# Falcon Key Recovery from Joint Sampler Leakages

Youngchang Kim, Yongseong Park, Jaeho Jeon, Young-Sik Kim

Department of Electrical Engineering and Computer Science, DGIST, Daegu, Korea,
yckim0518@dgist.ac.kr, reo91004@dgist.ac.kr, dgwogh@dgist.ac.kr, ysk@dgist.ac.kr

**Abstract.** Falcon and the forthcoming FN-DSA standard based on it generate signatures via discrete Gaussian sampling, during which power consumption leaks information about secret-dependent intermediate values. While prior attacks exploit sampler leakages individually or in pairs, we present a unified key-recovery framework that fuses three observations from each SamplerZ call: the signed-exponent class $e(k)$ of the integer center, the random-sign bit $b$, and a multi-class proposal magnitude $z_0$. These fused observations define magnitude-preserving intervals or rays for accepted sampler outputs. Leveraging an exact public linear relation that holds in the absence of modular wraparound, we map these candidate sets directly into slab and halfspace constraints on the secret key. The resulting system is solved via smooth softplus interval regression and bounded integer reconstruction with public-key validation, entirely bypassing computationally intensive lattice reductions such as LLL or BKZ. For PQClean Falcon-512 running at `-O3` on an ARM Cortex-M4, our 17-class center classifier achieves 99.34% call-level accuracy on held-out test signatures from the profiling keys. The resulting attack recovers all 50 disjoint evaluation keys using at most 34 signatures (median 11), with a median post-classification key-recovery runtime of 17.54 s per key. Furthermore, on c-fn-dsa, where center conversion relies on a single constant-time `CLZ` instruction, a reduced-resolution 7-class classifier combined with proposal leakage still recovers 38 of 50 keys within 200 signatures. These findings demonstrate that instruction-level normalization alone cannot prevent key recovery, underscoring that side-channel countermeasures for FN-DSA must protect all sampler leakage channels jointly.

**Keywords:** Falcon · FN-DSA · side-channel analysis · power analysis · Gaussian sampling · SamplerZ · interval regression · key recovery

## 1 Introduction

Falcon [FHK+20] is a hash-and-sign signature scheme over NTRU lattices that samples short lattice vectors with fast Fourier sampling [DP16]. Its compact signatures and public keys make it attractive for constrained devices, and it is the basis of the forthcoming FN-DSA standard. Its signing procedure, however, relies on floating-point arithmetic and discrete Gaussian sampling, both of which have proven susceptible to power and electromagnetic analysis.

Side-channel attacks on Falcon have targeted floating-point multiplication in the pre-image computation [KA21, GMRR22], including implementations in which only the multiplication is masked [CCCY26], as well as fault injection [MHS+19], the Gaussian sampling performed during key generation [QA26], and the floating-point processing of the secret key during signing [KLL+26, LMDG25].

The integer Gaussian sampler SamplerZ, invoked 1,024 times per Falcon-512 signing attempt, is a separate source of leakage. Previous work has exploited the half-Gaussian sam-

ple $z_0$ [GMRR22, ZLYW23, LZY+25], the random-sign bit $b$ [ZLYW23, LZY+25], the sign of the sampler output [BKM26], the sign of the center [LTY26], and the signed-exponent class $e(k)$ of the integer center $k$ exposed by its conversion to floating point [ZWSY26]. These observations have been used individually or in pairs; a sign determines the direction of a constraint but not the scale of the secret [BKM26], and a center class confines $k$ only to a power-of-two range. Zhou et al. recover keys from the center class alone using a lattice-reduction search [ZWSY26]; in this work, we investigate whether proposal observations enable key recovery without relying on lattice reduction, even when fewer than 17 center classes are observable. To our knowledge, no prior work combines the center class with the proposal observations into magnitude-preserving constraints on each accepted sampler output.

This paper fuses $e(k)$, $b$, and a multi-class observation of $z_0$ from each accepted SamplerZ call through $R = k + b + (2b - 1)z_0$. We prove that ffSampling outputs a known coefficient of its integer output $Z_1$ at each position of the call sequence (Lemma 3), that $Z_1$ satisfies a linear relation with the secret polynomials whose coefficients are computable from public data whenever the signature coefficients do not wrap around modulo $q$, which holds for every accepted PQClean signature whose floating-point computation is exact (Lemma 2), and that the fused candidate set of every accepted output is an integer interval or ray that contains the true output whenever the observed classes are correct (Lemma 4). Consequently, correct observations yield slab or halfspace constraints on $(f, g)$, whereas erroneous observations introduce perturbed constraints that the reconstruction algorithm must tolerate. We solve the resulting system by softplus interval regression, followed by integer reconstruction at two candidate scales with bounded integer correction and a validity test that uses only the public key (Proposition 2).

We evaluate the method on an STM32F415 (ARM Cortex-M4) running two implementations compiled with `-O3`: PQClean Falcon-512[1], whose conversion normalizes with conditional shifts, and c-fn-dsa[2], a pre-standard FN-DSA-512 implementation whose Cortex-M4 conversion normalizes with a single `CLZ` instruction.

## 1.1 Contributions

- **Magnitude-aware fusion.** Lemmas 2–4 turn $e(k)$, $b$, and $z_0$ from one SamplerZ call into an integer interval or ray for its accepted output, which contains the output whenever the observed classes are correct, and into a slab or halfspace on $(f, g)$.
- **Reconstruction with public validation.** Softplus interval regression, two-scale integer reconstruction, and bounded integer correction recover the key without LLL or BKZ reduction, ensuring that every accepted candidate is a valid private key for the target public key (Proposition 2). While formal error bounds remain open, the resilience of the solver against misclassified constraints is demonstrated experimentally.
- **Measured key recovery on PQClean.** With five profiling keys, a 17-class $e(k)$ classifier reaches 99.34% call-level accuracy on held-out signatures of the profiling keys at `-O3`, and all 50 disjoint evaluation keys are recovered with at most 34 retained signatures (median 11). Furthermore, an observation-subset ablation study quantifies the respective contributions of $b$ and $z_0$ within the interval decoder.
- **Key recovery under CLZ-based conversion.** On c-fn-dsa, a reduced-resolution 7-class center classifier combined with $b$ and $z_0$ recovers 38 of 50 evaluation keys within 200 signatures in an exploratory evaluation across four candidate postprocessing configurations (Section 6).

[1] https://github.com/PQClean/PQClean
[2] https://github.com/pornin/c-fn-dsa

**Table 1:** Comparison of SamplerZ-related key-recovery studies on Falcon-512. Counts and success criteria follow the cited works and are not directly comparable.

| Reference | Leakage | Target | Profiling or characterization | Attack data | Key success |
|---|---|---|---|---|---|
| [GMRR22] | $z_0 = 0$ | Falcon ref. | Threshold SPA | $10^7$; $10^6$ with BKZ (est.) | Full key at $10^7$ |
| [ZLYW23] | $z_0, b$ | Falcon ref. | 70–100k-trace sweep | 45k filtered | $\approx 25\%$ (40 inst.) |
| [LZY+25] | $z_0, b$ | Falcon ref. | 70–110k-trace sweep | 6,500 signatures | 15/40 |
| [BKM26] | Output sign | Simulated | N/A | 100 signatures (avg.) | 100%; trials NR |
| [ZWSY26]† | $e(k)$ | PQClean, `-O0`/`-O3` | 680,000 traces | 20/56 signatures | 100/100 each |
| [LTY26]† | $\mathrm{sgn}(\mu)$ | Falcon ref. | Few for training; 100k for characterization | 20,000 traces | 58/100 |
| **This work** | $e(k), b, z_0$ | **PQClean**, `-O3` | 2,500 signatures, 5 keys[a] | $\leq 34$ signatures | **50/50** |
| **This work** | $e(k), b, z_0$ | **c-fn-dsa**, `-O3` | 2,500 signatures, 5 keys[a] | $\leq 200$ signatures | **38/50**[b] |

Devices: ELMO (Cortex-M0) and STM32F3 for [GMRR22], STM32F407 for [ZLYW23], and STM32F415 otherwise (all Cortex-M4 except ELMO); optimization levels are given where reported. †Preprint (September 2026). [a]1,280,000 labeled SamplerZ calls, of which 768,000 are used for training. [b]Most effective configuration among the four evaluated postprocessing policies (Section 6). NR: not reported; N/A: not applicable.

## 1.2 Related Work

Guerreau et al. recovered the secret basis from base-sampler leakage with a hidden-parallelepiped approach [GMRR22]. Zhang et al. improved the statistical treatment of this leakage and exploited the random-sign operation of the integer sampler [ZLYW23]. Lin et al. analyzed several leakage locations along the Gaussian-sampling path, recovered keys from $z_0$ and $b$ together, and proposed countermeasures [LZY+25]; unlike these works, we combine the center class with the proposal observations of each accepted output.

Noh et al. formulated key recovery from partial Gaussian information in GPV signatures as ordinary least-squares regression and used MAP decoding to correct classification errors in measured traces [NCHS24]. Brinkmann et al. cast sign leakage as learning halfspaces, solved by linear programming and logistic regression [BKM26]. Our formulation differs in that each observation constrains an exact integer coefficient of $Z_1$ through Lemma 2, magnitude is retained as an interval or ray rather than a sign, and the loss is a smooth penalty inspired by the two-limit regression formulation [RN75].

Floating-point conversion leakage has been exploited both for the secret key and for the sampler center. Kim et al. apply simple power analysis to conditional operations in floating-point conversion and addition [KLL+26], and Li et al. combine conversion and multiplication leakage in secret-key expansion, which dynamic signing designs execute at every signature [LMDG25]. Zhou et al. target the conversion of the integer center in SamplerZ, use the same 17 signed-exponent classes as our PQClean classifier, and recover keys from center leakage with a confusion-matrix maximum-likelihood estimator and a lattice-reduction search, with full recovery from 20 and 56 signatures at single-trace accuracies of 0.9997 (`-O0`) and 0.884 (`-O3`); they also report results for Falcon-1024 and for sign-only center leakage [ZWSY26]. Lin, Tibouchi, and Yu exploit the sign of the center and propose twisting the target by random roots of unity at every recursive call of ffSampling as a countermeasure [LTY26]. Both works represent concurrent independent studies on Falcon sampler leakage. Table 1 summarizes sampler-related key-recovery results.

# 2 Preliminaries

## 2.1 Falcon Signing, ffSampling, and SamplerZ

Let $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$, and $\mathcal{K} = \mathbb{Q}[x]/(x^n+1)$, with $n = 512$ and $q = 12289$. The secret key consists of short polynomials $f, g, F, G \in \mathcal{R}$ satisfying

$$fG - gF = q, \tag{1}$$

with $f$ invertible in $\mathcal{R}_q$; the public key is $h = gf^{-1} \in \mathcal{R}_q$. To sign a message $\mathsf{msg}$, the signer samples a nonce $\rho$ and hashes $\rho \,\|\, \mathsf{msg}$ to $c \in \mathcal{R}_q$; polynomial expressions involving $c$ use the representative in $\mathcal{R}$ with coefficients in $\{0, \dots, q-1\}$. The signer runs ffSampling with fresh randomness on the target $t = (t_0, t_1) = (-cF/q,\ cf/q) \in \mathcal{K}^2$, represented in the FFT domain, and maps the sampled pair $(Z_0, Z_1) \in \mathcal{R}^2$ through the secret basis:

$$(s_1, s_2) = \big(t - (Z_0, Z_1)\big) \begin{pmatrix} g & -f \\ G & -F \end{pmatrix}. \tag{2}$$

If $\|(s_1, s_2)\|_2^2 > \lfloor \beta^2 \rfloor$, where $\lfloor \beta^2 \rfloor = 34\,034\,726$ for $n = 512$, or if $s_2$ cannot be encoded, signing restarts; otherwise, the signer outputs header $\| \rho \|$ Encode$(s_2)$, and the verifier recomputes $s_1$ from $h$ and checks the same bound [FHK+20].

ffSampling [DP16] (Algorithm 1) recursively splits the target and samples the subtree associated with $t_1$ before the subtree associated with $t_0$. In the coefficient domain, Split writes $p(x) = p_E(x^2) + x\, p_O(x^2)$ and returns $(p_E, p_O)$, the even- and odd-indexed coefficients of $p$; Merge is its inverse. At a leaf, SamplerZ (Algorithm 2) is invoked on $t_0$ and then on $t_1$.

**Algorithm 1** ffSampling

**Require:** Tree $\mathcal{T}$, target $(t_0, t_1)$
**Ensure:** Sampled pair $(Z_0, Z_1)$
**if** $\mathcal{T}$ is a leaf with deviation $\sigma'_{\mathcal{T}}$ **then**
  $Z_0 \leftarrow \text{SamplerZ}(t_0, \sigma'_{\mathcal{T}})$
  $Z_1 \leftarrow \text{SamplerZ}(t_1, \sigma'_{\mathcal{T}})$
**else**
  $(x_0, x_1) \leftarrow \text{Split}(t_1)$
  $(y_0, y_1) \leftarrow \text{ffSampling}(\mathcal{T}_1, (x_0, x_1))$
  $Z_1 \leftarrow \text{Merge}(y_0, y_1)$
  $t'_0 \leftarrow t_0 + (t_1 - Z_1) L_{\mathcal{T}}$
  $(x_0, x_1) \leftarrow \text{Split}(t'_0)$
  $(y_0, y_1) \leftarrow \text{ffSampling}(\mathcal{T}_0, (x_0, x_1))$
  $Z_0 \leftarrow \text{Merge}(y_0, y_1)$
**end if**
**return** $(Z_0, Z_1)$

**Algorithm 2** SamplerZ

**Require:** Center $\mu$, deviation $\sigma'$
**Ensure:** Accepted integer $R$
$k \leftarrow \lfloor \mu \rceil$; $r \leftarrow \mu - k$
$\gamma \leftarrow \sigma_{\min}/\sigma'$
**repeat**
  $z_0 \leftarrow \text{BaseSampler}(\,)$
  $b \leftarrow \text{RandomBit}(\,)$
  $d \leftarrow b + (2b-1) z_0$
  $\varepsilon \leftarrow \frac{(d-r)^2}{2\sigma'^2} - \frac{z_0^2}{2\sigma_{\max}^2}$
**until** $\text{BerExp}(\varepsilon, \gamma) = 1$
$R \leftarrow k + d$
**return** $R$

**SamplerZ notation.** The scalar $z_0 \geq 0$ is a half-Gaussian sample and $b \in \{0, 1\}$ is the random-sign bit. For center $\mu$, the accepted output satisfies

$$k = \lfloor \mu \rceil, \qquad d = b + (2b-1) z_0, \qquad R = k + d. \tag{3}$$

The leaf deviation satisfies $\sigma_{\min} \leq \sigma' \leq \sigma_{\max} = 1.8205$; BaseSampler uses $\sigma_{\max}$, and $\sigma_{\min}$ depends on the parameter set [FHK+20]. BerExp$(\varepsilon, \gamma)$ returns 1 with probability $\gamma e^{-\varepsilon}$, up to the finite precision of its evaluation [FHK+20]. An invocation may iterate several times; the values $z_0$ and $b$ of the final, accepted iteration determine $R$.

### 2.2 Negacyclic Structure

Following [BKM26, Section 4], let $N(a) \in \mathbb{Z}^{n\times n}$ denote the matrix of multiplication by $a \in \mathcal{R}$ in the coefficient basis, so that $\mathrm{vec}(ap) = N(a)\,\mathrm{vec}(p) = N(p)\,\mathrm{vec}(a)$ for $a, p \in \mathcal{R}$. Since $N(a)N(p) = N(ap)$, negacyclic matrices commute. Polynomial identities therefore yield coefficient-wise linear relations.

### 2.3 Threat Model and Observation Scope

The adversary controls a profiling device of the same type that runs the same binary with keys of its choice and provides labels for $k$, $b$, and $z_0$; classifiers are learned from five profiling keys and applied without refitting to disjoint evaluation keys. For the target key, the adversary obtains the public key, the signed messages, the signatures, and the power traces of the signing executions, in which GPIO pulses mark the beginning and end of every SamplerZ call. These pulses reveal no operation inside SamplerZ, whose observation windows are located from the power trace alone, but they obviate the need for the adversary to segment a signing trace into calls; furthermore, the end of each pulse, which follows the accepted iteration, serves as the reference from which that iteration is located (Section 5.1). Both profiling and attack measurements use these instrumented binaries, characterizing the recovery performance under this delimited setting; extending the attack to uninstrumented binaries would require autonomous call segmentation (e.g., via [TBW+22]), as discussed in Section 7.1. A signing attempt that fails the norm or encoding check restarts with another 1,024 SamplerZ calls; we retain only executions whose first attempt is accepted and whose observation windows are successfully located, so that every observed call belongs to the released signature, and all signature counts refer to retained executions.

## 3 Sampler Leakages

Within one SamplerZ call, we observe the integer center $k$ and the proposal magnitude $z_0$ and proposal bit $b$ of the accepted iteration, with implementation-dependent resolution. Listing 1 marks the integer-to-floating-point conversions of $k$ and $z_0^2$ that we use for center and magnitude classification.

**(a) PQClean** SamplerZ **excerpt**

```
k = (int)fpr_floor(mu);
r = fpr_sub(mu, fpr_of(k));                                    /* (1) */
...
x = fpr_sub(x, fpr_mul(fpr_of(z0 * z0), fpr_inv_2sqrsigma0));  /* (2) */
```

**(b) c-fn-dsa Cortex-M4 assembly implementing `fpr_of32` (excerpt)**

```
    clz   r3, r2                @ leading zeros of the absolute value
    lsls  r2, r3                @ normalize by that count
    subw  r3, r3, #1053         @ exponent derived from the count
```

Listing 1: (a) Conversions of the center (1) and of $z_0^2$ (2) in the PQClean SamplerZ. (b) Cortex-M4 implementation of `fpr_of32`, which c-fn-dsa uses for the same two conversions. The center variable is renamed `k`; omissions and explanatory comments in (b) are added for clarity.

## 3.1 Integer-Center Leakage in PQClean

PQClean converts $k = \lfloor \mu \rceil$ through `fpr_of(k)`, which calls `fpr_scaled(k,0)`. Its `FPR_NORM64` macro normalizes the magnitude with a fixed sequence of conditional masked shifts. Kim et al. showed that conditional operations of this kind in floating-point conversion leak under simple power analysis [KLL+26], and Zhou et al. exploit the resulting signed-exponent leakage of the SamplerZ center [ZWSY26]. We use the label

$$e(k) = \begin{cases} 0, & k = 0, \\ \operatorname{sgn}(k) \min\{8,\, 1 + \lfloor \log_2 |k| \rfloor\}, & k \neq 0, \end{cases}$$

which has 17 classes: $k = 0$, $|k| = 1$, $|k| \in \{2, 3\}$, and successive signed power-of-two ranges, with $|k| \geq 128$ forming the two outer classes.

## 3.2 CLZ-Based Conversion in c-fn-dsa

In c-fn-dsa, the Cortex-M4 routine implementing `fpr_of32` (Listing 1(b)) normalizes the absolute value of its input with one `CLZ` instruction and a left shift by the resulting count, from which it also derives the exponent. The routine has no branches and no conditional normalization sequence of the kind exploited by Kim et al. [KLL+26], but the shift count, the exponent, and the operand remain data dependent. Our traces retain center-dependent information, which is plausibly attributable to these data-dependent values (although instruction-level trace validation was not conducted); however, our LDA-based classifiers (Section 6.1) do not resolve this leakage into 17 distinct classes. We therefore employ a hierarchical classifier that determines the sign of $k$ prior to assigning a magnitude group.

We group $e(k)$ into seven classes: $e(k) \leq -4$, $e(k) \in \{-3, -2\}$, $e(k) = -1$, $e(k) = 0$, $e(k) = 1$, $e(k) \in \{2, 3\}$, and $e(k) \geq 4$. The five central classes decode to the center sets $[-7, -2]$, $\{-1\}$, $\{0\}$, $\{1\}$, and $[2, 7]$. The two outer classes correspond to $k \leq -8$ and $k \geq 8$; the *tight* decoding uses these rays, and the *conservative* decoding uses $k \leq -1$ and $k \geq 1$, which remain correct for a misassigned center of the correct sign. Section 6 compares both decodings.

## 3.3 Proposal Leakages

**Proposal magnitude $z_0$.** BaseSampler returns the nonnegative proposal magnitude $z_0$, whose generation is known to leak [GMRR22, ZLYW23, LZY+25]. Marker (2) in Listing 1 provides an additional observation through the conversion of $z_0^2$. Using profiling data, we classify $z_0$ into $\{0\}$, $\{1\}$, $\{2, 3\}$, and $\{4, \ldots, 7\}$; this four-class model thus assumes $z_0 \leq 7$. A value $z_0 \geq 8$ cannot be represented by any class and yields an incorrect observation, which is treated as noise by the robust reconstruction algorithm. Such values are no more likely in the accepted iteration than in a single BaseSampler output:

**Lemma 1** (Accepted proposal magnitudes)**.** *Fix $\sigma_{\min} \leq \sigma' \leq \sigma_{\max}$ and $r \in [0, 1)$, and let $p$ be the output distribution of* BaseSampler*. If every iteration of Algorithm 2 draws $z_0 \sim p$ and an independent uniform $b$ and is accepted with probability exactly $\gamma e^{-\varepsilon}$, then $\Pr[z_0 \geq t \mid \textit{accepted}] \leq \Pr_p[z_0 \geq t]$ for every integer $t \geq 1$.*

*Proof.* Let $\varepsilon_b(z)$ denote $\varepsilon$ for $z_0 = z$ and bit $b$. Expanding $d = b + (2b - 1)z$ gives $\varepsilon_b(z) = \alpha z^2 + c_b z + \frac{c_b^2 \sigma'^2}{2}$ with $\alpha = \frac{1}{2\sigma'^2} - \frac{1}{2\sigma_{\max}^2} \geq 0$, $c_1 = \frac{1-r}{\sigma'^2} \geq 0$, and $c_0 = \frac{r}{\sigma'^2} \geq 0$. Hence $\varepsilon_b \geq 0$ is nondecreasing on $\mathbb{Z}_{\geq 0}$, and the acceptance probability $a(z) = \frac{\gamma}{2}\left(e^{-\varepsilon_0(z)} + e^{-\varepsilon_1(z)}\right)$ of a proposal $z$ is positive, nonincreasing, and at most one, as $\gamma \leq 1$. Iterations are independent and identically distributed, so the accepted $z_0$ equals $z$ with probability $p(z)a(z)/\sum_{z'} p(z')a(z')$. Let $P = \Pr_p[z_0 \geq t]$, $A_{\geq} = \sum_{z \geq t} p(z)a(z) \leq a(t-1)P$, and $A_{<} = \sum_{z<t} p(z)a(z) \geq a(t-1)(1-P)$. If $A_{\geq} = 0$, the claim is immediate; otherwise, since

$u/(u+v)$ increases in $u>0$ and decreases in $v\geq 0$, $A_{\geq}/(A_{\geq}+A_{<}) \leq a(t-1)P/\big(a(t-1)P + a(t-1)(1-P)\big) = P$. □

An accepted call thus carries an unrepresentable proposal with probability at most $\Pr_p[z_0 \geq 8] \approx 2.49 \times 10^{-5}$, a conservative bound that does not account for the additional filtering of executions (Section 2.3); none of the 1,280,000 retained PQClean evaluation calls exhibited such a value (Section 5.2).

**Random-sign bit *b*.** Prior analyses have examined leakage from the random-sign operation [ZLYW23, LZY+25]. In Algorithm 2, $b$ selects between the two proposal branches, $d=-z_0$ for $b=0$ and $d=z_0+1$ for $b=1$. We use a two-class classifier for $b$.

# 4 Key Recovery from Joint Sampler Leakages

Sections 4.1–4.3 turn classifier outputs into linear constraints that $(f,g)$ satisfies whenever the observations are correct; Sections 4.4–4.5 solve them while tolerating incorrect observations.

## 4.1 Public Linear Relation

**Lemma 2** (Public linear relation). *Let $f,g,F,G \in \mathcal{R}$ satisfy (1), let $B = \begin{pmatrix} g & -f \\ G & -F \end{pmatrix}$, and let $(s_1,s_2) \in \mathcal{R}^2$ be the unreduced vector obtained from $(Z_0,Z_1)\in\mathcal{R}^2$ by (2).*

*(i) $(Z_0,Z_1) = (c-s_1,\,-s_2)\,B^{-1}$. In particular, $qZ_1 = (c-s_1)f - s_2 g$ and $qZ_0 = s_2 G - (c-s_1)F$ hold in $\mathcal{R}$.*

*(ii) Let $\bar{s}_2 \in \mathcal{R}$ satisfy $\bar{s}_2 \equiv s_2 \pmod q$, and let $\bar{s}_1 = \mathrm{center}_q(c-\bar{s}_2 h \bmod q)$, where $\mathrm{center}_q$ maps each residue to its representative in $(-q/2, q/2]$. Then $\bar{s}_1 \equiv s_1 \pmod q$, and writing $s_1 = \bar{s}_1 + q w_1$ and $s_2 = \bar{s}_2 + q w_2$,*

$$(c-\bar{s}_1)f - \bar{s}_2 g = q\,(Z_1 + w_1 f + w_2 g).$$

*Hence the public quantity $(c-\bar{s}_1)f - \bar{s}_2 g$ equals $qZ_1$ if and only if $w_1 f + w_2 g = 0$.*

*(iii) $w_1 = w_2 = 0$ if and only if $\bar{s}_2 = s_2$ and $\max_i |s_{1,i}| < q/2$. In that case, the public quantity in (ii) equals $qZ_1$.*

*(iv) If every coefficient of $\bar{s}_1$ and $\bar{s}_2$ has absolute value at most $B_\infty$, then $w_1 \neq 0$ (resp. $w_2 \neq 0$) implies that some coefficient of $s_1$ (resp. $s_2$) has absolute value at least $q - B_\infty$.*

*Proof.* (i) Because $\mathcal{K}$ is commutative, $t_0 g + t_1 G = (-cFg + cfG)/q = c(fG-gF)/q = c$ and $-t_0 f - t_1 F = (cFf - cfF)/q = 0$. Hence $tB = (c,0)$, and (2) gives $(s_1,s_2) = (c,0) - (Z_0,Z_1)B$, that is, $(Z_0,Z_1)B = (c-s_1,\,-s_2)$. The determinant of $B$ is $-gF+fG = q$, which is invertible in $\mathcal{K}$, and $B^{-1} = q^{-1}\left(\begin{smallmatrix} -F & f \\ -G & g \end{smallmatrix}\right)$. Multiplying $(c-s_1,-s_2)$ by $B^{-1}$ gives $qZ_0 = -(c-s_1)F + s_2 G$ and $qZ_1 = (c-s_1)f - s_2 g$. All polynomials involved lie in $\mathcal{R}$, so both identities hold in $\mathcal{R}$.

(ii) By (i), $c - s_1 = Z_0 g + Z_1 G$ and $s_2 = Z_0 f + Z_1 F$. Modulo $q$, where $f$ is invertible and $h \equiv g f^{-1}$,

$$s_1 + s_2 h \equiv c - Z_0 g - Z_1 G + (Z_0 f + Z_1 F) g f^{-1} = c - Z_1 f^{-1}(fG - gF) \equiv c \pmod q.$$

Since $\bar{s}_2 \equiv s_2$, it follows that $c - \bar{s}_2 h \equiv c - s_2 h \equiv s_1 \pmod q$, so $\bar{s}_1 \equiv s_1 \pmod q$, and the integer polynomials $w_1, w_2$ are well defined. Substituting $s_1 = \bar{s}_1 + q w_1$ and $s_2 = \bar{s}_2 + q w_2$

into $qZ_1 = (c - s_1)f - s_2 g$ gives the stated identity, and the equivalence follows because $qx = 0$ implies $x = 0$ in $\mathcal{R}$.

(iii) By definition, $w_2 = 0$ if and only if $\bar{s}_2 = s_2$. Since $\bar{s}_1 \equiv s_1$ and $\bar{s}_1$ lies in $(-q/2, q/2]$, which contains exactly one representative of each residue class, $w_1 = 0$ if and only if every $s_{1,i}$ lies in this interval, that is, $\max_i |s_{1,i}| < q/2$ because $q$ is odd.

(iv) If $w_{1,i} \neq 0$, then $|s_{1,i}| = |\bar{s}_{1,i} + q w_{1,i}| \geq q|w_{1,i}| - |\bar{s}_{1,i}| \geq q - B_\infty$; the same argument applies to $s_2$. □

**Unreduced and reduced representatives.** The attack computes $\bar{s}_1$ and $\bar{s}_2$ from public data. By Lemma 2(ii), the public quantity yields $Z_1$ exactly when $w_1 f + w_2 g = 0$, which we call the *relation condition*; it holds in particular whenever $w_1 = w_2 = 0$, and whether the latter holds depends on how the implementation forms $(s_1, s_2)$. PQClean computes $s_1 = c - (Z_0 g + Z_1 G)$ and $s_2 = Z_0 f + Z_1 F$ in floating point, rounds them to integers, applies the $\ell_2$ bound to these unreduced integers, and encodes this $s_2$. If the rounding returns the exact integer products, which we assume (Section 4.2 checks the resulting relation on recorded signatures), then $\bar{s}_2 = s_2$, and every accepted signature satisfies $\max_i |s_{1,i}| \leq \|(s_1, s_2)\|_2 \leq \sqrt{\lfloor \beta^2 \rfloor} < q/2$ because $\lfloor \beta^2 \rfloor = 34\,034\,726 < q^2/4$ for $n = 512$; by Lemma 2(iii), $w_1 = w_2 = 0$ for every accepted PQClean signature. The c-fn-dsa build evaluated in Section 6 (`FNDSA_ASM_CORTEXM4`) instead computes $s_1$ and $s_2$ modulo $q$ in the NTT domain and accepts a signature only if their centered representatives satisfy the $\ell_2$ bound and have infinity norm at most $B_\infty = 840$ (constant `B_INF` in `inner.h`, checked for $s_1$ in `sign_core.c` and for $s_2$ by the encoder in `codec.c`). Its published $s_2$ is therefore $\bar{s}_2$, and $w_1 = w_2 = 0$ is not enforced. By Lemma 2(iv), a nonzero coefficient of $w_1$ or $w_2$ requires an unreduced coefficient of absolute value at least $q - B_\infty = 11{,}449$, about $69\sigma$ in an idealized Gaussian model of the signature coefficients with $\sigma = 165.74$ [FHK+20]. We directly verified the relation condition across all c-fn-dsa evaluation signatures (Section 6.1).

*Remark* 1 (Scope). For $n = 1024$, $\lfloor \beta^2 \rfloor = 70\,265\,242$ exceeds $q^2/4$, so the $\ell_2$ bound alone does not imply $\max_i |s_{1,i}| < q/2$, and the coefficients of $f$ and $g$ are bounded by 15 rather than 31 [FHK+20]; we restrict our analysis and experiments to $n = 512$. The identity for $Z_0$ involves the unknown polynomials $F$ and $G$; we therefore use only the $n$ sampler outputs that form $Z_1$.

For signing invocation $j$, let $\mathbf{u}_{j,i}^\mathsf{T}$ and $\mathbf{v}_{j,i}^\mathsf{T}$ denote the $i$-th rows of $N(c_j - \bar{s}_{1,j})$ and $-N(\bar{s}_{2,j})$, which are computable from the public key, the message, and the signature. With $\mathbf{f} = \mathrm{vec}(f)$ and $\mathbf{g} = \mathrm{vec}(g)$, coefficient $i$ of Lemma 2(ii) under the relation condition reads

$$\mathbf{u}_{j,i}^\mathsf{T} \mathbf{f} + \mathbf{v}_{j,i}^\mathsf{T} \mathbf{g} = q\,(Z_{1,j})_i. \tag{4}$$

Each row involves all $2n$ secret coefficients.

## 4.2 Output-to-Coefficient Map

**Lemma 3** (Output-to-coefficient map)**.** *Apply Algorithm 1 with the coefficient-domain Split and Merge to a target whose two components have $m = 2^\zeta$ coefficients, and index its $2m$* SamplerZ *outputs in execution order by $\ell = \sum_{t=0}^{\zeta} \ell_t 2^t$ with $\ell_t \in \{0, 1\}$. The output with index $\ell$ equals $(Z_\chi)_\iota$, where $\chi = \ell_0$ and $\iota = 0$ if $\zeta = 0$, and, if $\zeta \geq 1$,*

$$\chi = 1 - \ell_\zeta, \qquad \iota = \ell_0\, 2^{\zeta-1} + \sum_{t=1}^{\zeta-1} (1 - \ell_t)\, 2^{\zeta-1-t}. \tag{5}$$

*In particular, for $n = 2^\zeta \geq 2$, the first $n$ outputs are the coefficients of $Z_1$, and the output with index $\ell < n$ equals $(Z_1)_{\pi_n(\ell)}$ with*

$$\pi_n(\ell) = \mathrm{bitrev}_\zeta(\ell) \oplus (n/2 - 1), \tag{6}$$

*where* $\mathrm{bitrev}_\zeta$ *reverses the* $\zeta$*-bit binary representation of its argument.*

*Proof.* We argue by induction on $\zeta$. For $\zeta = 0$, both components are scalars, and Lines 2–3 of Algorithm 1 output $Z_0$ and then $Z_1$; the output with index $\ell = \ell_0$ is therefore $(Z_{\ell_0})_0$.

Let $\zeta \geq 1$ and assume the claim for $\zeta - 1$. Lines 5–7 apply Algorithm 1 to $\mathrm{Split}(t_1) = (x_0, x_1)$, whose components have $2^{\zeta-1}$ coefficients, and this call produces the first $2 \cdot 2^{\zeta-1} = m$ outputs. By the induction hypothesis, its output with index $\ell' \in [0, m)$ equals $(y_{\chi'})_{\iota'}$, where $\chi'$ and $\iota'$ are given by the claim for $\zeta - 1$ applied to $\ell'$. Since Merge sets $Z_1(x) = y_0(x^2) + x\, y_1(x^2)$, we have $(y_{\chi'})_{\iota'} = (Z_1)_{2\iota'+\chi'}$. Line 8 changes the centers used by the subsequent call but not the order of outputs, and Lines 9–11 produce the next $m$ outputs in the same manner for $Z_0$. Hence the output with index $\ell$ is a coefficient of $Z_1$ if $\ell_\zeta = 0$ and of $Z_0$ if $\ell_\zeta = 1$, that is, $\chi = 1 - \ell_\zeta$, and its coefficient index is $\iota = 2\iota' + \chi'$ evaluated at $\ell' = \ell \bmod m = \sum_{t=0}^{\zeta-1} \ell_t 2^t$.

If $\zeta = 1$, then $\chi' = \ell_0$ and $\iota' = 0$, so $\iota = \ell_0$, in agreement with (5). If $\zeta \geq 2$, then $\chi' = 1 - \ell_{\zeta-1}$ and $\iota' = \ell_0\, 2^{\zeta-2} + \sum_{t=1}^{\zeta-2} (1 - \ell_t)\, 2^{\zeta-2-t}$, so $2\iota' + \chi' = \ell_0\, 2^{\zeta-1} + \sum_{t=1}^{\zeta-2} (1 - \ell_t)\, 2^{\zeta-1-t} + (1 - \ell_{\zeta-1})$, which is (5) because the last term is the summand $t = \zeta - 1$.

For the final statement, let $n = 2^\zeta$ and $0 \leq \ell < n$, so that $\ell_\zeta = 0$ and $\chi = 1$. Since $\mathrm{bitrev}_\zeta(\ell) = \sum_{t=0}^{\zeta-1} \ell_t 2^{\zeta-1-t}$, the index $\iota$ agrees with $\mathrm{bitrev}_\zeta(\ell)$ in bit position $\zeta - 1$ (the term $t = 0$) and is its complement in bit positions $0, \ldots, \zeta - 2$ (the terms $t = 1, \ldots, \zeta - 1$). Complementing these $\zeta - 1$ low-order bits is the exclusive-or with $2^{\zeta-1} - 1 = n/2 - 1$. □

PQClean and c-fn-dsa perform Split and Merge in the FFT domain, which represents the same decomposition [FHK+20], and inline the lowest recursion levels in the order of Algorithm 1: every level processes the $t_1$ half first, and the lowest level samples coefficient 0 before coefficient 1. For PQClean, we also confirmed the map experimentally: all retained outputs of 50 signatures of one evaluation key equal $(Z_1)_{\pi_n(\ell)}$ computed from the signatures through Lemma 2(ii). This check covers 25,600 outputs and confirms, for these signatures, that the public quantity of Lemma 2(ii) equals $qZ_1$; it does not certify the rounding itself, since an error $a(g, -f)$ in $(s_1, s_2)$ only replaces $Z_0$ by $Z_0 - a$.

## 4.3 Fused Output Intervals and Linear Constraints

Let $\hat{c}_k$, $\hat{c}_b$, and $\hat{c}_{z_0}$ denote the predicted classes for $e(k)$, $b$, and $z_0$, decoded into candidate sets $D_k = \mathrm{Dec}_k(\hat{c}_k)$, $D_b = \mathrm{Dec}_b(\hat{c}_b)$, and $D_{z_0} = \mathrm{Dec}_{z_0}(\hat{c}_{z_0})$. For PQClean, $D_k = \{k' \in \mathbb{Z} : e(k') = \hat{c}_k\}$; for c-fn-dsa, $\mathrm{Dec}_k$ maps each grouped class to the interval or ray of Section 3.2. For both implementations, $D_b = \{\hat{c}_b\}$, and $\mathrm{Dec}_{z_0}$ maps the four magnitude classes to $\{0\}$, $\{1\}$, $\{2, 3\}$, and $\{4, \ldots, 7\}$. The fused candidate set of the accepted output is

$$D_R = \{k' + b' + (2b' - 1)z' : k' \in D_k,\ b' \in D_b,\ z' \in D_{z_0}\}.$$

For signing invocation $j$ and output index $\ell$, we write $D_{R,j,\ell}$ and let $a^L_{j,\ell}, a^U_{j,\ell} \in \{0, 1\}$ indicate whether this set has a finite lower or upper endpoint, given by $L_{j,\ell} = \min D_{R,j,\ell}$ and $U_{j,\ell} = \max D_{R,j,\ell}$. Every decoded center set has at least one finite endpoint, and so does every fused set. The implementation propagates endpoints and indicators directly and never enumerates an unbounded class. We call an observation *correct* if $k \in D_k$, $b \in D_b$, and $z_0 \in D_{z_0}$; with the four-class $z_0$ model, this requires $z_0 \leq 7$.

**Lemma 4** (Fused candidate sets)**.** *Let* $D_k = I \cap \mathbb{Z}$ *for a nonempty closed interval or closed ray* $I \subset \mathbb{R}$ *with integer endpoints,* $D_b = \{\hat{b}\}$ *with* $\hat{b} \in \{0, 1\}$*, and* $D_{z_0} = \{z_{\mathrm{lo}}, \ldots, z_{\mathrm{hi}}\}$ *with* $0 \leq z_{\mathrm{lo}} \leq z_{\mathrm{hi}}$*.*

*(i) If the observation is correct, then the accepted output* $R$ *lies in* $D_R$*.*

*(ii)* $D_R = \mathrm{conv}(D_R) \cap \mathbb{Z}$*.*

*(iii) For every $x \in \mathbb{R} \setminus (\mathbb{Z} + \frac{1}{2})$, $\mathrm{round}(x) \in D_R$ if and only if $x \geq L - \frac{1}{2}$ (when $a^L = 1$) and $x \leq U + \frac{1}{2}$ (when $a^U = 1$).*

*Proof.* (i) follows from (3) and the definition of $D_R$. For (ii), let $\Delta = \{\hat{b}+(2\hat{b}-1)z : z \in D_{z_0}\}$. If $\hat{b} = 1$, then $\Delta = \{1 + z_{\mathrm{lo}}, \ldots, 1 + z_{\mathrm{hi}}\}$; if $\hat{b} = 0$, then $\Delta = \{-z_{\mathrm{hi}}, \ldots, -z_{\mathrm{lo}}\}$. In both cases $\Delta = \{\delta_{\mathrm{lo}}, \ldots, \delta_{\mathrm{hi}}\}$ consists of consecutive integers, and $D_R = D_k + \Delta$ is a Minkowski sum.

The inclusion $D_R \subseteq \mathrm{conv}(D_R) \cap \mathbb{Z}$ is immediate. Conversely, let $m$ be an integer in $\mathrm{conv}(D_R)$; then $m \geq \underline{k} + \delta_{\mathrm{lo}}$ if $D_k$ has a finite minimum $\underline{k}$, and $m \leq \bar{k} + \delta_{\mathrm{hi}}$ if $D_k$ has a finite maximum $\bar{k}$. Suppose first that $\bar{k}$ exists, and set $k' = \min(\bar{k}, m - \delta_{\mathrm{lo}})$ and $\delta = m - k'$. If $k' = \bar{k}$, then $\bar{k} \leq m - \delta_{\mathrm{lo}}$ gives $\delta \geq \delta_{\mathrm{lo}}$, and $m \leq \bar{k} + \delta_{\mathrm{hi}}$ gives $\delta \leq \delta_{\mathrm{hi}}$. If $k' = m - \delta_{\mathrm{lo}} < \bar{k}$, then $\delta = \delta_{\mathrm{lo}}$, and $k' \geq \underline{k}$ whenever $\underline{k}$ exists. If $\bar{k}$ does not exist, then $I = [\underline{k}, \infty)$ with $\underline{k}$ finite; set $k' = m - \delta_{\mathrm{lo}} \geq \underline{k}$ and $\delta = \delta_{\mathrm{lo}}$. In every case, $k' \in D_k$ and $\delta \in \Delta$, so $m = k' + \delta \in D_R$.

(iii) For $x \notin \mathbb{Z} + \frac{1}{2}$, $|x - \mathrm{round}(x)| < \frac{1}{2}$, so $\mathrm{round}(x) \geq L$ if and only if $x \geq L - \frac{1}{2}$, and $\mathrm{round}(x) \leq U$ if and only if $x \leq U + \frac{1}{2}$. By (ii), the integer $\mathrm{round}(x)$ lies in $D_R$ exactly when it satisfies the finite endpoint conditions. □

Combining (4) with Lemma 3, the output with index $\ell < n$ of invocation $j$ satisfies $R_{j,\ell} = (Z_{1,j})_{\pi_n(\ell)}$. If the observation is correct, then $R_{j,\ell} \in D_{R,j,\ell}$ by Lemma 4(i), and the true secret satisfies

$$\begin{aligned} qL_{j,\ell} &\leq \mathbf{u}_{j,\pi_n(\ell)}^{\mathsf{T}}\mathbf{f} + \mathbf{v}_{j,\pi_n(\ell)}^{\mathsf{T}}\mathbf{g} \quad \text{if } a_{j,\ell}^L = 1, \\ qU_{j,\ell} &\geq \mathbf{u}_{j,\pi_n(\ell)}^{\mathsf{T}}\mathbf{f} + \mathbf{v}_{j,\pi_n(\ell)}^{\mathsf{T}}\mathbf{g} \quad \text{if } a_{j,\ell}^U = 1. \end{aligned} \tag{7}$$

Conversely, under the hypotheses of Lemma 4, a secret that satisfies (7) yields an integer output in $\mathrm{conv}(D_{R,j,\ell}) \cap \mathbb{Z} = D_{R,j,\ell}$, so the constraint loses no information relative to the candidate set. An incorrect observation may exclude the true output, in which case the true secret violates (7). Each constraint is a slab if both indicators equal one, an equality if, in addition, $L_{j,\ell} = U_{j,\ell}$, and a halfspace if only one endpoint is finite. For a finite $D_k$, the Minkowski sum in the proof of Lemma 4 has $|D_R| = |D_k| + |D_{z_0}| - 1 \leq |D_k| + 3$ elements, whereas marginalizing both $b$ and $z_0$ over their modeled support gives $|D_k| + 15$; for $|k| \leq 1$, the proposal observations thus narrow the candidate set from 16 integers to at most 4.

## 4.4 Softplus Interval Regression

Unlike a sign observation, which constrains the direction of the secret but not its scale [BKM26], an interval constraint retains magnitude information. For a candidate $\theta = [\tilde{f}; \tilde{g}] \in \mathbb{R}^{2n}$, define the normalized prediction

$$x_{j,\ell}(\theta) = \frac{\mathbf{u}_{j,\pi_n(\ell)}^{\mathsf{T}}\tilde{f} + \mathbf{v}_{j,\pi_n(\ell)}^{\mathsf{T}}\tilde{g}}{q}. \tag{8}$$

For nearest-integer rounding, each admissible integer $m \in D_{R,j,\ell}$ corresponds to the cell $[m - \frac{1}{2}, m + \frac{1}{2}]$, and by Lemma 4 the union of these cells is the interval or ray bounded by $L_{j,\ell} - \frac{1}{2}$ and $U_{j,\ell} + \frac{1}{2}$. We penalize each finite boundary by a softplus term,

$$\mathcal{L}_{j,\ell}(\theta) = a_{j,\ell}^L \,\mathrm{softplus}\Big(L_{j,\ell} - \tfrac{1}{2} - x_{j,\ell}(\theta)\Big) + a_{j,\ell}^U \,\mathrm{softplus}\Big(x_{j,\ell}(\theta) - U_{j,\ell} - \tfrac{1}{2}\Big), \tag{9}$$

and minimize

$$\mathcal{J}(\theta) = \frac{1}{M}\sum_{j,\ell} w_{j,\ell}\,\mathcal{L}_{j,\ell}(\theta) + \frac{\lambda}{2}\,\|\theta\|_2^2, \tag{10}$$

where $\mathrm{softplus}(t) = \log(1 + e^t)$, $M$ is the number of retained observations, $w_{j,\ell} > 0$ are observation weights, and $\lambda \geq 0$ controls $\ell_2$ regularization. The model builds upon the

framework of two-limit regression, where an observation confines a latent linear predictor to an interval [RN75], substituting a smooth penalty for the likelihood function.

**Proposition 1** (Convexity, minimizers, and coercivity)**.** *Write* $x_{j,\ell}(\theta) = \langle \xi_{j,\ell}, \theta \rangle$ *with* $\xi_{j,\ell} = (\mathbf{u}_{j,\pi_n(\ell)}; \mathbf{v}_{j,\pi_n(\ell)})/q$*, where every retained row has* $a^L_{j,\ell} = 1$*,* $a^U_{j,\ell} = 1$*, or both, and* $w_{j,\ell} > 0$*.*

*(i) If* $\lambda > 0$*, then* $\mathcal{J}$ *is strongly convex and has a unique global minimizer.*

*(ii) Let* $\mathcal{J}_0(\theta) = \frac{1}{M}\sum_{j,\ell} w_{j,\ell}\mathcal{L}_{j,\ell}(\theta)$ *and*

$$\mathcal{C} = \{\nu \in \mathbb{R}^{2n} : \langle \xi_{j,\ell}, \nu \rangle \geq 0 \text{ if } a^L_{j,\ell} = 1, \quad \langle \xi_{j,\ell}, \nu \rangle \leq 0 \text{ if } a^U_{j,\ell} = 1\}.$$

*Then* $\mathcal{J}_0$ *is coercive if and only if* $\mathcal{C} = \{0\}$*. In particular,* $\mathcal{C} = \{0\}$ *whenever the vectors* $\xi_{j,\ell}$ *of the two-sided rows span* $\mathbb{R}^{2n}$*.*

*Proof.* (i) For $t \in \mathbb{R}$, $\mathrm{softplus}''(t) = e^t/(1+e^t)^2 > 0$, so softplus is convex; it is also increasing and positive. Each $x_{j,\ell}$ is linear in $\theta$, so each boundary term in (9) is convex and nonnegative, and so is $\mathcal{J}_0$. For $\lambda > 0$, $\mathcal{J} = \mathcal{J}_0 + \frac{\lambda}{2}\|\theta\|_2^2$ is $\lambda$-strongly convex and satisfies $\mathcal{J}(\theta) \geq \frac{\lambda}{2}\|\theta\|_2^2$. The sublevel set $\{\theta : \mathcal{J}(\theta) \leq \mathcal{J}(0)\}$ is therefore nonempty and compact, so the continuous function $\mathcal{J}$ attains a global minimum on it, which is unique by strict convexity.

(ii) Let $\psi(\nu) = \frac{1}{M}\sum_{j,\ell} w_{j,\ell}\big(a^L_{j,\ell}\max(-\langle \xi_{j,\ell}, \nu\rangle, 0) + a^U_{j,\ell}\max(\langle \xi_{j,\ell}, \nu\rangle, 0)\big)$. This function is continuous, nonnegative, and positively homogeneous, and, because every $w_{j,\ell} > 0$, $\psi(\nu) = 0$ if and only if $\nu \in \mathcal{C}$. Suppose that $\mathcal{C} = \{0\}$. Then $\psi$ attains a minimum $\psi_{\min} > 0$ on the unit sphere, so $\psi(\theta) \geq \psi_{\min}\|\theta\|_2$. Since $\mathrm{softplus}(u) \geq \max(u, 0)$ and $\max(u+a, 0) \geq \max(u, 0) - |a|$, each loss term satisfies $\mathcal{L}_{j,\ell}(\theta) \geq a^L_{j,\ell}\max(-x_{j,\ell}(\theta), 0) + a^U_{j,\ell}\max(x_{j,\ell}(\theta), 0) - c_{j,\ell}$ with $c_{j,\ell} = a^L_{j,\ell}|L_{j,\ell} - \frac{1}{2}| + a^U_{j,\ell}|U_{j,\ell} + \frac{1}{2}|$. Hence $\mathcal{J}_0(\theta) \geq \psi_{\min}\|\theta\|_2 - \bar{c}$ for a constant $\bar{c}$, and $\mathcal{J}_0$ is coercive. Conversely, let $\nu \in \mathcal{C} \setminus \{0\}$. Because softplus is increasing, each term $\mathrm{softplus}(L_{j,\ell} - \frac{1}{2} - t\langle \xi_{j,\ell}, \nu\rangle)$ with $a^L_{j,\ell} = 1$ and each term $\mathrm{softplus}(t\langle \xi_{j,\ell}, \nu\rangle - U_{j,\ell} - \frac{1}{2})$ with $a^U_{j,\ell} = 1$ is nonincreasing in $t \geq 0$. Thus $\mathcal{J}_0(t\nu) \leq \mathcal{J}_0(0)$ for all $t \geq 0$ while $\|t\nu\|_2 \to \infty$, and $\mathcal{J}_0$ is not coercive. Finally, a two-sided row forces $\langle \xi_{j,\ell}, \nu\rangle = 0$ for every $\nu \in \mathcal{C}$, so $\mathcal{C} = \{0\}$ if the vectors $\xi_{j,\ell}$ of these rows span $\mathbb{R}^{2n}$. □

Proposition 1 concerns only the optimization geometry. It guarantees neither the optimality of the true secret nor exact integer recovery: wide intervals admit many integer solutions, the softplus terms favor interval midpoints, and incorrect observations shift the minimizer. When $\mathcal{C} \neq \{0\}$ and $\lambda$ is small, the minimizer can be displaced along $\mathcal{C}$, changing its scale, which motivates the scale candidates of Section 4.5. Success probabilities and signature counts are established empirically. We minimize (10) over the box $\Theta = [-31, 31]^{2n}$, on which a minimizer exists by compactness and is unique for $\lambda > 0$ by strict convexity, with L-BFGS-B and the settings of Section 5.3.

## 4.5 Integer Reconstruction and Public Validation

**Scale candidates.** The reconstruction considers two scales in order: the unit scale $\alpha = 1$ and an integer-distance scale adapted from Brinkmann et al. [BKM26, Section 3.2, Remark 6],

$$\alpha_{\mathrm{heur}} = \arg\min_{\alpha \in \mathcal{A}(\tilde{f})} \left\| \alpha\tilde{f} - \mathrm{round}(\alpha\tilde{f}) \right\|_2,$$

where $\mathcal{A}(\tilde{f})$ contains the positive ratios $a/\tilde{f}_i$ with $0 \leq i < n$, $\tilde{f}_i \neq 0$, and $a \in \{\pm 1, \ldots, \pm 15\}$. The range $|a| \leq 15$ covers about 3.7 standard deviations of the secret coefficients, whose standard deviation is $1.17\sqrt{q/(2n)} \approx 4.05$ for $n = 512$ [FHK+20]. The heuristic can select a small $\alpha$ that shrinks the estimate; because every candidate is strictly validated, such a choice incurs only the overhead of evaluating one additional candidate family and cannot yield an invalid key.

**Public validation.** For each scale, coefficient-wise rounding produces an integer candidate $f'$. Given $h$ and $f'$, NTRUValid$(h, f')$ applies the checks of Falcon key generation [FHK+20, PP19]: (1) $f'$ is invertible in $\mathcal{R}_q$, and $g' = \mathrm{center}_q(hf')$, so that $g'(f')^{-1} \equiv h \pmod q$ by construction; (2) $\max_i\{|f'_i|, |g'_i|\} \le 31$; (3) $\gamma(f', g') \le 1.17\sqrt{q}$, where

$$\gamma(f', g') = \max\Big(\|(f', g')\|_2\,,\ \Big\|\Big(\tfrac{q\ \mathrm{adj}(f')}{S},\ \tfrac{q\ \mathrm{adj}(g')}{S}\Big)\Big\|_2\Big), \qquad S = f'\,\mathrm{adj}(f') + g'\,\mathrm{adj}(g'),$$

$\mathrm{adj}(a) = a(x^{-1}) \bmod (x^n + 1)$, and the division is taken in $\mathcal{K}$ and evaluated in the FFT domain; and (4) only if (1)–(3) pass, the NTRU completion procedure [PP19] returns $(F', G')$ within the Falcon private-key encoding bounds with $f'G' - g'F' = q$. On success, the routine returns $(g', F', G')$; otherwise, it returns $\perp$. The validation procedure relies strictly on the public key $h$ and candidate $f'$. To optimize runtime, our implementation evaluates this conjunction in order of increasing computational cost (first testing the coefficient bounds on $f'$ and $g'$, then verifying $\|(f', g')\|_2 \le 1.17\sqrt{q}$, and finally performing the remaining checks only if prior bounds hold), which preserves the exact set of accepted candidates. The NTRU completion of key generation is the only lattice computation in the reconstruction; no LLL or BKZ reduction is performed.

**Integer correction (IC).** For a fixed scale $\alpha$, let $y = \alpha\tilde{f}$, $f^{(0)} = \mathrm{round}(y)$, and $\varrho_i = |y_i - f_i^{(0)}|$, and let $\mathcal{S}_K(y)$ contain the indices of the $K$ largest residuals $\varrho_i$. IC searches

$$\mathcal{F}_{\mathrm{IC}}(y; K, \omega) = \Big\{ f^{(0)} + \delta : \delta \in \{-1, 0, 1\}^n,\ \mathrm{supp}(\delta) \subseteq \mathcal{S}_K(y),\ \|\delta\|_0 \le \omega \Big\}, \tag{11}$$

which contains at most $\sum_{j=0}^{\omega} \binom{K}{j} 2^j$ candidates, that is, 34,281 for $K = 30$ and $\omega = 3$. Both signs are searched so that coordinates whose estimate deviates from the secret by more than $1/2$ can still be corrected. For each candidate, IC derives $g' = \mathrm{center}_q(hf')$ rather than rounding $\tilde{g}$ and applies NTRUValid$(h, f')$. Valid candidates are ranked lexicographically by $\mathrm{score}_y(f') = (\|f' - y\|_2^2\,, \|(f', g')\|_2^2)$. The residual $\varrho_i$ is a heuristic selection criterion, not a calibrated error probability.

**Proposition 2** (Public validity)**.** *Suppose that* NTRUValid$(h, f')$ *returns* $(g', F', G')$*, and let* $B'$ *be the matrix* $\left(\begin{smallmatrix} g' & -f' \\ G' & -F' \end{smallmatrix}\right)$*.*

*(a) The* $2n$ *vectors* $x^i(g', -f')$ *and* $x^i(G', -F')$*,* $0 \le i < n$*, form a basis of the lattice* $\Lambda_h = \{(u, v) \in \mathcal{R}^2 : u + vh \equiv 0 \pmod q\}$*, identified with a sublattice of* $\mathbb{Z}^{2n}$*.*

*(b)* $(f', g', F', G')$ *satisfies the acceptance conditions of Falcon key generation and is a private key for* $h$*.*

*(c) Every signature that the signing procedure of [FHK+20] outputs with* $B'$ *is accepted by the verifier under* $h$*.*

*Proof.* (a) *Membership.* Check (1) gives $g' \equiv hf'$, hence $g' - f'h \equiv 0 \pmod q$. Check (4) gives $f'G' - g'F' = q$; multiplying by $(f')^{-1}$ modulo $q$ and using $g'(f')^{-1} \equiv h$ yields $G' - F'h \equiv q(f')^{-1} \equiv 0 \pmod q$. Since $\Lambda_h$ is closed under multiplication by $x$, all $2n$ vectors lie in $\Lambda_h$.

*Determinant.* Let $\mathbf{W} \in \mathbb{Z}^{2n \times 2n}$ have the rows $(\mathrm{vec}(x^i g'), \mathrm{vec}(-x^i f'))$ for $0 \le i < n$, followed by the rows $(\mathrm{vec}(x^i G'), \mathrm{vec}(-x^i F'))$. Since $N(a)\mathbf{e}_i = \mathrm{vec}(ax^i)$, $\mathbf{W}$ has the blocks $W_{11} = N(g')^{\mathsf{T}}$, $W_{12} = -N(f')^{\mathsf{T}}$, $W_{21} = N(G')^{\mathsf{T}}$, and $W_{22} = -N(F')^{\mathsf{T}}$. Negacyclic matrices commute (Section 2.2), and so do their transposes; in particular, $W_{21}W_{22} = W_{22}W_{21}$. For a block matrix whose lower blocks commute, $\det \mathbf{W} = \det(W_{11}W_{22} - W_{12}W_{21})$ [Sil00, Theorem 3]. Using $N(a)N(p) = N(ap)$, $\det \mathbf{W} = \det\big((N(G')N(f') - N(F')N(g'))^{\mathsf{T}}\big) = \det N(f'G' - g'F') = \det(qI_n) = q^n$.

**Algorithm 3** Magnitude-aware Falcon key recovery with an ordered scale cascade

**Require:** Public key $h$, increasing signature counts $\mathcal{P}$, public row data, fused sets $D_{R,j,\ell}$, IC budget $(K, \omega)$
**Ensure:** Valid private key $(f, g, F, G)$, or failure

**for all** $p \in \mathcal{P}$ in increasing order **do**
  Retain the observations of the first $p$ signatures and map each output $\ell$ to row $\pi_n(\ell)$
  Convert each $D_{R,j,\ell}$ to $(L_{j,\ell}, U_{j,\ell}, a^L_{j,\ell}, a^U_{j,\ell})$ and form (8) and (9)
  Minimize (10) over $\Theta$ to obtain $(\tilde{f}_p, \tilde{g}_p)$
  **for** $\alpha$ in the ordered list $[1, \alpha_{\mathrm{heur}}(\tilde{f}_p)]$ **do**
    $\mathcal{V}_\alpha \leftarrow \{(f', g', F', G') : f' \in \mathcal{F}_{\mathrm{IC}}(\alpha\tilde{f}_p; K, \omega),\ \mathrm{NTRUValid}(h, f') = (g', F', G')\}$
    **if** $\mathcal{V}_\alpha \neq \emptyset$ **then**
      **return** $\arg\min_{(f',g',F',G') \in \mathcal{V}_\alpha} \mathrm{score}_{\alpha\tilde{f}_p}(f')$
    **end if**
  **end for**
**end for**
**return** failure

*Index.* Let $N(h)$ denote the matrix of multiplication by an integer representative of $h$. The map $\mathbb{Z}^{2n} \to (\mathbb{Z}/q\mathbb{Z})^n$, $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{u} + N(h)\mathbf{v} \bmod q$, is a group homomorphism, surjective already on $\mathbf{v} = 0$, with kernel $\Lambda_h$; hence $[\mathbb{Z}^{2n} : \Lambda_h] = q^n$. The rows of $\mathbf{W}$ are linearly independent because $\det \mathbf{W} \neq 0$, and they generate a sublattice of $\Lambda_h$ whose index in $\mathbb{Z}^{2n}$ is $|\det \mathbf{W}| = q^n$. As the two indices coincide, this sublattice equals $\Lambda_h$.

(b) By [FHK+20, PP19], checks (1)–(4) are the acceptance conditions of Falcon key generation, and $g'(f')^{-1} \equiv h$, so the public key of $(f', g', F', G')$ is $h$ and the signing procedure of [FHK+20] is defined for this key.

(c) Let $(s'_1, s'_2)$ be the vector produced with $B'$ by the signing procedure. The congruence $s_1 + s_2 h \equiv c \pmod q$ in the proof of Lemma 2(ii) uses only (1), the invertibility of $f$ modulo $q$, and $h \equiv gf^{-1}$; these properties hold for $(f', g', F', G')$, so $s'_1 \equiv c - s'_2 h \pmod q$. The signer outputs a signature only if $\|(s'_1, s'_2)\|_2^2 \le \lfloor \beta^2 \rfloor$ and $s'_2$ can be encoded. The verifier decodes $s'_2$ and computes $\tilde{s}_1 = \mathrm{center}_q(c - s'_2 h)$. Because $\mathrm{center}_q$ returns a representative of minimum absolute value in each residue class, $|\tilde{s}_{1,i}| \le |s'_{1,i}|$ for every $i$, so $\|(\tilde{s}_1, s'_2)\|_2^2 \le \lfloor \beta^2 \rfloor$, and the verifier accepts. This argument does not require $\lfloor \beta^2 \rfloor < q^2/4$. □

Let $T_{\mathrm{valid}}$ denote the first prefix at which Algorithm 3 accepts a candidate, and $T_{\mathrm{exact}}$ the first prefix at which the candidate selected by the same per-prefix rule is equivalent to the recorded secret, that is, $f' = \pm x^i f$ for some $0 \le i < n$, when every prefix is evaluated without early termination. The recorded secret is used only to report $T_{\mathrm{exact}}$ and affects neither ranking nor acceptance. Because Algorithm 3 evaluates both candidate scales at a given prefix before incrementing the signature budget, and the candidate selected at prefix $T_{\mathrm{exact}}$ belongs to the set $\mathcal{V}_\alpha$ of validated candidates, we have $T_{\mathrm{valid}} \le T_{\mathrm{exact}}$.

## 5 Evaluation on PQClean

PQClean is our primary target because its staged normalization supports 17-class $e(k)$ classification. All measurements use PQClean Falcon-512 compiled with `-O3` and GPIO-delimited SamplerZ calls, and classifiers trained on five profiling keys are applied without refitting to 50 disjoint evaluation keys.

## 5.1 Measurement Setup and Alignment

The target is a ChipWhisperer CW308 board with an STM32F415 Cortex-M4 microcontroller, clocked at 10 MHz and controlled by a ChipWhisperer-Lite. Power is acquired with a PicoScope 3415E at a 9.6 ns sampling interval with 10-bit resolution, an AC-coupled 50 mV range, and a 20 MHz bandwidth limit. Both implementations are compiled with `arm-none-eabi-gcc` 10.3.1 at `-O3`; the c-fn-dsa build enables its Cortex-M4 assembly path.

**Call boundaries and diagnostic localization.** GPIO pulses placed immediately before and after each call to SamplerZ delimit the invocations (Figure 1(a)). From each retained execution, we use the first 512 of the 1,024 calls, which form $Z_1$ by Lemma 3; all sample sizes reported below refer to retained executions. Before profiling, we acquired separate diagnostic traces with additional GPIO pulses bracketing `fpr_of(k)` and `fpr_of(z0 * z0)`. The $z_0^2$ pulse occurs once per rejection iteration, because a failed BerExp test repeats the loop and a successful test ends SamplerZ; its last occurrence within a call therefore identifies the accepted iteration, and its rising edge defines the accepted-iteration anchor (Figure 1(b)). These internal pulses are absent from all profiling and recovery measurements.

**Alignment without internal markers.** From the regions identified in the diagnostic traces, we construct class-independent templates for the accepted-iteration and output regions from a single signature, without labels, and freeze them. Starting from the falling edge of the call-boundary pulse, which remains present, the locator searches within $\pm 512$ samples around the expected output anchor 1,948 samples earlier and then within $\pm 512$ samples around the expected accepted-iteration anchor 18,075 samples before the located output anchor. Each search averages non-overlapping groups of eight samples and maximizes the normalized correlation with its frozen template, and the two anchors must appear in the expected order within the call boundaries. The same locator is applied to profiling and evaluation traces without label-dependent adjustment, and executions for which a search fails or the anchor order is violated are discarded. The profiling and evaluation datasets each contain 2,500 retained signatures, corresponding to 1,280,000 SamplerZ observations per dataset. Relative to the ground-truth anchors provided by internal diagnostic pulses, this marker-free locator degraded $e(k)$ accuracy by only about one percentage point while leaving $b$ and $z_0$ classification accuracies virtually unaffected; all reported experimental results are based on this marker-free alignment.

## 5.2 Leakage Classification

**Populations and data split.** The measured populations (keys $\times$ signatures per key) are $5 \times 500$ for profiling and $50 \times 50$ for recovery; for c-fn-dsa, they are $5 \times 500$ and $50 \times 200$. The five profiling keys were selected to provide broad coverage of the $e(k)$ classes, and no evaluation key is a profiling key. Each profiling key's 500 signatures are split into 300 training, 100 calibration, and 100 internal-test signatures, with all 512 calls of a signature in the same partition, giving 1,500/500/500 signatures in total. Points of interest (POIs) are selected by training-only SNR [MOP07]. The calibration partition is used only for model and threshold selection, and all test sets are evaluated after these choices are frozen.

**Classifiers.** Each retained trace contributes two blocks: an 8,192-sample block starting at the rising edge of the call-boundary pulse and a 20,480-sample block in which the accepted-iteration anchor lies 12,288 samples after the start. The $e(k)$ classifier uses samples $[0, 2048)$ of the first block, divides the signed ADC values by $2^{15}$, averages non-overlapping groups of eight samples, and applies a one-dimensional CNN with 16 filters

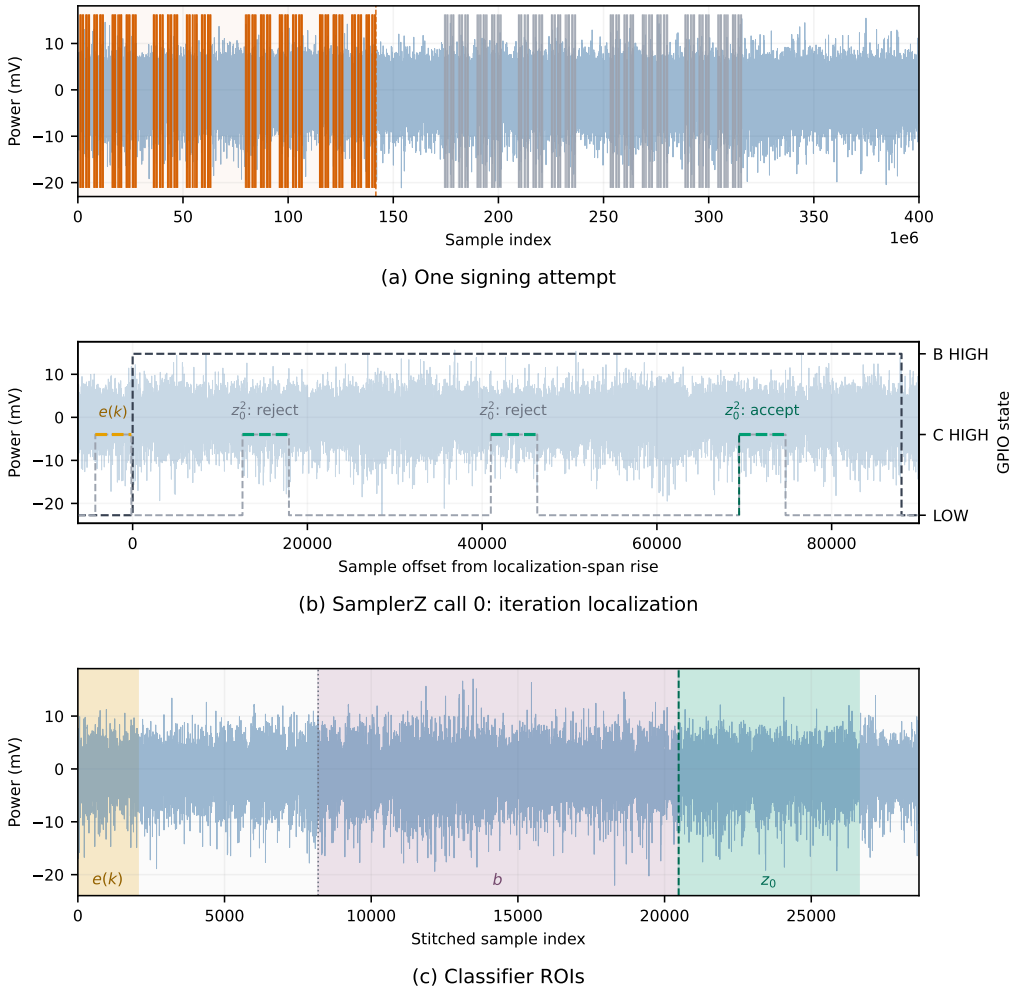



**Figure 1:** PQClean measurement (9.6 ns/sample). (a) One signing attempt with 1,024 GPIO-delimited SamplerZ calls; orange marks the first 512 calls retained for analysis, and gray marks the remaining 512. (b) A separate diagnostic capture of the first SamplerZ call (index 0), in which the last $z_0^2$ pulse marks the accepted iteration. (c) Observation windows for $e(k)$, $b$, and $z_0$, located without internal markers; the two stored blocks are stitched but are not contiguous in time.

of size 9, ReLU activation, flattening, and a 17-class linear output layer. Training uses inverse-frequency-weighted cross entropy, AdamW (learning rate $10^{-3}$, weight decay $10^{-4}$), mini-batches of two signatures, and ten epochs; the checkpoint is selected by calibration accuracy, with balanced accuracy breaking ties. Because the $b$-dependent computation of the signed proposal precedes the $z_0^2$ conversion, the $b$ classifier uses 64 raw-sample POIs in $[-12288, 0)$ relative to the accepted-iteration anchor, whereas the $z_0$ classifier uses 64 mean-bin POIs in $[0, 6144)$. Both employ shrinkage LDA with uniform class priors, which corresponds to a Gaussian classifier with pooled covariance analogous to standard template attacks [CRR02]. On the calibration partition, the CNN outperforms LDA for $e(k)$, whereas LDA already exceeds 99.99% for $b$ and $z_0$.

**Table 2:** PQClean call-level classification on the internal-test partition of the five profiling keys (256,000 calls). The last row is the fraction of calls whose fused set contains the true output.

| Quantity | Model | Rate (%) |
|---|---|---|
| $e(k)$ | LDA | 98.4543 |
| $e(k)$ | 1-layer CNN (adopted) | 99.3449 |
| $b$ | LDA | 99.9961 |
| $z_0$ | LDA | 99.9922 |
| Fused set | True $R$ contained | 99.3480 |

**Results.** Table 2 reports the call-level results. Among the 1,280,000 accepted calls of the evaluation keys, none has $z_0 \geq 8$, consistent with Lemma 1; this does not exclude such values, which the four-class model would misclassify. The $e(k)$ accuracy of 99.34% achieved at `-O3` compares favorably with the single-trace accuracy of 0.884 reported by Zhou et al. for PQClean at `-O3` [ZWSY26]. Because the acquisition chains, alignment procedures, and classifier architectures differ, this comparison highlights the achievable resolution rather than a strictly controlled head-to-head benchmark.

## 5.3 Synthetic Error Sensitivity and Integer Correction

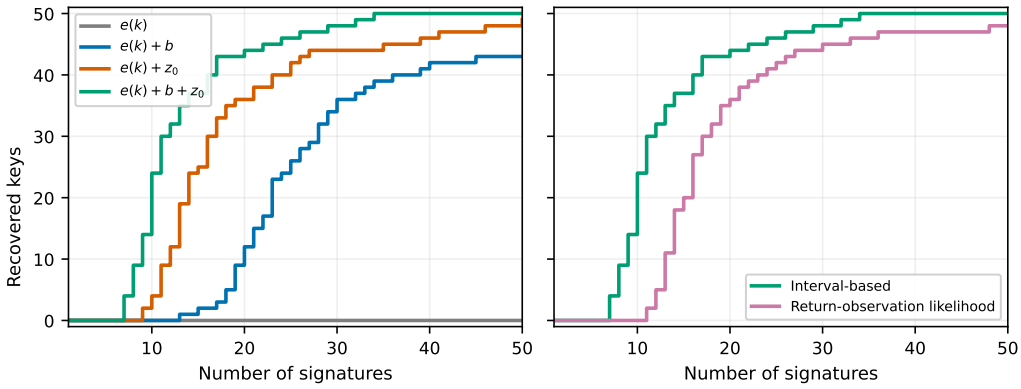
**Recovery settings.** In (10), we set $\lambda = 10^{-5}$ and $w_{j,\ell} = 1$ for every retained row, so classifier confidence and interval width do not reweight the loss. L-BFGS-B uses the analytic gradient, the box $\Theta$, at most 2,000 iterations, `ftol` $= 10^{-12}$, and `gtol` $= 10^{-7}$. The first prefix starts at the zero vector, and later prefixes start from the preceding estimate. Algorithm 3 evaluates the unit and integer-distance scales sequentially with $K = 30$ and $\omega = 3$.

**Error sensitivity.** To isolate the performance of integer correction from classifier imperfections, we simulate 50 Falcon-512 keys with exact 17-class fused intervals for the first 512 outputs of each signing attempt and, at a given corruption rate, replace the endpoints of each selected observation by those of a uniformly sampled admissible interval or ray excluding the true output; these corruption rates model perturbation directly at the level of fused constraints rather than raw classification errors, serving to benchmark algorithmic robustness under controlled noise. With 0%, 10%, and 20% corruption, IC reduces the number of signatures needed to recover all 50 keys from 24 to 22, from 425 to 295, and from more than 1,000 to 950, respectively. At 20% corruption, IC increases the number of recovered keys from 24 to 34 (out of 50) at 100 signatures and from 43 to 48 at 200. Every validated candidate in this simulation is exact.

## 5.4 Measured Key Recovery, Observation Subsets, and Decoders

**Measured key recovery.** We apply the three-observation procedure with frozen classifiers to the 50 evaluation keys with 50 signatures each. Signature prefixes are processed in increasing order, and each key stops at the first candidate accepted by NTRUValid; the recorded secret is consulted only afterward. All 50 keys are recovered, and for every key the first accepted candidate is equivalent to the recorded secret, so $T_{\text{valid}} = T_{\text{exact}}$. The number of signatures at first success has mean 13.12, median 11, and maximum 34; 41 keys first succeed at the unit scale and 9 at the integer-distance scale. Treating the keys as independent draws, the two-sided 95% Clopper–Pearson interval [CP34] for the success rate within 50 signatures is $[92.9\%, 100\%]$, indicating that successfully recovering 50 out of 50 evaluated keys does not guarantee universal success across arbitrary keys.

**Observation subsets.** Figure 2(a) compares $\{e(k)\}$, $\{e(k), b\}$, $\{e(k), z_0\}$, and $\{e(k), b, z_0\}$ under the interval decoder with identical keys, signature order, optimizer settings, IC budget, and validation. Missing channels are marginalized over their modeled support, $D_b = \{0, 1\}$ or $D_{z_0} = \{0, \ldots, 7\}$; without $b$, $D_R$ may contain gaps (e.g., $D_k = \{0\}$ and $D_{z_0} = \{2, 3\}$ give $\{-3, -2, 3, 4\}$), so $\{e(k), z_0\}$ uses the interval envelope, a relaxation of its information. The four configurations recover 0, 12, 36, and 44 keys at 20 signatures and 0, 43, 49, and 50 keys at 50 signatures. Among the 49 keys recovered by both $\{e(k), z_0\}$ and $\{e(k), b, z_0\}$, adding $b$ reduces the median number of required signatures by 4; among the 43 keys recovered by both $\{e(k), b\}$ and $\{e(k), b, z_0\}$, adding $z_0$ reduces it by 13. Since $\{e(k), z_0\}$ already recovers 49 keys within 50 signatures, incorporating the third observation primarily accelerates convergence by reducing the required signature count, rather than being strictly indispensable for recovery.




(a) Interval-based leakage ablation (b) Decoder comparison

**Figure 2:** Cumulative recovery of the 50 PQClean evaluation keys with identical keys and signature order. (a) Interval decoder with the four observation subsets. (b) Interval decoder and return-likelihood decoder, both with all three observations.

**Return-observation likelihood.** A second decoder replaces interval regression by a profiling-calibrated likelihood over the same $2n$ variables. For retained observation $i = 1, \ldots, M$, let $Y_i = (\hat{c}_{k,i}, \hat{c}_{b,i}, \hat{c}_{z_0,i})$. From the calibration partition, we estimate $Q(y \mid m) = \Pr(Y_i = y \mid R_i = m)$ for integers $m$ with a pseudocount of 0.5, and $\widetilde{Q}$ interpolates linearly between adjacent integers. A candidate $(f, g)$ predicts $R_i(f, g)$ through (8), and the decoder minimizes $-\frac{1}{M}\sum_{i=1}^{M} \log \widetilde{Q}(Y_i \mid R_i(f, g)) + \frac{\lambda}{2}\|(f, g)\|_2^2$. Both decoders share the retained observations, coefficient bounds, IC budget, and validation. The likelihood decoder is initialized for each prefix from a least-squares estimate based on profiling-calibrated conditional mean returns, without warm starts or restarts, executing the L-BFGS-B solver (via SciPy) for at most 2,000 iterations. Outside the calibrated return range $[-207, 186]$, $\widetilde{Q}$ is set to $1/136$, where $136 = 17 \times 2 \times 4$ is the number of observation-label triples, so the corresponding data-loss term is constant with zero gradient. This is an empirical observation model, not the exact generative likelihood of SamplerZ. With all three observations, the interval decoder recovers 24, 44, and 50 keys at 10, 20, and 34 signatures, whereas the likelihood decoder recovers 0, 36, and 46 keys at the same counts and reaches 48 keys at 48 signatures, with no additional keys recovered up to 50 signatures (Figure 2(b)). Because the decoders also differ in initialization and tail handling, this comparison reflects these specific solver implementations rather than the two underlying paradigms in general.

**Relation to center-only attacks.** Under the interval decoder, $\{e(k)\}$ recovers no key within 50 signatures despite a 99.34% classifier accuracy. This outcome is attributable to the conservative relaxation in the interval decoder, which marginalizes the missing $b$ and $z_0$ over the modeled proposal set $\{-7, \ldots, 8\}$, rather than an inherent insufficiency of center information: Zhou et al. recover keys from center leakage alone at lower classification accuracy, using a confusion-matrix maximum-likelihood estimator followed by a lattice-reduction search [ZWSY26].

### 5.5 Computational Cost

We benchmark post-classification recovery with Algorithm 3 on an Intel Core i9-14900K with 24 physical cores and 32 logical CPUs in the process affinity, with 32 threads. The CNN runs on an NVIDIA RTX 4060 with CUDA 12.8; L-BFGS-B, IC, and validation run on the CPU. Excluding trace acquisition, profiling and classification of all traces take approximately 38 minutes in total, representing a one-time offline profiling cost rather than a per-target attack time.

Post-classification recovery of the 50 keys takes 1,350.74 s in total (sum of per-key wall times); per-key wall time has median 17.54 s, 75th percentile 30.51 s, 90th percentile 54.07 s, and maximum 109.72 s. L-BFGS-B and IC prior to the NTRU validator account for 68.57% and 27.98% of the total runtime, respectively, while the validator accounts for 0.012%. Because the algorithm terminates upon discovering the first valid candidate, the IC search enumerates $43{,}571{,}151 = 34{,}281 \times 1{,}271$ candidates across the 1,271 evaluated pairs of prefix and scale; preliminary coefficient-range checks on $f'$ and on the derived $g'$ filter out all but 50 candidates (a retention rate of 0.000115%), each of which subsequently satisfies all remaining validation criteria.

## 6 Evaluation on CLZ-Based c-fn-dsa

This section examines whether sampler leakage enables key recovery when the staged normalization is replaced by the CLZ-based conversion of Section 3.2, using disjoint profiling and evaluation keys and the call boundaries of Section 5.1.

### 6.1 Target and Reduced-Resolution Classification

**Target.** The evaluated version of c-fn-dsa (commit `1b858880dcac`) is a pre-standard FN-DSA-512 implementation with Cortex-M4 assembly routines that shares the ffSampling and SamplerZ structure of Falcon-512 (Section 4.2). Its Cortex-M4 build computes $(s_1, s_2)$ modulo $q$, so the constraints rely on the relation condition $w_1 f + w_2 g = 0$ of Section 4.1. We directly checked the relation for all 512 coefficients of each of the 10,000 signatures in the c-fn-dsa evaluation dataset; the condition held in all 5,120,000 coefficient checks.

**Classifier comparison.** On the internal-test partition, a flat 17-class LDA reaches 46.59% call-level accuracy and the hierarchical 7-class LDA described next 82.61%; the two differ in resolution and architecture, but both show that center information remains observable.

**Hierarchical classifier.** Each retained observation comprises a 4,096-sample block starting at the rising edge of the call-boundary pulse and a 20,480-sample block spanning 12,288 samples before and 8,192 samples after the accepted-iteration anchor. The anchor is located using a frozen class-independent template spanning $[-4096, -512)$ samples relative to the anchor. For call-boundary edges $t_{\text{rise}}$ and $t_{\text{fall}}$, the search interval is $[\max(t_{\text{rise}} + 4096,\ t_{\text{fall}} - 14000),\ t_{\text{fall}} - 1536]$, with all positions expressed in samples. The classifier first predicts $\text{sgn}(k) \in \{-1, 0, +1\}$ from 96 POIs of eight-sample means. Zero maps to

$e(k) = 0$, and a sign-specific 3-class LDA assigns one of the three magnitude groups of Section 3.2, using 448 POIs of four-sample means with four-sample spacing. Separate windows are used to classify $b$ and $z_0$.

**Prior correction and low-margin fallback.** Within each nonzero branch, let $\phi_\kappa(\mathbf{x})$ be the LDA score of class $\kappa$ and $\hat{p}_\kappa$ its conditional frequency in the training partition. The corrected score is

$$\phi_\kappa^{\text{prior}}(\mathbf{x}) = \phi_\kappa(\mathbf{x}) + \eta_{\text{prior}} \log\left(\frac{\hat{p}_\kappa}{1/3}\right), \qquad \eta_{\text{prior}} = 0.2.$$

Let $\Delta$ be the gap between the two largest corrected scores. For a fallback rate of 40%, the threshold $\tau = 1.876688$ is the 40th percentile of $\Delta$ over nonzero predictions in the profiling data. Predictions with $\Delta \le \tau$ are reassigned to the outer class of their predicted sign; otherwise, the class with the largest score is retained.

**Configuration selection.** The weight $\eta_{\text{prior}}$ was tuned in preliminary experiments to balance overall accuracy against class-balanced accuracy, and $\tau$ was derived from the profiling partition. The fallback rate and outer-range decoding strategy were explored across the four combinations summarized in Table 3 on the evaluation keys. Consequently, the c-fn-dsa recovery results represent an exploratory evaluation rather than an independent validation of a pre-fixed policy.

**Classification results.** Relative to the base classifier (uniform class priors, no fallback, conservative outer ranges), the selected configuration raises, on the internal-test partition, the $e(k)$ accuracy from 82.61% to 89.55%, the rate of fully correct calls from 81.79% to 88.66%, and the fused containment rate from 84.97% to 89.71%; the $b$ and $z_0$ classifiers (99.65% and 99.17%; every $z_0$ class has recall of at least 96.65%) are unchanged. The baseline classifier operates without prior correction and serves as an uncalibrated reference outside the four tuning configurations in Table 3.

## 6.2 Measured Key Recovery

**Selected configuration.** We apply Algorithm 3 to postprocessed 7-class $e(k)$, $b$, and $z_0$ observations with $K = 30$ and $\omega = 3$. The selected configuration recovers 38/50 keys within 200 signatures, with $T_{\text{valid}} = T_{\text{exact}}$ for every recovered key. The number of signatures required for initial key recovery ranges from 38 to 159 (median 80; mean 82.08).

**Fallback and outer-range decoding.** Table 3 compares four combinations of fallback (none or 40%) and outer-range decoding (conservative or tight), all with prior correction and with other settings fixed. Both fallback and tight decoding improve recovery, and their combination performs best on these keys. Since validation relies strictly on the public key and no combination yields a false positive, evaluating all four configurations sequentially and accepting the first validated candidate guarantees the recovery of at least these 38 keys without a prior policy selection, albeit with additional computational overhead; the configuration parameters therefore warrant further systematic validation.

**Computational cost.** For the 38 recovered keys, post-classification recovery takes a median of 592.94 s per key (90th percentile 1,305.81 s; maximum 1,969.45 s), of which L-BFGS-B accounts for 91.60% in aggregate; these statistics are conditioned on success and exclude the 12 keys unresolved at 200 signatures.

**Table 3:** c-fn-dsa keys recovered within 200 signatures for each combination of low-margin fallback and outer-range decoding, with prior correction ($\eta_{\text{prior}} = 0.2$) in all four cases. The highlighted configuration yielded the highest recovery rate among the four evaluated candidates. No configuration yields a false positive.

| Fallback | Conservative ($k \leq -1$, $k \geq 1$) | Tight ($k \leq -8$, $k \geq 8$) |
|---|---|---|
| None | 1/50 | 13/50 |
| 40% low-margin fallback | 21/50 | **38/50** |

# 7 Discussion

## 7.1 Threat Model, Limitations, and Scope

Our measurements assume known SamplerZ call boundaries, provided by GPIO pulses (Section 2.3); segmenting a signing trace into its 1,024 calls without external markers requires a separate localization step, for which automated pattern-locating techniques [TBW+22] may be applicable. We focus our evaluation on single-device profiling and attack setups under identical hardware conditions; investigating profile portability across distinct devices and trace attrition rates under varied environmental noise constitutes valuable future work. For c-fn-dsa, deterministic replay verified the relation condition (Section 4.1) for all 5,120,000 coefficients in the 10,000-signature evaluation set, and our empirical exploration demonstrates the viability of exploiting reduced-resolution CLZ leakage. Finally, our analysis and experiments are restricted to $n = 512$ (Remark 1).

Secret-key floating-point attacks [KLL+26, LMDG25] are complementary: sampler leakage occurs at every signature regardless of key-expansion protection, so both components require separate, dedicated protection.

## 7.2 Countermeasures

Combining $e(k)$ with either $b$ or $z_0$ already enables recovery in Figure 2(a), and Lin et al. recovered keys from $z_0$ and $b$ without center observations [LZY+25]. Protecting the center conversion alone therefore does not address sampler leakage as a whole, and in the evaluated CLZ-based implementation our classifiers still extracted seven center classes that, combined with $b$ and $z_0$, enabled recovery; countermeasures should cover $e(k)$, $b$, and $z_0$ jointly. Isochronous samplers [HPRR20] remove timing variation but not necessarily value-dependent power consumption, and masking of floating-point arithmetic [CC24] is effective only if it covers all relevant operations [CCCY26], including the conversions and the base sampler within SamplerZ. Twisting [LTY26] is expected to hide the public correspondence of Lemma 3 between execution order and the coefficients of $Z_1$ on which our constraints rely; whether magnitude information remains exploitable under twisting remains an open question. A comprehensive side-channel evaluation of these countermeasures falls outside the scope of this work.

# 8 Conclusion

Three observations per SamplerZ call can be fused into integer intervals that contain the outputs of Falcon's fast Fourier sampler whenever the observations are correct, and a public linear relation maps these intervals to slab and halfspace constraints on the secret key. Interval regression with public validation recovers all 50 PQClean keys and, in an exploratory evaluation, 38 of 50 c-fn-dsa keys despite its CLZ-based conversion, so sampler protection should cover all three observation channels.

## References

[BKM26] Marcus Brinkmann, Nicolai Kraus, and Alexander May. Halfspace learning for lattice signature key recovery from signs. In *Advances in Cryptology – CRYPTO 2026*, Lecture Notes in Computer Science, pages 359–390. Springer, 2026. Full version: Cryptology ePrint Archive, Paper 2026/1366, https://eprint.iacr.org/2026/1366.

[CC24] Keng-Yu Chen and Jiun-Peng Chen. Masking floating-point number multiplication and addition of Falcon: First- and higher-order implementations and evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(2):276–303, 2024.

[CCCY26] Keng-Yu Chen, Ming Qing Ching, Jiun-Peng Chen, and Bo-Yin Yang. When masking multiplication isn't enough: Exploiting floating-point leakage in Falcon's pre-image computation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2026(2):686–708, 2026.

[CP34] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*, Lecture Notes in Computer Science, pages 13–28. Springer, 2002.

[DP16] Léo Ducas and Thomas Prest. Fast Fourier orthogonalization. In *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 191–198, 2016.

[FHK+20] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU, specification v1.2. Submission to the NIST Post-Quantum Cryptography Standardization Process, 2020. https://falcon-sign.info.

[GMRR22] Morgane Guerreau, Ange Martinelli, Thomas Ricosset, and Mélissa Rossi. The hidden parallelepiped is back again: Power analysis attacks on Falcon. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(3):141–164, 2022.

[HPRR20] James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Isochronous Gaussian sampling: From inception to implementation, with applications to the Falcon signature scheme. In *International Conference on Post-Quantum Cryptography (PQCrypto 2020)*, Lecture Notes in Computer Science, pages 53–71. Springer, 2020.

[KA21] Emre Karabulut and Aydin Aysu. Falcon down: Breaking the FALCON post-quantum signature scheme through side-channel attacks. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 691–696. IEEE, 2021.

[KLL+26] GyuSang Kim, JeongHwan Lee, Myeonghoon Lee, Seokhie Hong, and HeeSeok Kim. Secret key recovery of FALCON using simple power analysis in conditional calculator. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2026(2):953–975, 2026.

[LMDG25] Kang Li, Shouran Ma, Haochen Dou, and Qian Guo. One fell swoop: A single-trace key-recovery attack on the Falcon signing algorithm. Cryptology ePrint Archive, Paper 2025/2159, 2025. https://eprint.iacr.org/2025/2159.

[LTY26] Xiuhan Lin, Mehdi Tibouchi, and Yang Yu. Swing the lure: How to cheaply mitigate sign leakage in Falcon. Cryptology ePrint Archive, Paper 2026/2170, 2026. https://eprint.iacr.org/2026/2170.

[LZY+25] Xiuhan Lin, Shiduo Zhang, Yang Yu, Weijia Wang, Qidi You, Ximing Xu, and Xiaoyun Wang. Thorough power analysis on Falcon Gaussian samplers and practical countermeasure. In *IACR International Conference on Public-Key Cryptography (PKC 2025)*, Lecture Notes in Computer Science, pages 229–258. Springer, 2025.

[MHS+19] Sarah McCarthy, James Howe, Neil Smyth, Séamus Brannigan, and Máire O'Neill. BEARZ attack FALCON: Implementation attacks with countermeasures on the FALCON signature scheme. In *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (SECRYPT)*, pages 61–71, 2019.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.

[NCHS24] Jaesang Noh, Hyunseo Choi, Dongwoo Han, and Dong-Joon Shin. Efficient error-tolerant side-channel attacks on GPV signatures based on ordinary least squares regression. Cryptology ePrint Archive, Paper 2024/2043, 2024. https://eprint.iacr.org/2024/2043.

[PP19] Thomas Pornin and Thomas Prest. More efficient algorithms for the NTRU key generation using the field norm. In *IACR International Workshop on Public Key Cryptography (PKC 2019)*, Lecture Notes in Computer Science, pages 504–533. Springer, 2019.

[QA26] Jinyi Qiu and Aydin Aysu. SHIFT SNARE: Uncovering secret keys in FALCON via single-trace analysis. *Journal of Cryptographic Engineering*, 16(3):14, 2026. Preprint: Cryptology ePrint Archive, Paper 2025/146.

[RN75] Richard N. Rosett and Forrest D. Nelson. Estimation of the two-limit probit regression model. *Econometrica*, 43(1):141–146, 1975.

[Sil00] John R. Silvester. Determinants of block matrices. *The Mathematical Gazette*, 84(501):460–467, 2000.

[TBW+22] Jens Trautmann, Arthur Beckers, Lennert Wouters, Benedikt Gierlichs, Stefan Wildermann, Ingrid Verbauwhede, and Jürgen Teich. Semi-automatic locating of cryptographic operations in side-channel traces. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):345–366, 2022.

[ZLYW23] Shiduo Zhang, Xiuhan Lin, Yang Yu, and Weijia Wang. Improved power analysis attacks on Falcon. In *Advances in Cryptology – EUROCRYPT 2023*, Lecture Notes in Computer Science, pages 565–595. Springer, 2023.

[ZWSY26] Yuanyuan Zhou, Weijia Wang, Yiteng Sun, and Yu Yu. Every signing leaks: Breaking Falcon via floating-point conversion leakage. Cryptology ePrint Archive, Paper 2026/2124, 2026. https://eprint.iacr.org/2026/2124.