variable whose expectation, by Lemma 1 (extended to arbitrary β_{eff} via Proposition 1), is monotonically related to the error structure of $\mathbf{x}^*$: Correcting any one wrong position can only decrease the expected count, and by Corollary 1 does so strictly. Thus, for sufficiently large α, the empirical count serves as a proxy for distance from the true key, as reflected in Algorithm 2.

Excess-based scoring. The count-based score treats every constraint violation equally, regardless of *how far* $\langle \mathbf{c}, \mathbf{x}^* \rangle$ falls outside the acceptance window $[\widetilde{z} - \beta_{\text{eff}}, \widetilde{z} + \beta_{\text{eff}}]$. A more informative variant accumulates the *excess* by which the constraint is violated:

$$r_t(\mathbf{x}^*) \;\coloneqq\; \max\big(0,\; |\delta_t| - \beta_{\text{eff}}\big),$$

[6] As the name of our paper already suggests, *hill descending* would be a more apt description—we are, after all, *minimizing* the cost function. Nevertheless, we adhere to the conventional algorithmic terminology.

Algorithm 2: ML-DSA Subkey Evaluation (count-based)

Input: α preprocessed informative relations $(\mathbf{c}_t, \widetilde{z}_t)_{t=1,\dots,\alpha}$, effective bound β_{eff}, leakage index j, verification candidate $\mathbf{x}^*$
Output: *score*, the number of constraint violations (0 if $\mathbf{x}^* = \mathbf{x}$)

```
score ← 0
for t = 1, ..., α do
    Compute residual δ_t as in Algorithm 1
    if |δ_t| > β_eff then
        score ← score + 1
return score
```

where δ_t denotes the residual of Algorithm 1. In addition to the merits of the previous count metric, the excess scoring also carries *gradient-like* information: A candidate that violates a constraint by a large margin receives a higher penalty than one that barely violates it, which in practice leads to better discrimination between candidates of similar quality.

Moreover, for a single relation, the expected excess $g(\epsilon) \coloneqq \mathbb{E}_{\widetilde{y}}[r_t(\mathbf{x}^*)]$ is a convex function of $\epsilon = \langle \mathbf{c}, \mathbf{x}^* - \mathbf{x} \rangle$, since it is the expectation of the composition of the convex non-decreasing function $u \mapsto \max(0, u - \beta_{\text{eff}})$ with the convex function $\epsilon \mapsto |\epsilon - \widetilde{y}|$. Consequently, the same type of Jensen-based argument as in the proof of Lemma 1 applies in all leakage regimes: Correcting a wrong position can only ever decrease the expected cumulative excess. This convexity, in fact, gives the excess score a structural benefit over the count-based score within the low leakage-index regime (i.e. when $\beta_{\text{eff}} < \beta$). With the count-based indicator, the per-relation detection function $f_{\text{eff}}(\epsilon) = \min\left(1, \frac{|\epsilon|}{2\beta_{\text{eff}}+1}\right)$ saturates at 1 for large $|\epsilon|$ and is therefore *not* convex on its entire domain when $\beta_{\text{eff}} < \beta$ (cf. Proposition 1). The hinge function underlying the excess, by contrast, grows without bound: For each fixed $\widetilde{y}$, the map $\epsilon \mapsto \max(0, |\epsilon - \widetilde{y}| - \beta_{\text{eff}})$ is convex on all of $\mathbb{R}$, and so is its expectation $g(\epsilon)$. The Jensen-based monotonicity argument therefore applies to the excess score *directly* in both leakage regimes, without the indirect comparison via Proposition 1 that the count-based score requires.

In our experimental evaluation (see Section 6), we found that the excess-based score of Algorithm 3 consistently yields better discrimination between candidates and we therefore use it as the default scoring function for the hill-climbing algorithm introduced next. However, we note that in the presence of noise—where the informative relations are not exact—the count-based score of Algorithm 2 is needed; we revisit this observation in Section 5.

4.2 Hill-Climbing Algorithm

With a scoring function that assigns lower values to better candidates and reaches zero exactly at the true key, we can formulate the key recovery as a local optimization problem: Starting from an initial guess, we iteratively

Algorithm 3: ML-DSA Subkey Evaluation (excess-based)

Input: α preprocessed informative relations $(\mathbf{c}_t, \widetilde{z}_t)_{t=1,\dots,\alpha}$, effective bound β_{eff}, leakage index j, verification candidate $\mathbf{x}^*$
Output: *score*, the cumulative constraint excess (0 if $\mathbf{x}^* = \mathbf{x}$)

score $\leftarrow 0$
for $t = 1, \dots, \alpha$ **do**
    Compute residual δ_t as in Algorithm 1
    if $|\delta_t| > \beta_{\text{eff}}$ **then**
        score $\leftarrow$ *score* $+ |\delta_t| - \beta_{\text{eff}}$
return *score*

modify the intermediate candidate to reduce its score. The theoretical grounding for this approach is provided by the following assumption, which we derive from the monotonicity results of Section 3. However this assumption will not hold in practice and we will introduce optimization strategies in Section 4.3 to counterattack the failing assumption.

Assumption 1. Given a fixed set of α informative relations and two subkey candidates $\mathbf{x}', \mathbf{x}''$ for the correct subkey $\mathbf{x}$ such that $x'_i = x_i \neq x''_i$ for one position i and $x'_j = x''_j$ for all $j \neq i$, the candidate with the correct value at position i receives a strictly lower score:

$$\textsc{SubkeyEvaluation}(\mathbf{x}') \;<\; \textsc{SubkeyEvaluation}(\mathbf{x}'').$$

Under Assumption 1, the following greedy strategy recovers the true key: For each position i in turn, evaluate all $(2\eta + 1)$ possible values and keep the one that minimizes the overall score. Since the true value at each position is by assumption the unique minimizer, a single pass through all n positions suffices.

Algorithm 4: ML-DSA Subkey Finder (under Assumption 1)

Input: α preprocessed informative relations $(\mathbf{c}_t, \widetilde{z}_t)_{t=1,\dots,\alpha}$, effective bound β_{eff}, leakage index j, distribution width η, initial candidate $\mathbf{x}^*$
Output: ML-DSA subkey $\mathbf{x}$

score $\leftarrow \textsc{SubkeyEvaluation}\big((\mathbf{c}_t, \widetilde{z}_t)_{t\in[\alpha]},\ \beta_{\text{eff}},\ j,\ \mathbf{x}^*\big)$
for $i \in \{1, \dots, n\}$ **do**
    for $v \in \{-\eta, \dots, \eta\} \setminus \{x^*_i\}$ **do**
        $\mathbf{x}' \leftarrow \mathbf{x}^*$ with $x'_i \leftarrow v$
        score$'$ $\leftarrow \textsc{SubkeyEvaluation}\big((\mathbf{c}_t, \widetilde{z}_t)_{t\in[\alpha]},\ \beta_{\text{eff}},\ j,\ \mathbf{x}'\big)$
        if *score*$'$ $<$ *score* **then**
            score $\leftarrow$ *score*$'$
            $\mathbf{x}^* \leftarrow \mathbf{x}'$
return $\mathbf{x}^*$

A natural choice for the initial candidate is the rounded output of a linear regression similar to that of Damm *et al.* [5], which already places many positions at their correct values and thus reduces the number of corrections needed.

Limitations of the base algorithm. In practice, Assumption 1 does not hold for the amounts of informative relations that are of interest to us: With moderate α, the stochastic evaluation score does not always preserve the expected ordering, and the single-pass algorithm may fix a position to an incorrect value. In the following subsection, we describe several optimization strategies that make the algorithm robust to violations of Assumption 1, enabling reliable key recovery even in settings with comparatively little leakage information.

4.3 Optimization Strategies

The base algorithm of Section 4.2 often gets trapped in local minima when Assumption 1 is violated. We address this with a layered optimization scheme: The algorithm first exhausts cheap single-position sweeps, then gradually increases the search radius through adaptive block sizes and lateral moves, and finally resorts to perturbations when local methods fail. We now describe each component in turn.

Multiple sweep rounds. The most immediate extension is to repeat the sweep multiple times rather than stopping after a single pass. In each round, we visit all n positions and, for each position, try all $(2\eta+1)$ values, keeping any improvement found. Multiple rounds are necessary because correcting position i can change the score landscape at position j: A value that appeared optimal for j in an earlier round may no longer be so once i has been corrected. The algorithm terminates as soon as the score reaches zero, indicating that the candidate satisfies all α constraints.

Adaptive block size. When single-position sweeps stall—no position can be improved in isolation—the algorithm may be trapped in a local minimum caused by *compensating errors*: Two or more wrong positions whose individual contributions to the score cancel each other. To escape such configurations, we increase the *block size* w, i.e. the number of positions varied simultaneously: rather than trying all values for a single position, we select w positions and enumerate all $(2\eta+1)^w$ joint assignments. Since the cost grows exponentially in w, the algorithm starts with $w = 1$ (standard sweeps) and increases w only after a configurable number of rounds without improvement. This ensures that the expensive multi-position search is only invoked when cheaper strategies have been exhausted.

Lateral moves. Even with multiple sweeps, the algorithm may reach a *plateau*: A region of the search space where no single-position change (or small-block change) yields a strict improvement, yet the current candidate is not the true key. For these situations, we accept *lateral moves* by default, accepting a new candidate $\mathbf{x}'$ with the same score as the current candidate $\mathbf{x}^*$, provided $\mathbf{x}' \neq \mathbf{x}^*$. By traversing the plateau rather than stalling on it, the algorithm can reach a point from which further descent becomes possible.

Perturbation-based restarts. If the algorithm remains stuck despite increased block sizes and lateral moves, a more drastic intervention is needed. A *perturbation* randomly resets a configurable number of positions in the current candidate to fresh values drawn from $\{-\eta, \ldots, \eta\}$, effectively teleporting the search to a different region of the solution space. After a perturbation, the block size is reset to $w = 1$ and normal sweeps resume, in the hope that the new starting point lies in a basin of attraction that leads to the true key. The number of positions perturbed each time (the *perturbation strength*) controls a trade-off between exploration and risk of undoing previous progress.

5 Presence of Noise

In this section, we shift our attention to the practical settings where the leaked bit y_j is corrupted by some noise in the side channel. Damm *et al.* [6] address this scenario in their latest revised ePrint version for the high-leakage regime ($2^{j-1} > \beta$) with experiments on a single parameter set (ML-DSA-44 at $j = 8$). We extend their treatment to both leakage regimes, and analyze the implications for our scoring functions and our previously described algorithm.

Noisy leakage model. We generalize Definition 1 by replacing the exact bit y_j with a noisy observation.

Definition 3 (Noisy Leaky-Signature-LWE). *In the setting of Definition 1, let $p \in [0, \frac{1}{2})$ be an error rate. Instead of the true bit y_j, the attacker receives $y_j^{(p)} = y_j \oplus \zeta$ with $\zeta \sim \mathrm{Ber}(p)$. Given $(\mathbf{A}, \mathbf{t})$ and arbitrarily many tuples $(\mathbf{c}, z, y_j^{(p)})$, the attacker's goal is again to find $\mathbf{x}$.*

As a consequence, a fraction of approximately $(1-p)$ of the collected relations carry the correct bit $y_j^{(p)} = y_j$, and the remaining fraction p carry the flipped bit $y_j^{(p)} = y_j \oplus 1$. Since the attacker cannot distinguish the two types at first glance, all of those relations are processed by the extraction and transformation procedures of Section 2.

5.1 Effect of a Bit Flip on the Extracted Relation

In order to characterize how the extracted value $\overline{z}$ changes with flips of the leaked bit, we import the following result of Damm *et al.* [6]. We observe that, although their proof was written for the high-leakage regime, it directly carries over to the low-leakage setting, as it does not rely on the constraint $j \geq \log_2(\beta) + 1$:

Theorem 4 (Theorem 9 in [6]). *Let $\overline{z}^{(c)}$ denote the result of the relation extraction on input (z, y_j), and let $\overline{z}^{(f)}$ denote the result on input $(z, y_j \oplus 1)$. Then*

$$\overline{z}^{(f)} = \overline{z}^{(c)} - \mathrm{sign}\left(\overline{z}^{(c)}\right) \cdot 2^j. \tag{6}$$

The identity holds over $\mathbb{Z}$, regardless of the leakage regime.

A flipped leaked bit therefore perturbs $\overline{z}$ by exactly 2^j, a large displacement relative to the typical magnitude of the underlying error. The next subsection traces this displacement through the post-extraction filtering and shows that it has structurally different consequences in the two leakage regimes.

5.2 Consequences by Leakage Regime

The bit-flip shift from Theorem 4 has different consequences among the two leakage regimes, which we now analyze in turn.

High-Leakage Regime ($2^{j-1} > \beta$) In this regime the j-independence transformation of Theorem 3 is applied, $\beta_{\text{eff}} = \beta$, and not all relations are informative. The noise-flipped relations split into two categories depending on whether the *underlying correct extraction* $\overline{z}^{(c)}$ was informative.

Identifiable noise. If $\overline{z}^{(c)}$ was *not* informative, i.e. $|\overline{z}^{(c)}| < 2^{j-1} - \beta$, then Theorem 4 gives $|\overline{z}^{(f)}| = |\overline{z}^{(c)}| + 2^j > 2^{j-1} + \beta$. Since under correct extraction we always have $|\overline{z}| \leq 2^{j-1} + \beta$, such values are impossible in the noiseless setting and can be identified and discarded. Formally, we refine the notion of informative relations in the presence of noise:

Definition 4 (Informative relation under noise, high-leakage regime). *In the high-leakage regime with noise rate p, an extracted relation is called* informative *if $2^{j-1} + \beta \geq |\overline{z}| \geq 2^{j-1} - \beta$. Relations with $|\overline{z}| > 2^{j-1} + \beta$ are identifiably noise-corrupted and discarded. Relations with $|\overline{z}| < 2^{j-1} - \beta$ are non-informative (as in the noiseless case) and also discarded*[7]*.*

This filtering is specific to the high-leakage regime. In particular, it is a strict extension of Definition 2: In the noiseless case ($p = 0$) the upper bound is automatically satisfied by all relations and Definition 4 reduces to Definition 2.

Unidentifiable noise. If $\overline{z}^{(c)}$ *was* informative, i.e. $2^{j-1} + \beta \geq |\overline{z}^{(c)}| \geq 2^{j-1} - \beta$, then the flipped extraction $\overline{z}^{(f)}$ also satisfies the informativeness criterion of Definition 4, but falls in the opposite half of the informative window. The noise is then invisible to the attacker. In this case the j-independence transformation is applied to the (incorrectly extracted) value, and we can characterize its effect on $\tilde{z}$:

Corollary 2. *In the high-leakage regime, let $\tilde{z}^{(c)}$ and $\tilde{z}^{(f)}$ denote the transformed values (via (2)) for the correct and flipped bit respectively, where the underlying $\overline{z}^{(c)}$ is informative. Then*

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

[7] This mechanism can also be used to estimate the noise rate p if it is unknown: The fraction of identifiably noise-corrupted relations $|\overline{z}| > 2^{j-1} + \beta$ among all discarded relations approximates the noise rate p.

This follows by substituting the shift of Theorem 4 into the two branches of the j-independence transformation (2): The flipped $\overline{z}^{(f)}$ lands in the opposite branch, and the 2^j shift reduces to 2β after the branch offsets cancel.

For the correct key $\mathbf{x}^* = \mathbf{x}$, the residual of an unidentifiably noisy relation is therefore $\delta^{(f)} = \text{sign}(\overline{z}^{(c)}) \cdot 2\beta - \tilde{y}$, where $\tilde{y} \in [\pm\beta]$ is the original error. In particular, $|\delta^{(f)}| \in [\beta, 3\beta]$, so every such relation triggers a constraint violation even for the correct key.

Low-Leakage Regime ($2^{j-1} \leq \beta$) In this regime the j-independence transformation is not applied, $\beta_{\text{eff}} = 2^{j-1}$, and all relations are informative (cf. Section 2). We work with the extracted value $\overline{z}$ directly, and all arithmetic is modulo $M = 2^{j+1}$.

No identifiable noise. Since all relations are informative and $\overline{z}$ is always taken modulo M with values in $[-2^j, 2^j - 1]$, a noise-flipped relation produces a value $[\overline{z}^{(f)}]_M$ that is again a valid element of this range. There exists no value of $\overline{z}$ that would be impossible under correct extraction, so the attacker cannot distinguish noisy from clean relations. In particular, the filtering mechanism of Definition 4 is not available in this regime.

Residual analysis. For the correct key $\mathbf{x}^* = \mathbf{x}$ with a noise-flipped relation, the residual before modular reduction is

$$\delta^{(f)} = \langle \mathbf{c}, \mathbf{x} \rangle - \overline{z}^{(f)} = -[y]_{2^j} + \text{sign}\big(\overline{z}^{(c)}\big) \cdot 2^j,$$

where $[y]_{2^j} \in [-2^{j-1}, 2^{j-1} - 1]$. After centered reduction modulo $M = 2^{j+1}$ (Line 2 of Algorithm 1), the residual satisfies

$$\big|[\delta^{(f)}]_M\big| = 2^j - |[y]_{2^j}|. \tag{7}$$

We verify this for $\text{sign}(\overline{z}^{(c)}) = +1$ (the -1 case is symmetric). Then $\delta^{(f)} = 2^j - [y]_{2^j}$, and we distinguish:

Case $[y]_{2^j} > 0$: Here $\delta^{(f)} = 2^j - [y]_{2^j} \in [2^{j-1} + 1, 2^j - 1]$ already lies in the centered range $[-2^j, 2^j - 1]$, so $[\delta^{(f)}]_M = \delta^{(f)}$ and $\big|[\delta^{(f)}]_M\big| = 2^j - [y]_{2^j} = 2^j - |[y]_{2^j}|$.

Case $[y]_{2^j} \leq 0$: Here $\delta^{(f)} = 2^j - [y]_{2^j} \in [2^j, 3 \cdot 2^{j-1}]$ is at least as large as $2^j = M/2$, so centered reduction yields $[\delta^{(f)}]_M = \delta^{(f)} - M = -2^j - [y]_{2^j} \in [-2^j, -2^{j-1}]$, hence $\big|[\delta^{(f)}]_M\big| = 2^j + [y]_{2^j} = 2^j - |[y]_{2^j}|$.

In either case, $\big|[\delta^{(f)}]_M\big| \in [2^{j-1}, 2^j]$, which exceeds $\beta_{\text{eff}} = 2^{j-1}$ unless $|[y]_{2^j}| = 2^{j-1}$ — attained only at $[y]_{2^j} = -2^{j-1}$, i.e., exactly 1 out of 2^j possible values (using our boundary conventions). Thus a noise-flipped relation triggers a constraint violation at the correct key with probability

$$\Pr[\text{violation} \mid \text{noisy, correct key}] = \frac{2^j - 1}{2^j}, \tag{8}$$

which approaches 1 rapidly (e.g. $\approx 98.4\%$ for $j = 6$). The behavior is thus nearly identical to the high-leakage regime: Noise-flipped relations almost always lead to constraint violations at the correct key.

5.3 Scoring Under Noise

The preceding analysis shows that in both leakage regimes, a fraction p of relations produce large constraint violations for *any* candidate—including the correct key $\mathbf{x}$. This has direct consequences for the choice of scoring function.

Unreliability of excess-based scoring under noise. The excess-based score (Algorithm 3) accumulates the violation magnitude $\max(0, |\delta| - \beta_{\text{eff}})$. Under noise, each flipped relation contributes an excess of up to 2β (high-leakage) or $\beta_{\text{eff}} = 2^{j-1}$ (low-leakage) for the correct key. In contrast, a single wrong position at Hamming distance 1 contributes at most 2η of additional excess per relation (when $c_i \neq 0$). Since $\beta = \eta\tau \gg \eta$ and $2^{j-1} \gg 2\eta$ for practical parameters, a single noisy relation can induce more excess than many relations' combined signal from a genuinely wrong position. The excess-based score therefore does not reliably preserve the monotonicity of Lemma 1 under noise, and using it in this setting would misdirect the hill-climbing algorithm.

Count-based scoring. The count-based score (Algorithm 2) counts the number of violated constraints without regard to the violation magnitude. Each relation, whether clean or noisy, contributes at most $+1$ to the score. The noisy fraction contributes a violation count of approximately $p \cdot \alpha$ for any candidate (correct or wrong), since the large shift from the bit flip dominates the small perturbation $\epsilon = \langle \mathbf{c}, \mathbf{x}^* - \mathbf{x} \rangle$. The clean fraction, in contrast, discriminates between candidates exactly as in the noiseless case: An incorrect candidate receives an additional expected count of $(1-p) \cdot \alpha \cdot f_{\text{eff}}(\epsilon)$ violations beyond what the correct key would incur (where f_{eff} is the per-relation detection function of Section 3). Since $(1-p) > p$, this gap grows linearly with α and eventually dominates the statistical fluctuations, so that the count-based score reliably identifies the correct key for sufficiently large α. For this reason, the count-based score is our chosen scoring function in the presence of noise.

*Remark 3 (**ILP formulation**).* We note that the key recovery problem admits natural integer linear programming formulations both in the noisy and in the exact setting. After we first released this work as a preprint and briefly described these options, Bashiri, Geuenich, and Mittmann [29] extensively pursued this direction in a new preprint, additionally including NLP formulations and experimental comparisons drawing on a range of open-source and commercial solvers, and in turn improved on our experimental results. We refer to their work for a comprehensive analysis of those methods.

5.4 Algorithmic Adaptations

The presence of noise requires further modifications to the hill-climbing algorithm of Section 4. Most notably, under noise, the correct key $\mathbf{x}$ does not achieve a score of zero; instead, it incurs a positive expected count of approximately $p \cdot \alpha \cdot \Pr[\text{violation} \mid \text{noisy}]$, which is strictly positive for $p > 0$. The attacker does not know this noise floor *a priori*, so they need a different termination condition.

Patience-based termination. We replace the zero-score termination with a *patience-based* criterion: The algorithm records the best score encountered and terminates when this score has not improved for a configurable number of iterations ("patience"). This design reflects an inherent time–accuracy trade-off. A low patience value yields fast termination, but risks stopping prematurely. Higher patience values give the optimization strategies more time to escape local minima, but may waste computation if the algorithm has already found the correct key and the remaining score is entirely due to noise. An attacker cannot confidently distinguish between these situations without additional information.

Warm start. As in the non-noisy case, the OLS regression of Damm *et al.* [6] can be used to initialize the hill-climbing candidate. However, as they showed in their Theorem 12, a rescaling of $\hat{x} \mapsto \hat{x}/(1-2p)$ is needed (following their Algorithm 3) to accommodate a shifted bias of the least-squares expectation value.

6 Results

This section presents the experimental evaluation of our hill-climbing attack. Section 6.1 covers the non-noisy setting, in which the leakage measurements are assumed to be exact; Section 6.2 extends the evaluation to the noisy leakage model. Our Python reference implementation, experiment scripts, and all raw data including additional information (such as wall-clock times) are publicly available at `https://github.com/D-VAmpire/mldsa-hill-climbing`.

Test setup. A fixed random seed (42) is used both for (simulated) key generation and all stochastic components of the hill-climbing procedures. In regards to the optimization parameters: The adaptive block size starts at $w = 1$ and increases to $w = 2$ after a single full position sweep, thereafter increasing by one after every 100 iterations, up to a maximum of $w = 4$. Improvements reset w to 1. Perturbations also reset w to 1, change 30 randomly selected positions, and are triggered after 150 consecutive iterations at maximum block size.

6.1 Exact Setting: Minimum Leakage Requirements

We evaluate our hill-climbing attack across all three ML-DSA parameter sets (ML-DSA-44, ML-DSA-65, and ML-DSA-87) and up to six leakage indices ($j \in \{4, 5, 6, 7, 8, 9\}$) in the non-noisy setting, sweeping the number of informative relations downward—in decrements of 500 informative relations each—to identify the minimum count at which all tested keys can still be recovered. We chose those leakage indices for the reasons outlined in Section 2.2.

Success criterion. We say that a *key recovery is successful* if, for a given subkey, the hill-climbing procedure recovers the true secret subkey within a budget of $T = 100{,}000$ *iterations*, where one iteration counts either as a full parallel sweep over all n positions at block size $w = 1$ or as a single block evaluation at adaptive block size $w \geq 2$. More precisely, one of the following two conditions must hold:

– **Unique recovery:** The constraint system admits exactly one feasible solution which is found at termination and coincides with the true secret subkey; or
– **Ambiguous recovery:** The constraint system admits more than one feasible solution, yet the number of found candidates is at most 20, and the true secret subkey is among them.[8]

Reporting in a conservative fashion, we define a *parameter configuration* (i.e. a pair of ML-DSA variant and leakage-index) to be successful if *all of ten* independently generated keys satisfy this criterion. As described in Section 4.1, we used excess-based scoring for all subkey evaluations. Additionally, all optimizations described in Section 4.3 are enabled throughout.

		Parameter set		
Leakage j	**Metric**	ML-DSA-44	ML-DSA-65	ML-DSA-87
4	m^* (inf. rels)	1,500	–	2,500
	$\overline{N}_{\text{sig}}$	1,500	–	2,500
5	m^* (inf. rels)	3,000	3,000	2,500
	$\overline{N}_{\text{sig}}$	3,000	3,000	2,500
6	m^* (inf. rels)	6,000	5,500	5,000
	$\overline{N}_{\text{sig}}$	6,000	5,500	5,000
7	m^* (inf. rels)	11,500	11,500	9,500
	$\overline{N}_{\text{sig}}$	11,500	11,500	9,500
8	m^* (inf. rels)	13,500	22,500	17,500
	$\overline{N}_{\text{sig}}$	22,329	22,500	18,740
9	m^* (inf. rels)	14,500	35,000	17,500
	$\overline{N}_{\text{sig}}$	48,105	45,818	37,517

Table 2: Leakage information requirements in the exact setting. For each parameter set and leakage index j, we report the minimum number of informative relations m^* at which all 10 keys were recovered, and the average number of generated signatures $\overline{N}_{\text{sig}}$. Since we did not achieve a recovery at leakage index 4 for ML-DSA-65 no numbers are reported.

Results. Table 2 reports, for each parameter configuration, the minimum number of informative relations m^* at which a 10/10 success rate is achieved, as well as the corresponding average number of signatures $\overline{N}_{\text{sig}}$ required. m^* equals $\overline{N}_{\text{sig}}$ at $j \in \{4, 5, 6, 7\}$ across all variants and additionally at $j = 8$ for ML-DSA-65, since every signature in those configurations is informative (cf. Definition 2).

[8] This notion mimics the scenario where, despite an underdetermined constraint system, the attacker still computes $\mathbf{x}$ with reasonable effort and directly verifies it.

ML-DSA-44	$j = 6$	$j = 7$	$j \geq 8$
Damm *et al.* [5]	$90\,000^{\dagger}$	$280\,000^{\dagger}$	$500\,000$
Our Attack	$6\,000$	$11\,500$	$13\,500$
Improvement Factor $\geq$	$15.0^{\dagger}$	$24.3^{\dagger}$	37.0
ML-DSA-65	$j = 7$	$j = 8$	$j \geq 9$
Damm *et al.* [5]	$375\,000^{\dagger}$	$1\,100\,000^{\dagger}$	$2\,400\,000$
Our Attack	$11\,500$	$22\,500$	$35\,000$
Improvement Factor $\geq$	$32.6^{\dagger}$	$48.9^{\dagger}$	68.5
ML-DSA-87	$j = 6$	$j = 7$	$j \geq 8$
Damm *et al.* [5]	$75\,000^{\dagger}$	$185\,000^{\dagger}$	$750\,000$
Our Attack	$5\,000$	$9\,500$	$17\,500$
Improvement Factor $\geq$	$15.0^{\dagger}$	$19.5^{\dagger}$	42.8

Table 3: Results Comparison for High Leakage Indices with [5]. We provide the number of informative relations needed for the attack from Damm *et al.* [5] and for our attack as well as the resulting improvement factor. Numbers marked with a dagger (†) may not be accurate, since they are taken from figures 8-9 in [5] or depend on such numbers.

Comparison with Damm et al. Our main contribution compared to the previous work is a reduction in the number of informative relations necessary to recover the subkey $\mathbf{x}$. To make these improvements transparent, Table 3 provides a direct numerical comparison of the data requirements stated in Damm *et al.* [5] to those of our hill-climbing algorithm, together with the resulting reduction factors. Since both attacks extract informative relations with the same per-signature probability at any given leakage index, these reduction factors translate directly into reductions in the required number of leaky signatures. We further achieved key recovery at leakage bit indices as low as 4 (for ML-DSA-44 and ML-DSA-87) and 5 (for ML-DSA-65), which are regimes not addressed in their work.

6.2 Noisy Setting: Leakage Requirement Analysis

We now evaluate the hill-climbing attack under the noisy leakage model introduced by Definition 3, sweeping the corruption rate over $p \in \{0.10, 0.20, 0.30, 0.40, 0.45\}$; the upper end already exceeds the maximum of $p = 0.43$ tested by Damm *et al.* Figure 2 shows the resulting scaling of the required number of informative relations m^*.

Scope. We cover all three ML-DSA parameter sets, evaluating leakage indices $j \in \{6, 7, 8\}$ for ML-DSA-44 and ML-DSA-87, and additionally $j = 9$ for ML-DSA-65; this asymmetry reflects the variant-dependent boundary at which the j-independence transformation (Section 2.2) renders larger j informationally redundant. Coverage thus spans the high-leakage regime for every variant.

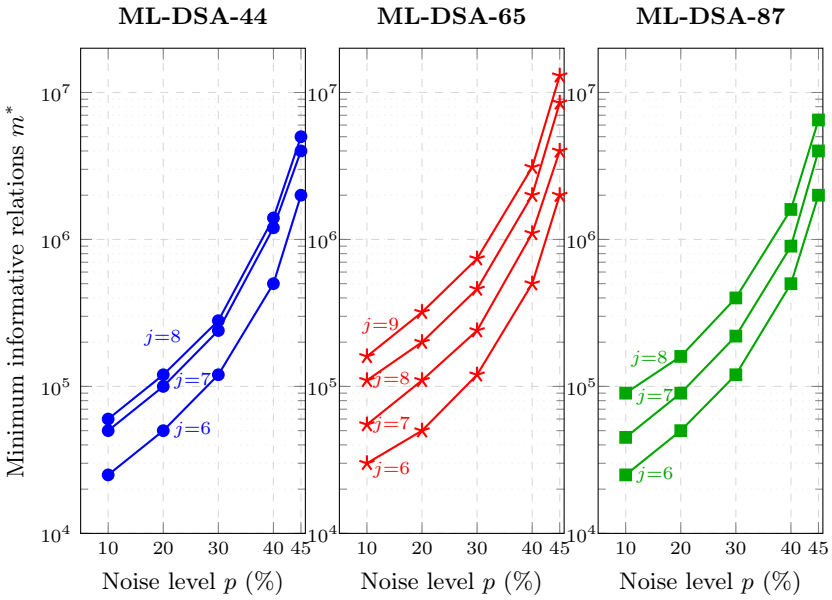

Fig. 2: Minimum number of informative relations m^* required for a successful hill-climbing key-recovery attack on ML-DSA, shown separately for each security level as a function of the leakage noise level level $p \in \{10, 20, 30, 40, 45\}\%$. Each data point shows the smallest m^* for which the hill-climbing algorithm attained at least a 4/5 success rate across five independent trials. Within each panel, curves correspond to different leakage indices $j \in \{6, 7, 8\}$ (ML-DSA-44 and ML-DSA-87) or $j \in \{6, 7, 8, 9\}$ (ML-DSA-65).

Experimental design choices. As described in Section 5.4, we use a patience budget as termination criterion in this setting: In our experiments every search terminated if no improvement of the best known intermediate result was found within an interval of 1,000 iterations. A real-world attacker investing more time per key would benefit from a higher patience budget and may thus succeed at fewer informative relations. The sweep interval size was approximately scaled to the magnitude of the number of informative relations at each noise level: We used decrements of 5,000; 10,000; 20,000; 100,000; and 500,000 at noise levels $p = 0.1$; $p = 0.2$; $p = 0.3$; $p = 0.4$; and $p = 0.45$; respectively. At $p \in \{0.1, 0.2, 0.3, 0.4\}$ we also reduced the maximum adaptive block size parameter w from 4 to 3. Further, compared to the previous exact setting (Section 6.1), we slightly relax the aggregate key recovery success threshold: A parameter configuration is considered successful if at least four out of five independently generated keys are recovered.

Comparison with Damm et al. Damm *et al.* [6] test ML-DSA-44 (at $j = 8$) with OLS regression for noise levels $p \in \{0.10, 0.20, 0.30, 0.40\}$ (Table 4 of their paper). Our sweep covers the four noise levels they report, plus the additional level

Noise level p	0.10	0.20	0.30	0.40
Attack of Damm *et al.* [6]	1.4M	3.1M	9.0M	47M
Our attack	0.06M	0.12M	0.28M	1.4M
Improvement factor $\geq$	23.3	25.8	32.1	33.5

Table 4: Matched-noise comparison with Damm *et al.* [6] on ML-DSA-44 at leakage index $j = 8$ (taken from their Table 4), each entry is the respective number of required informative relations in millions (M) to recover a subkey.

$p = 0.45$ for which they have no data. Table 4 shows the numerical comparison: The reduction factor is relatively stable, lying in the range 23–33×. Most notably, our attack at $p = 0.45$ requires fewer informative relations than they do in the strictly easier setting $p = 0.40$. Since Damm *et al.* [6] restrict their noisy evaluation to ML-DSA-44, no direct comparison is possible for ML-DSA-65 and ML-DSA-87. However, based on our numerical improvements in the exact setting, we expect the reduction factors to be even larger there. As in the non-noisy case, those reduction factors directly reflect a reduction in the number of signatures to be collected.

7 Conclusion and Discussion

We have shown that the leakage-based attack on ML-DSA introduced by Liu *et al.* [4] and substantially refined by Damm *et al.* [5] becomes dramatically more potent when the linear-regression key-recovery step is replaced by an iterative local-optimization procedure. The original insight of these works—that rejection sampling can be "circumvented" by exploiting even a single leaked bit of the signing randomness per signature—already established a powerful attack paradigm. Now, our multi-tier hill-climbing algorithm exploits the extracted information far more efficiently, reducing the number of required signatures by roughly 37–68× in the exact setting and at least 23× across all noisy settings compared to the regression attack of Damm *et al.* [6].

Practical implications. In the most favorable exact setting our attack reliably recovers a 256-dimensional ML-DSA subkey from approximately 1 500 signatures for ML-DSA-44 (leakage index $j = 4$), 2 500 signatures for ML-DSA-87 ($j = 4$), and 3 000 signatures for ML-DSA-65 ($j = 5$). Each recovered subkey can then be used as a tool to reduce the complexity of recovering $\mathbf{s_1}$ [5, 26, 27, 28].

These signature counts are well within reach in many practical deployment scenarios. Organizations maintaining certificate authorities, code-signing infrastructure, or automated authentication services routinely produce thousands of signatures per day with the same key. An attacker with access to a side channel that reliably exposes a single bit of the signing randomness per signature could thus accumulate the required number of leaky signatures within hours to

days of normal operation. Implementors must therefore treat the protection of even individual bits of the signing randomness as security-critical and deploy corresponding defense strategies, such as a combination of higher-order masking schemes and meticulous hardware shielding.

Robustness to noisy leakage. The threat is not limited to idealized leakage scenarios. Our noisy-leakage extension (Section 5) demonstrates that the attack remains effective when the leaked bit is independently flipped with some noise rate p, which we could verify to be conceivable as high as 45% while still keeping our attack viable to be computed on a standard Apple notebook. Of course, a higher noise rate comes at the attacker's cost of requiring increasingly more leaky signatures.

Open directions. A particularly promising direction for future work is the adaptation of *belief propagation* (BP) to our setting. Hermelink *et al.* [23] recently introduced a generic framework based on *distribution hints* for recovering LWE secrets from side-channel information in the ML-KEM setting, proposing both a BP-based solver and a greedy solver. Their Theorem 3 establishes a precise connection between the two: The greedy solver is the algorithm obtained by replacing all sum distributions by their modes, collapsing probability distributions after each step, and updating only a limited number of variables at a time. In this sense, our hill-climbing algorithm is similar to such a "collapsed BP" specialized to two-sided interval constraints arising from ML-DSA's ILWE relations. These observations suggest that using a BP-based solver could further lessen the amount of leaky signatures required (especially in our noisy setting), potentially providing further alternatives to our solver and the ILP/NLP-based approach that is currently being investigated by Bashiri, Geuenich, and Mittmann [29].

Further, our noise model strictly assumes *symmetric* bit flips (Definition 3); characterizing the effect of asymmetric flip rates ($p_{0\to1} \neq p_{1\to0}$)—as they arise e.g. in several power analysis settings—remains open. Finally, side channels that expose more than a single bit, such as a noisy Hamming-weight estimate of the target coefficient, should be at least as informative and could possibly enter our verification and scoring framework as soft or multi-valued constraints.

Reproducibility of Results. Our attack's reference implementation as well as companion scripts are publicly available at `https://github.com/D-VAmpire/mldsa-hill-climbing` (also see Section 6).

Acknowledgements

We thank Maximilian J. Kramer, Jonas Thietke, Nicolai Kraus and Alexander May for insightful discussions. We also used Anthropic's Claude AI tools to generate code and review our texts. 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. Standards, N.I.O., 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
2. 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
3. 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
4. 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
5. 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.) PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part I. LNCS, vol. 15674, pp. 284–316. Springer, Heidelberg (2025). https://doi.org/10.1007/978-3-031-91820-9_10
6. 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).
7. Ulitzsch, V.Q., Marzougui, S., Tibouchi, M., Seifert, J.-P.: Profiling Side-Channel Attacks on Dilithium - A Small Bit-Fiddling Leak Breaks It All. In: Smith, B., Wu, H. (eds.) SAC 2022: 29th Annual International Workshop on Selected Areas in Cryptography. LNCS, vol. 13742, pp. 3–32. Springer, Heidelberg (2024). https://doi.org/10.1007/978-3-031-58411-4_1
8. Steffen, H.M., Land, G., Kogelheide, L.J., Güneysu, T.: Breaking and Protecting the Crystal: Side-Channel Analysis of Dilithium in Hardware. In: Johansson, T., Smith-Tone, D. (eds.) Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023, pp. 688–711. Springer, Cham, Switzerland, College Park, USA (2023). https://doi.org/10.1007/978-3-031-40003-2_25
9. Qiao, Z., Liu, Y., Zhou, Y., Shao, M., Sun, S.: When NTT Meets SIS: Efficient Side-channel Attacks on Dilithium and Kyber, Cryptology ePrint Archive, Report 2023/1866 (2023).
10. Bronchain, O., Azouaoui, M., ElGhamrawy, M., Renes, J., Schneider, T.: Exploiting Small-Norm Polynomial Multiplication with Physical Attacks Application to CRYSTALS-Dilithium. IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(2), 359–383 (2024). https://doi.org/10.46586/tches.v2024.i2.359-383
11. Espitau, T., Fouque, P.-A., Gérard, B., Tibouchi, M.: Loop-Abort Faults on Lattice-Based Fiat-Shamir and Hash-and-Sign Signatures. In: Avanzi, R., Heys, H.M. (eds.) SAC 2016: 23rd Annual International Workshop on Selected Areas in Cryptography. LNCS, vol. 10532, pp. 140–158. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-69453-5_8
12. Ulitzsch, V.Q., Marzougui, S., Bagia, A., Tibouchi, M., Seifert, J.-P.: Loop Aborts Strike Back: Defeating Fault Countermeasures in Lattice Signatures with ILP.

IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 367–392 (2023). https://doi.org/10.46586/tches.v2023.i4.367-392
13. Krahmer, E., Pessl, P., Land, G., Güneysu, T.: Correction Fault Attacks on Randomized CRYSTALS-Dilithium. IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(3), 174–199 (2024). https://doi.org/10.46586/tches.v2024.i3.174-199
14. Migliore, V., Gérard, B., Tibouchi, M., Fouque, P.-A.: Masking Dilithium - Efficient Implementation and Side-Channel Evaluation. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 2019: 17th International Conference on Applied Cryptography and Network Security. LNCS, vol. 11464, pp. 344–362. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-21568-2_17
15. Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) Advances in Cryptology – CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_27
16. Rivain, M., Prouff, E.: Provably Secure Higher-Order Masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2010. LNCS, vol. 6225, pp. 413–427. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_28
17. Azouaoui, M., Bronchain, O., Cassiers, G., Hoffmann, C., Kuzovkova, Y., Renes, J., Schneider, T., Schönauer, M., Standaert, F.-X., van Vredendaal, C.: Protecting Dilithium against Leakage Revisited Sensitivity Analysis and Improved Implementations. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 58–79 (2023). https://doi.org/10.46586/tches.v2023.i4.58-79
18. Coron, J.-S., Gérard, F., Trannoy, M., Zeitoun, R.: Improved Gadgets for the High-Order Masking of Dilithium. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 110–145 (2023). https://doi.org/10.46586/tches.v2023.i4.110-145
19. Covic, A., Ganji, F., Forte, D.: Circuit Masking: From Theory to Standardization, A Comprehensive Survey for Hardware Security Researchers and Practitioners, (2021). https://doi.org/10.48550/arXiv.2106.12714. arXiv: 2106.12714 [cs].
20. Hermelink, J., Ning, K.-C., Petri, R.: Finding and Protecting the Weakest Link - On Side-Channel Attacks on y in Masked ML-DSA. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology – CRYPTO 2025, Part V. LNCS, vol. 16004, pp. 3–37. Springer, Heidelberg (2025). https://doi.org/10.1007/978-3-032-01901-1_1
21. Damm, S., Fischer, A., May, A., Marzougui, S., Schwarz, L., Seidler, H., Seifert, J.-P., Thietke, J., Ulitzsch, V.Q.: Solving Concealed ILWE and Its Application for Breaking Masked Dilithium. In: Hanaoka, G., Yang, B.-Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part II. LNCS, vol. 16246, pp. 608–640. Springer, Heidelberg (2025). https://doi.org/10.1007/978-981-95-5096-8_19
22. Hermelink, J., Mårtensson, E., Samardjiska, S., Pessl, P., Rodosek, G.D.: Belief Propagation Meets Lattice Reduction: Security Estimates for Error-Tolerant Key Recovery from Decryption Errors. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(4), 287–317 (2023). https://doi.org/10.46586/tches.v2023.i4.287-317
23. Hermelink, J., Streit, S., Mårtensson, E., Petri, R.: A Generic Framework for Side-Channel Attacks Against LWE-Based Cryptosystems. In: Fehr, S., Fouque, P.-A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part VIII. LNCS, vol. 15608, pp. 3–32. Springer, Heidelberg (2025). https://doi.org/10.1007/978-3-031-91101-9_1

24. 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
25. 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
26. 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, Part II. LNCS, vol. 12171, pp. 329–358. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-56880-1_12
27. Dachman-Soled, D., Gong, H., Hanson, T., Kippen, H.: Revisiting Security Estimation for LWE with Hints from a Geometric Perspective. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part V. LNCS, vol. 14085, pp. 748–781. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-38554-4_24
28. May, A., Nowakowski, J.: Too Many Hints - When LLL Breaks LWE. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part IV. LNCS, vol. 14441, pp. 106–137. Springer, Heidelberg (2023). https://doi.org/10.1007/978-981-99-8730-6_4
29. Bashiri, K., Geuenich, J., Mittmann, J.: Exploiting Noisy Single-Bit Leakage in ML-DSA, Cryptology ePrint Archive, Paper 2026/580 (2026).