

# Rejection Matters: Efficient Non-Profiling Side-Channel Attack on ML-DSA via Exploiting Public Templates

Yuhan Zhao\*, Wei Cheng\*<sup>✉</sup>, Zehua Qiao<sup>‡</sup>, Yuejun Liu\*, Yongbin Zhou\*<sup>‡✉</sup>,

\*School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

<sup>†</sup>LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France

<sup>‡</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

{yhzhao96, wei.cheng, liuyuejun, zhouyongbin}@njjust.edu.cn, qiaozehua@iie.ac.cn

**Abstract**—ML-DSA (formerly CRYSTALS-Dilithium), NIST’s primary post-quantum signature standard, is increasingly deployed along with the post-quantum transitions. Yet when the implementations of ML-DSA are deployed in practice, their physical security remains underexplored. In this work, we reveal a new attack surface against ML-DSA by exploiting the leakages from both rejected signing trials and the final accepted signing trial. We present, to the best of our knowledge, the first side-channel attack that *simultaneously* leverages leakage from both trials without relying on clone devices. Unlike traditional Secret-based Template Attacks, which require profiling the leakage of the sensitive intermediates on a clone device, our PTA (Public-based Template Attack) builds leakage templates solely from publicly available data on the target device itself. With challenge  $c$  known, we then perform CPA on the sensitive intermediates using traces from both rejected and accepted signing trials, quadrupling (on average) exploitable leakage per signing request for ML-DSA-44. The experimental results on power traces from an ARM Cortex-M4 board show that challenges  $c$  are fully recovered with only 96 traces, and then the key recovery succeeds in around 300 traces — a fact of 10x fewer than prior art. We highlight that our attack can be applied across all three ML-DSA variants with different security levels. Moreover, our attack works straightforwardly in the hedged (non-deterministic) mode of ML-DSA, demonstrating that the hedging offers no SCA protection in this scenario.

**Index Terms**—Side-Channel Analysis, CRYSTALS-Dilithium, Deep Learning-based Attack, Correlation Power Analysis.

## I. INTRODUCTION

In 2024, NIST standardized ML-DSA (Module-Lattice-Based Digital Signature Algorithm, formerly CRYSTALS-Dilithium) [1] as its primary recommendation for quantum-resistant digital signatures. Unlike classical schemes whose security relies on factoring or discrete logarithms, ML-DSA derives its security from the hardness of the Module-LWE and Module-SIS problems — conjectured to remain intractable even against adversaries equipped with quantum computers. However, as implementations migrate to embedded and general-purpose platforms, the dominant residual risk is not only the theoretical asymptotic hardness but also physical leakage. Side-channel analysis (SCA) has repeatedly shown that mathematically secure designs can be compromised when implementation-level leakage is exploitable [2]–[4].

This work is supported in part by National Key R&D Program of China (No. 2022YFB3103800) and National Natural Science Foundation of China (No.U2336205, No.62202230, No.62202231, No.62302224, No.62302226).

Side-channel analysis is generally categorized into profiling and non-profiling ones. In particular, *profiling* side-channel attacks, which construct leakage templates from a clone or profiling device under known-secret conditions (referred to as *Secret-based Template Attacks* in this work), can achieve high efficiency against cryptographic implementations [5], [6]. Recent works have demonstrated successful profiling-based attacks against ML-DSA [7]–[11]. However, these approaches often rely on strong assumptions that are hard to hold in real-world evaluation, especially the clone devices for profiling.

In contrast, *non-profiling* side-channel attacks represent the most practical and scalable threat model in practice, which correlate power or electromagnetic leakage with public or efficiently computable intermediate values. One of the representative non-profiling attacks against ML-DSA is correlation power analysis (CPA) [12]–[14] (see Table I for a detailed comparison). In particular, Chen et al. [12] proposed the conservative CPA and an efficient two-stage CPA to recover the secret key of ML-DSA during computing. Then Liu et al. [13] continued to optimize the two-stage CPA mainly on its efficiency. A recent line of work introduce the *Public-based Template Attack* (PTA) [15], [16] has emerged. While PTA employs a modeling phase similar to traditional template attacks, it is fundamentally categorized under the *non-profiling threat model*. In PTA, the leakage templates are built only with publicly available variables (like signature output or plaintext/ciphertext) and their leakages, thereby eliminating the dependency on clone devices. However, those works usually focus only on the polynomial multiplication operation  $\mathbf{u} = c\mathbf{s}_1$  during the last accepted signing trial, where  $c$  is the known challenge in the final signature output, and  $\mathbf{s}_1$  is the secret key to be recovered. As a result, it is still open to exploit leakages from other operations (attack surfaces) to launch effective non-profiling attacks against ML-DSA.

To systemically investigate other possible attack surfaces, we first revisit the structural signing procedure of ML-DSA. A core feature of ML-DSA is rejection sampling [18], [19]: a valid signature is typically preceded by several unsuccessful trials. From an implementation viewpoint, each attempt—rejected or accepted—executes essentially the same computational skeleton, including hashing to a challenge, transforms and pointwise operations in the NTT domain, and so on [1], [20]. This

TABLE I: Comparison with SOTA works on ML-DSA (Dilithium).

Work	Target Leakage	Profiling Status	Valid/Reject	Platform (Channel)	Method	Traces/Signatures
Chen et.al. [12] ICCD 2021	$\hat{c} \circ \hat{s}_1$	Non-Profiling	Valid	STM32F405 (1.25GSa/s Power)	CPA	10,000
Qiao et.al. [17] ePrint 2023	$\hat{c} \circ \hat{s}_1$	Non-Profiling	Valid	STM32F405 (100MSa/s Power)	CPA	5,000
Liu et.al. [13] ETS 2021	$\hat{c} \circ \hat{s}_1$	Non-Profiling	Valid	STM32F405 (125MSa/s Power)	CPA	3,000
Tosun et.al. [14] TIFS 2024	$\hat{c} \circ \hat{s}_1$	Non-Profiling	Valid	MAX32 (10GSa/s EM)	CPA	220 (ML-DSA-65)
Bronchain et.al. [8] CHES 2024	$s \circ c, y$	Secret-based Profiling	Valid or Reject	Simulated	SASCA+FA	4 (Valid) / 1.5M (Reject)
Zhou et.al. [9] CHES 2025	$z, c$	Secret-based Profiling	Reject	STM32F (62.5MSa/s Power)	DL-TA+ILP	$3 \times 10^6$
<b>Ours</b>	$\hat{c} \circ \hat{s}_1$	<i>Public-based Profiling</i>	<b>Valid + Reject</b>	STM32F405 (100MSa/s Power) STM32F405 (25MSa/s Power)	PTA+CPA	96 ( $c$ ) / <b>300</b> ( $s_1$ ) 147 ( $c$ ) / <b>3,000</b> ( $s_1$ )

repetition produces multiple independently drawn realizations of an almost identical instruction and memory-access flow per signing trial. Finally, the accepted trial binds publicly to their challenges via the signature output, whereas the rejected attempts do not. Therefore, a natural question arises: can the rejected procedure be targeted for launching attacks?

Latest studies demonstrated that rejected signing trials can aid key recovery against ML-DSA [8]–[10] in a profiling attack context. Again, these attacks assume a profiled setting with access to secret key-labeled traces on a clone device. On the contrary, in this work, we study the exploitation of leakage from the rejected signing trials under a non-profiled threat model. The guiding question is whether publicly derivable information from accepted signatures suffices to reconstruct the missing per-attempt challenges in rejected rounds with adequate accuracy, and whether leakage from both phases can then be consolidated to empower the following non-profiling attacks.

We answer the question positively by presenting a two-step attack pipeline in the non-profiling context. First, we recover all rejected challenges  $c$  by leveraging a public-based template attack. Specifically, we propose a shallow MLP and CNN-based models by training on the leakages from the final accepted trial, using their public challenges as labels. Then the trained models are applied to rejected signing trials to infer the rejected challenges. Second, we perform CPA by targeting  $cs_1$  with the recovered  $c$  and aggregated traces from both rejected and accepted signing trials. When an implementation exhibits on average  $n_r$  rejections per acceptance, the exploitable samples per signature increase roughly by a factor of  $n_r + 1$ . As a result, the overall number of traces (signatures) to achieve a successful key recovery attack reduces to around  $\frac{1}{n_r + 1}$ .

In summary, our main contributions are as follows:

- The first non-profiling attack against ML-DSA that systematically exploits leakage from rejected trials, cutting the required number of traces by 90% (from 3,000 down to 300) through improved leakage modeling and acquisition.
- A novel *public-based template attack* (PTA) constructed entirely from public values to recover the challenge  $c$ . Our PTA achieves a success rate of 100% by a shallow MLP/CNN model, with only 96 signatures.
- Experimental validation on real-world power measurements acquired from an ARM Cortex-M4 microcontroller, demonstrating full secret key recovery with around 300 signatures to achieve a success rate of 96.2%.

In particular, we clarify the concept of PTA by differentiating it from the conventional secret-based TA (STA), which needs secret-related intermediate values to build templates (profiles).

However, PTA still needs a profiling phase. We highlight our attack’s applicability that can succeed without modification across all ML-DSA security levels and also defeats the hedged mode of it. This, in turn, exposes a critical physical vulnerability inherent to rejection sampling in lattice-based signatures. Our results, moreover, call for implementing effective countermeasures such as high-order masking [21] across all signing trials in ML-DSA implementations.

## II. PRELIMINARIES

### A. Notations

Let  $\mathbb{Z}$  be the integers,  $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$  with prime  $q$ , and  $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ ; vectors in  $R_q^\ell$  are bold lowercase (e.g.,  $\mathbf{s}_1, \mathbf{s}_2, \mathbf{y}, \mathbf{z}$ ), matrices in  $R_q^{k \times \ell}$  are bold uppercase (e.g.,  $\mathbf{A}$ ), coefficients of  $w \in R_q$  use subscripts  $w_i$  and entries of  $\mathbf{w} \in R_q^\ell$  use bracket indices  $\mathbf{w}[j]$ . Centered modular reduction is  $r' \equiv r \bmod \pm q \in (-q/2, q/2]$ ; the infinity norms are  $\|w\|_\infty = \max_i |w_i \bmod \pm q|$  for  $w \in R_q$  and  $\|\mathbf{w}\|_\infty = \max_j \|\mathbf{w}[j]\|_\infty$  for  $\mathbf{w} \in R_q^\ell$ . For bounded sets we use  $S_\eta = \{a \in R_q : \|a\|_\infty \leq \eta\}$ ; the sparse ternary ball is  $B_\tau = \{c \in R_q : \#\{i : c_i \neq 0\} = \tau, c_i \in \{\pm 1\}\}$ .

The Number Theoretic Transform (NTT) of a polynomial  $c$  is  $\hat{c} = \text{NTT}(c)$  with inverse INTT; pointwise (Hadamard) multiplication in the NTT domain is denoted by  $\circ$ ; ring multiplication in  $R_q$  is written by juxtaposition, e.g.,  $cs_1$ ; for brevity we use  $\langle cs_1 \rangle := \text{INTT}(\hat{c} \circ \hat{s}_1)$ .

### B. Module-Lattice-Based Digital Signature Algorithm

ML-DSA is the FIPS 204 digital signature scheme built over the ring  $R_q = \mathbb{Z}_q[X]/(X^n + 1)$  with fixed  $(n, q) = (256, 2^{23} - 2^{13} + 1)$  across all parameter sets [1]. Its security relies on the hardness of the Module Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems, reducible to hard lattices [22] with tight security in the Quantum Random Oracle Model (QROM) [23]. ML-DSA comprises key generation, signing, and verification. The standard distinguishes external APIs (KEYGEN, SIGN) from internal routines (KEYGEN\_INTERNAL, SIGN\_INTERNAL) [1]. The external layer handles only byte-level formatting and seed/encoding; all arithmetic resides in the internal routines. We therefore specify KEYGEN\_INTERNAL to fix the key material and analyze SIGN\_INTERNAL, where physical leakage and rejection sampling occur, ensuring standard-aligned notation and reproducible results across compliant implementations.

The key generation algorithm generates a uniform public matrix  $\mathbf{A}$  and secret vectors,  $\mathbf{s}_1$  and  $\mathbf{s}_2$ , whose polynomial coefficients are sampled from a narrow distribution  $S_\eta$  (e.g.,

---

**Algorithm 1** ML-DSA.SIGN\_INTERNAL( $sk, M', rnd$ )

---

**Require:** Secret key  $sk$ ; formatted message  $M'$ ; per-message randomness or dummy  $rnd \in \{0, 1\}^{32}$

**Ensure:** Signature  $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$

```

1:  $(\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0) \leftarrow \text{SKDECODE}(sk)$ 
2:  $\hat{\mathbf{s}}_1 \leftarrow \text{NTT}(\mathbf{s}_1)$ ;  $\hat{\mathbf{s}}_2 \leftarrow \text{NTT}(\mathbf{s}_2)$ ;  $\hat{\mathbf{t}}_0 \leftarrow \text{NTT}(\mathbf{t}_0)$ 
3:  $\hat{\mathbf{A}} \leftarrow \text{EXPANDA}(\rho)$ 
4:  $\mu \leftarrow H(\text{BYTESTOBITS}(tr) \parallel M', 64)$ 
5:  $\rho' \leftarrow H(K \parallel rnd \parallel \mu, 64)$ ;  $\kappa \leftarrow 0$ 
6:  $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 
7: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do
8:    $\mathbf{y} \leftarrow \text{EXPANDMASK}(\rho', \kappa) \in R_q^\ell$ 
9:    $\mathbf{w} \leftarrow \text{INTT}(\hat{\mathbf{A}} \circ \text{NTT}(\mathbf{y}))$ 
10:   $\mathbf{w}_1 \leftarrow \text{HIGHBITS}_q(\mathbf{w})$ 
11:   $\tilde{c} \leftarrow H(\mu \parallel \text{W1ENCODE}(\mathbf{w}_1), \lambda/4)$ 
12:   $c \leftarrow \text{SAMPLEINBALL}(\tilde{c}) \in B_\tau$ 
13:   $\hat{c} \leftarrow \text{NTT}(c)$  /*Target operation*/
14:   $\mathbf{z} \leftarrow \mathbf{y} + \text{INTT}(\hat{c} \circ \hat{\mathbf{s}}_1)$  /*Target operation*/
15:   $\mathbf{r}_0 \leftarrow \text{LOWBITS}_q(\mathbf{w} - \text{INTT}(\hat{c} \circ \hat{\mathbf{s}}_2))$ 
16:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta \vee \|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
17:     $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 
18:  else
19:     $\langle \mathbf{ct}_0 \rangle \leftarrow \text{INTT}(\hat{c} \circ \hat{\mathbf{t}}_0)$ 
20:     $\mathbf{h} \leftarrow \text{MAKEHINT}_q(\langle \mathbf{ct}_0 \rangle, \mathbf{w} - \text{INTT}(\hat{c} \circ \hat{\mathbf{s}}_2) + \langle \mathbf{ct}_0 \rangle)$ 
21:    if  $\|\langle \mathbf{ct}_0 \rangle\|_\infty \geq \gamma_2 \vee \#\mathbf{1}(\mathbf{h}) > \omega$  then
22:       $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 
23:    end if
24:  end if
25:   $\kappa \leftarrow \kappa + \ell$ 
26: end while
27: return  $\sigma \leftarrow \text{SIGENCODE}(\tilde{c}, \mathbf{z} \bmod \pm q, \mathbf{h})$ 

```

---

integers in  $[-\eta, \eta]$ ). The public key consists of  $\rho$  (a seed for  $\mathbf{A}$ ) and  $\mathbf{t}_1$ , where  $\mathbf{t}_1$  is derived from  $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ .

The signing process, shown in Algorithm 1, is our target. A fresh mask<sup>1</sup> vector  $\mathbf{y}$  with small coefficients is generated for each signature. The core of the signature is computing  $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ , where  $c$  is a sparse challenge polynomial with  $\tau$  non-zero coefficients of  $\pm 1$ . A critical security feature of ML-DSA is rejection sampling [18], [19]: if the infinity norm of the coefficients in  $\mathbf{z}$  or an intermediate vector  $\mathbf{r}_0$  exceeds a specific bound, the signature is discarded, and the process restarts with a new  $\mathbf{y}$ . This ensures that the final public signature  $\mathbf{z}$  does not leak statistical information about the private key  $\mathbf{s}_1$ .

ML-DSA offers three parameter sets, denoted ML-DSA-44/65/87, which differ only in module dimensions and bounds while keeping  $(n, q)$  fixed, as summarized in Table II.

### C. Polynomial Multiplications in ML-DSA

Polynomial multiplication is the computational core of ML-DSA and is accelerated using NTT, which reduces the complexity from  $\mathcal{O}(n^2)$  to  $\mathcal{O}(n \log n)$ . As a result, polynomial multiplication operations like  $c\mathbf{s}_1$  are computed by transforming

<sup>1</sup>This algorithmic mask is structurally distinct from the masking schemes typically deployed as countermeasures against side-channel analysis.

TABLE II: ML-DSA parameters [1]

	ML-DSA-44	ML-DSA-65	ML-DSA-87
$q$ (modulus)	$2^{23} - 2^{13} + 1$	$2^{23} - 2^{13} + 1$	$2^{23} - 2^{13} + 1$
$\tau$ (# of $\pm 1$ 's in $c$ )	39	49	60
$\gamma_1$ ( $y$ -range bound)	$2^{17}$	$2^{19}$	$2^{19}$
$\eta$ (secret key range)	2	4	2
$\beta$ ( $=\tau\eta$ )	78	196	120

the polynomials to the NTT domain, performing coefficient-wise multiplication, and then transforming the result back.

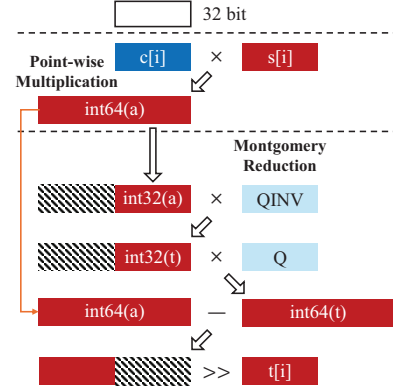


Fig. 1: The procedure of `montgomery_reduce` function. It contains two distinct leaking points: the initial product  $a$  and the final result  $t$  (adapted from [11]).

As illustrated in Figure 1, the pointwise multiplication, `poly_pointwise_montgomery`, is the target of our attack. This function iterates through the 256 coefficients, performing a multiplication followed by a Montgomery reduction. The literature [12], [13] has identified and experimentally verified two primary points of leakage within this loop:

- **Leaking Point 1:** The initial 64-bit product,  $a = c[i] \cdot s[i]$ , where  $c[i]$  is a known public value (during accepted signing trial) and  $s[i]$  is a secret coefficient of  $\hat{\mathbf{s}}_1$ .
- **Leaking Point 2:** The final 32-bit result,  $t[i]$ , after the subtraction and shift. Interestingly, this operation produces a cleaner signal than that of the leaking point 1.

## III. RECOVERING REJECTED CHALLENGES VIA PUBLIC-BASED TEMPLATE ATTACK

This section introduces a novel approach to recovering the rejected challenges  $c$  in ML-DSA using *Public-based Template Attack* (PTA). Unlike traditional Secret-based Template Attacks (STA) [9], [10] which require a clone device to profile sensitive intermediates, PTA aligns with the *non-profiling threat model*. It constructs leakage templates by utilizing publicly available challenge labels solely from the accepted signing trials on the target device. Specifically, we utilize MLP/CNN to model the leakages from the valid challenges and then to recover the rejected ones, reducing dependency on device-specific profiling and improving efficiency compared to existing methods.

### A. Rejection Sampling in ML-DSA

As shown in Algorithm 1, in ML-DSA, a signing trial is accepted if the following conditions hold:

$$\|\mathbf{z}\|_\infty < \gamma_1 - \beta, \quad \|\mathbf{r}_0\|_\infty < \gamma_2 - \beta, \quad \#\mathbf{1}(\mathbf{h}) \leq \omega. \quad (1)$$

TABLE III: Number of Rejections in ML-DSA.

Parameter set	$\mathbb{E}[T]$	$p_{\text{acc}} = 1/\mathbb{E}[T]$	$\mathbb{E}[R] = \mathbb{E}[T] - 1$
ML-DSA-44	4.25	0.235	3.25
ML-DSA-65	5.10	0.196	4.10
ML-DSA-87	3.85	0.260	2.85

Otherwise, the trial is rejected, and a fresh challenge  $c$  is drawn. The probability of acceptance,  $p_{\text{acc}}$ , depends on the joint probability of those conditions being met. Let  $T$  be the total number of signing trials (including the final accepted one). We estimate acceptance probabilities in a black-box manner by counting admissible values of uniformly drawn coefficients  $y_i$  inside the centered ranges of width  $2\gamma_1$  and  $2\gamma_2$ .

We then investigate the probability of per-coefficient acceptance on  $z_i$  and overall vector-level acceptance on  $\mathbf{z}$  as follows.

a) *Per-coefficient acceptance*: Write the polynomial products  $x = cs_1$  and  $x' = cs_2$ . Since  $c \in B_\tau$  and  $|(s_1)_j| \leq \eta$ , it follows that  $|x_i| \leq \beta$  and similarly  $|x'_i| \leq \beta$ . For any fixed  $x_i$  with  $|x_i| \leq \beta$ , the predicate  $|z_i| < \gamma_1 - \beta$  is satisfied by exactly  $2(\gamma_1 - \beta) - 1$  values of  $y_i$  among  $2\gamma_1$  candidates, hence

$$p_z^{(\text{coef})} = \Pr[|z_i| < \gamma_1 - \beta] = \frac{2(\gamma_1 - \beta) - 1}{2\gamma_1}. \quad (2)$$

The same counting for  $r_0$  yields

$$p_{r_0}^{(\text{coef})} = \Pr[|(r_0)_i| < \gamma_2 - \beta] = \frac{2(\gamma_2 - \beta) - 1}{2\gamma_2}. \quad (3)$$

Both expressions are entirely independent of the actual values of  $x_i, x'_i$  as long as the bounds hold.

b) *Vector-level acceptance and signing trial success probability*: Approximating coordinates as independent for each coefficient, the per-vector acceptance probabilities are

$$p_z \approx \left(p_z^{(\text{coef})}\right)^n, \quad p_{r_0} \approx \left(p_{r_0}^{(\text{coef})}\right)^n. \quad (4)$$

Let  $p_h = \Pr[\#\mathbf{1}(\mathbf{h}) \leq \omega] \in (0, 1]$  denote the hint-weight acceptance (absorbing implementation-specific rounding interactions). Assuming the three predicates in Equation (1) are weakly dependent, a conservative product lower bound is

$$\begin{aligned} p_{\text{acc}} &\gtrsim p_z \cdot p_{r_0} \cdot p_h \\ &= \left(\frac{2(\gamma_1 - \beta) - 1}{2\gamma_1}\right)^n \left(\frac{2(\gamma_2 - \beta) - 1}{2\gamma_2}\right)^n \cdot p_h. \end{aligned} \quad (5)$$

Therefore, the number of rejections before the first acceptance follows a geometric distribution in the average case:

$$R \sim \text{Geom}(p_{\text{acc}}), \quad \mathbb{E}[R] = \frac{1 - p_{\text{acc}}}{p_{\text{acc}}}.$$

The probability of requiring more than  $n_t$  attempts (including the acceptance) follows the geometric tail:

$$\Pr[T \geq n_t] = (1 - p_{\text{acc}})^{n_t}.$$

Table III reports the expected number of attempts  $\mathbb{E}[T]$  for the three parameter sets used in ML-DSA. For instance, with  $p_{\text{acc}} = 0.196$  (for ML-DSA-65), the expected number of rejections is 4.10, and the probability of more than 814 total attempts is negligible:  $\Pr[T > 814] \leq 2^{-256}$ .

## B. Public-based Template Attack (PTA) Framework

We aim to recover rejected challenge coefficients as integers  $c_i \in \{0, \pm 1\}$ —unlike hints [24]—without a clone device. Since the adversary initially has no knowledge of the rejected challenges, standard secret-based profiling (STA) is impossible. Our proposed PTA treats valid signing trials as a labeled dataset for public-based profiling. Such approach is justified by two fundamental properties of ML-DSA implementation:

Firstly, the function `poly_ntt(c)` executes identically in valid and rejected trials, yielding consistent leakage among all signing trials; and secondly, the challenge coefficients are drawn from the same distribution  $B_\tau$  within every trial.

Leveraging these properties, the PTA proceeds in two phases:

1) *Phase 1: Public Profiling (on Accepted Trials)*: The adversary collects a set of signatures. For each signature  $k$ , the accepted challenge  $c^{(k)}$  is publicly known from the signature bundle. We extract the trace segments  $\mathbf{t}^{(k)}$  corresponding to the execution of `NTT(c^{(k)})`. These public challenges serve as ground-truth labels to train a probabilistic classifier  $\mathcal{M}$ :

$$\mathcal{M} \leftarrow \text{Train} \left( \left\{ (\mathbf{t}^{(k)}, c^{(k)}) \right\}_{k=1}^{N_{\text{acc}}} \right) \quad (6)$$

2) *Phase 2: Attack Recovery (on Rejected Trials)*: For a rejected trial  $r$ , the challenge  $c^{(r)}$  is unknown. The adversary captures the corresponding trace  $\mathbf{t}^{(r)}$  and queries the pre-trained model  $\mathcal{M}$  to predict the coefficients:

$$\tilde{c}^{(r)} = \underset{v \in \{0, \pm 1\}}{\text{argmax}} \mathcal{M}(\mathbf{t}^{(r)}) \quad (7)$$

And these recovered data can be aggregated with valid traces for the subsequent Key Recovery CPA (Section IV).

## C. Model Architecture

In this work, we choose four PTA approaches: TA, pooled TA, CNN, and MLP, where the latter two extend the original PTA [15], [16] with neural network-based approaches. In particular, TA and pooled TA use correlation-based POI selection, while CNN and MLP operate directly on the raw traces. We use a CNN with a shallow one-dimensional convolutional network followed by dense layers, and the MLP has a single-hidden-layer network with dropout and L2 regularization. The corresponding hyperparameters are summarized in Table IV.

TABLE IV: CNN and MLP model architectures.

Model	Layer	Size / Params	Details
MLP	Dense	128	ReLU, L2 reg.: 0.01
	Dropout	same	Rate: 0.20
	Output	#classes	Softmax, Adam
CNN	Conv1D	K4, F16	ReLU, padding: same
	Dense	64	ReLU
	Output	#classes	Softmax, Adam

## IV. KEY RECOVERY VIA CORRELATION POWER ANALYSIS

With the rejected challenges successfully recovered via the Public-based Template Attack (PTA), we proceed to the final stage: extracting the full secret key  $\mathbf{s}_1$ . This section details our non-profiled attack methodology, which leverages Correlation

Power Analysis (CPA). Our central contribution is the joint exploitation of leakages from all signing trials—both rejected and accepted—associated with a single signature.

### A. Joint Exploitation of Leakages from All Signing Trials

The rejection sampling mechanism in ML-DSA, as detailed in Section III-A, dictates that for each valid signature, an average of 3.25 rejected trials occur for ML-DSA-44. While the challenge  $c$  is unique to each trial, the secret key  $s_1$  remains static throughout the entire process. Our PTA provides us with the challenge label for each of these trials, enabling a powerful data aggregation strategy that amplifies the exploitable leakage.

Our method involves capturing the full trace of each signing process, which contains multiple rejected trials and the final accepted one. We then align the leakage segments corresponding to the  $\hat{c} \circ \hat{s}_1$  computation from each trial. This essentially produces more traces for the following secret key recovery. We hypothesize that a CPA performed on this jointly exploited trace will be much more efficient than an attack on any single-trial trace, which we will validate in Section V.

### B. CPA Strategies for Key Recovery

*a) Conservative CPA:* The first strategy is the “conservative approach”, a direct, brute-force CPA for each coefficient of  $s_1$ , as detailed in [12]. The adversary iterates through all possible values in the NTT domain (a space of size close to  $2^{27}$ ), computes the correlation coefficient for each key guess, and identifies the key guess with the highest correlation coefficient. While computationally intensive (“time-inefficient”), this method is highly successful and requires fewer traces to succeed. Our joint exploitation technique makes this approach practical with very few signature captures.

*b) Two-Stage CPA:* The second strategy is the “fast two-stage approach” [12], [13], designed to reduce the high time complexity of the conservative method.

- Stage 1 (LSB Filtering): An initial, rapid CPA is performed on only the  $m$  least significant bits (LSBs) of the secret coefficient. This step identifies a small list of candidates.
- Stage 2 (Focused Recovery): A second CPA is performed on the remaining bits, but only for the candidates that passed Stage 1. This drastically prunes the search space.

By aggregating the traces from both rejected and accepted trials, we significantly amplify the dataset from each signature (trace), thereby enhancing the statistical power of the CPA.

## V. EXPERIMENTAL RESULTS

This section evaluates the practical efficacy of our proposed two-stage attack using acquired power traces. We begin by outlining the experimental environment, including the target device and data acquisition setup. We then present our findings in two main parts. First, we validate the performance of PTA, demonstrating its ability to recover the rejected challenge  $c$  with 100% accuracy. Second, using the perfectly recovered challenge, we assess the efficiency of the subsequent key recovery stage, where the full secret key  $s_1$  is recovered with a remarkably low requirement of approximately 300 traces.

### A. Experimental Setup

We target the unprotected, C-based reference implementation of ML-DSA-44 that was submitted to NIST during its 3rd-round evaluation. The implementation runs on a ChipWhisperer UFO development board fitted with a 32-bit STM32F405GTx ARM Cortex-M4 microcontroller. Power consumption was measured using WaveRunner 9104 oscilloscope, connected to a Mini-Circuits BLP-40+ low-pass filter and a PA303 pre-amplifier. All experiments focus on the signing procedure, where each trace (measurement) contains both rejected and valid signing trials. In total, 10,000 power traces were acquired for each sampling rate setting (25 MSa/s and 100 MSa/s).

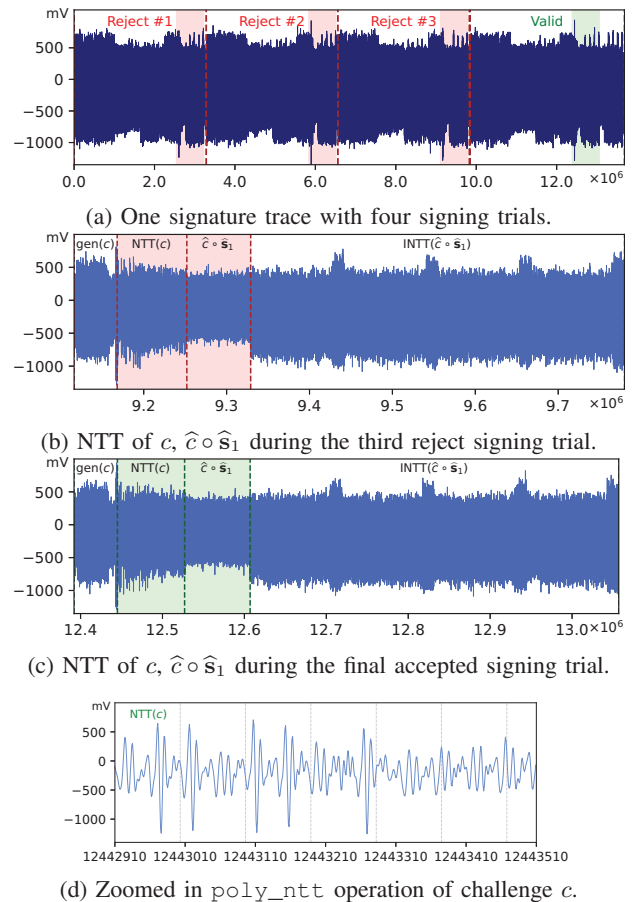


Fig. 2: Power traces of the reference ML-DSA-44 implementation on STM32F405, captured at 25MSa/s.

To highlight the impact of the rejection mechanism and validate our approach, we collected traces corresponding to the final four signing trials (including one accepted one) of each signing process. It is worth noting that there are 4.25 trials for each input message  $msg$ , and a larger range of measurements can be collected with more trials. Overall, our experimental setups ensure that the collected dataset covers leakage from the rejected signing trials and the final valid signature.

As illustrated in Figure 2a, we take a trace with four signing trials as an example, and the target regions are highlighted with colors. Figures 2b and 2c are zoomed into the NTT segment of the third rejected signing trial and the final valid signing trial, respectively. The marked regions highlight the

two operations we target in this work:  $\text{poly\_ntt}(c)$  and  $\text{polyvecl\_pointwise\_montgomery\_reduce}(c, s_1)$ . At last, Figure 2d shows the trace segments for the first few coefficients of  $c$  (the first layer of NTT operation). In particular, we shall observe distinct features for different values of  $c_i$ . However, distinguishing all 256 coefficients  $c_i$  with 100% accuracy is still not feasible without PTA.

### B. Recovering Challenges via PTA

We present the results in Table V of success rates in recovering rejected challenges  $c$ . Moreover, the evolution of the success rate along with the number of traces is depicted in Figure 3 for the sampling rates of 100MSa/s and 25MSa/s, respectively. The main takeaway is that PTA with CNN outperforms others, resulting in recovering all 256 coefficients of challenge  $c$  with 96 and 147 traces in two acquisition scenarios, respectively.

TABLE V: Recovery accuracy of challenge  $c$  in  $\text{poly\_ntt}$  on ML-DSA-44 via different profiling approaches.

Dataset	Approach <sup>1</sup>	Profiling Traces	SR <sub>128:255</sub>	SR <sub>0:127</sub>
100MSa/s	TA	96	97.72%	88.58%
	Pooled TA	96	99.19%	97.31%
	MLP	96	99.93%	96.74%
	CNN	96	<b>100%</b>	<b>100%</b>
25MSa/s	TA	147	94.79%	80.09%
	Pooled TA	147	96.42%	90.50%
	MLP	147	98.77%	93.31%
	CNN	147	<b>100%</b>	<b>100%</b>

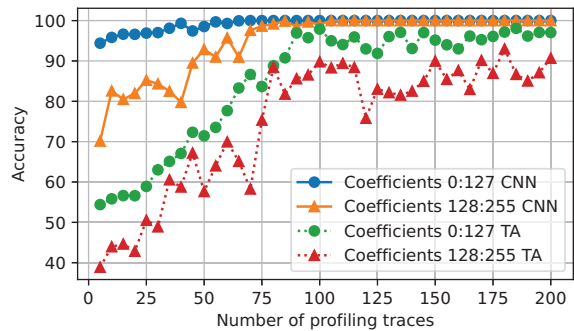
<sup>1</sup> TA and Pooled TA select the top 20 PoIs of Pearson correlation coefficients, while the shallow MLP and CNN use the entire sub-traces.

### C. Secret Key Recovery by CPA

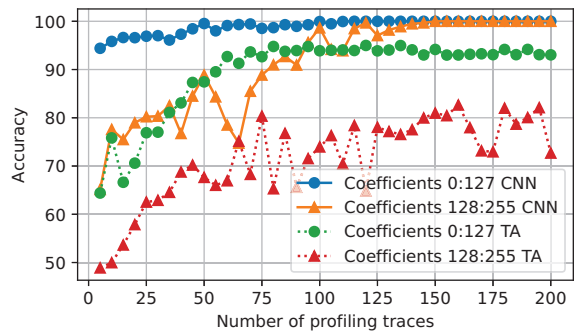
Next, we present the results of secret key recovery as in Figure 4 by applying the two-stage CPA on two datasets. The comparative results of ‘R’ and ‘V’ represent rejected and valid signing trials, respectively. In both scenarios, the leakage from the two trials shows consistency, validating our conjecture on the same operation  $\hat{c} \circ \hat{s}_1$ . The results show that the joint exploitation of leakages from both the multiple rejected and the final valid trials (marked as ‘3R+V’) shows significant improvements in the number of recovered coefficients with a fixed number of traces. In particular, if one target is recovering 192 coefficients in the dataset of 100MSa/s, the required number of traces for success is about 150 traces for the joint exploited case, while it requires around 600 traces for a single rejected (‘R’) trial or valid trial (‘V’), with a factor of 4x.

### D. Extending to Other ML-DSA and Hedged Mode

The new attack surface we revealed in this work only target the NTT operation of  $c$  and the multiplication  $\hat{c} \circ \hat{s}_1$ . Therefore, our approach can be directly applied across all three ML-DSA variants with different security levels. The same reasoning exists for applying in the hedged (non-deterministic) mode [1] of ML-DSA given the same algorithmic structure from a side-channel perspective, demonstrating that the hedging offers no SCA protection in this scenario.

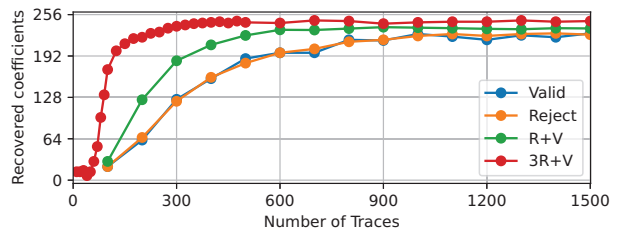


(a) Dataset collected at 100MSa/s

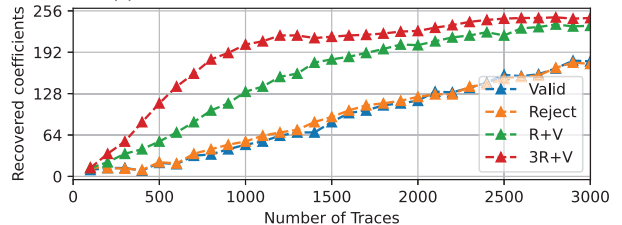


(b) Dataset collected at 25MSa/s

Fig. 3: Coefficient recovery accuracy of  $c$  in the integer domain by using TA and CNN, respectively.



(a) On the dataset collected at 100MSa/s



(b) On the dataset collected at 25MSa/s

Fig. 4: Experimental results of coefficient recovery by utilizing two-stage CPA.

## VI. CONCLUSIONS AND PERSPECTIVES

In this work, we revealed a new attack surface against ML-DSA that exploits rejection sampling by utilizing the Public-based Template Attack (PTA). Unlike prior Secret-based approaches that rely on clone devices to profile the leakage of the sensitive intermediates, our method builds leakage templates solely from public data available on the target device itself. By recovering rejected challenges via PTA and aggregating leakages from all signing trials, we achieve a trace reduction factor roughly equal to the average number of signing trials.

From a perspective, one can optimize the joint exploitation of leakages to launch efficient attacks, and also target other NIST candidates or finalists that employ rejection sampling.

## REFERENCES

- [1] National Institute of Standards and Technology (NIST), “FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA),” U.S. Department of Commerce, Standard, August 2024, federal Information Processing Standards Publication 204. [Online]. Available: <https://csrc.nist.gov/pubs/fips/204/final>
- [2] P. C. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *CRYPTO '99, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, ser. LNCS, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 388–397. [Online]. Available: [https://doi.org/10.1007/3-540-48405-1\\_25](https://doi.org/10.1007/3-540-48405-1_25)
- [3] G. Alagic, D. Cooper, Q. Dang, T. Dang, J. M. Kelsey, J. Lichtinger, Y.-K. Liu, C. A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and D. Apon, “Status report on the third round of the nist post-quantum cryptography standardization process,” 2022.
- [4] R. Primas, P. Pessl, and S. Mangard, “Single-trace side-channel attacks on masked lattice-based encryption,” in *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, ser. Lecture Notes in Computer Science, W. Fischer and N. Homma, Eds., vol. 10529. Springer, 2017, pp. 513–533. [Online]. Available: [https://doi.org/10.1007/978-3-319-66787-4\\_25](https://doi.org/10.1007/978-3-319-66787-4_25)
- [5] H. Maghrebi, T. Portigliatti, and E. Prouff, “Breaking cryptographic implementations using deep learning techniques,” in *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, ser. Lecture Notes in Computer Science, C. Carlet, M. A. Hasan, and V. Saraswat, Eds., vol. 10076. Springer, 2016, pp. 3–26. [Online]. Available: [https://doi.org/10.1007/978-3-319-49445-6\\_1](https://doi.org/10.1007/978-3-319-49445-6_1)
- [6] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, “Deep learning for side-channel analysis and introduction to ASCAD database,” *J. Cryptogr. Eng.*, vol. 10, no. 2, pp. 163–188, 2020. [Online]. Available: <https://doi.org/10.1007/s13389-019-00220-8>
- [7] V. Q. Ulitzsch, S. Marzougui, M. Tibouchi, and J. Seifert, “Profiling side-channel attacks on dilithium - A small bit-fiddling leak breaks it all,” in *Selected Areas in Cryptography - 29th International Conference, SAC 2022, Windsor, ON, Canada, August 24-26, 2022, Revised Selected Papers*, ser. Lecture Notes in Computer Science, B. Smith and H. Wu, Eds., vol. 13742. Springer, 2022, pp. 3–32. [Online]. Available: [https://doi.org/10.1007/978-3-031-58411-4\\_1](https://doi.org/10.1007/978-3-031-58411-4_1)
- [8] O. Bronchain, M. Azouaoui, M. ElGhamrawy, J. Renes, and T. Schneider, “Exploiting small-norm polynomial multiplication with physical attacks application to crystals-dilithium,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, no. 2, pp. 359–383, 2024. [Online]. Available: <https://doi.org/10.46586/tches.v2024.i2.359-383>
- [9] Y. Zhou, W. Wang, Y. Sun, and Y. Yu, “Rejected signatures’ challenges pose new challenges: Key recovery of crystals-dilithium via side-channel attacks,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2025, no. 4, pp. 817–847, 2025. [Online]. Available: <https://doi.org/10.46586/tches.v2025.i4.817-847>
- [10] Z. Liu, A. Wang, C. Wei, Y. Ding, J. Zhang, A. Liu, and L. Zhu, “Release the power of rejected signatures: An efficient side-channel attack on the ML-DSA cryptosystem,” *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 13 356–13 369, 2025. [Online]. Available: <https://doi.org/10.1109/TIFS.2025.3643784>
- [11] H. Fan, H. Zhang, Y. Wang, W. Wang, H. Zhang, and Q. Yuan, “Multivariate template attack against ntt-based polynomial multiplication of dilithium,” *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 8570–8582, 2025. [Online]. Available: <https://doi.org/10.1109/TIFS.2025.3598577>
- [12] Z. Chen, E. Karabulut, A. Aysu, Y. Ma, and J. Jing, “An efficient non-profiled side-channel attack on the crystals-dilithium post-quantum signature,” in *39th IEEE International Conference on Computer Design, ICCD 2021, Storrs, CT, USA, October 24-27, 2021*. IEEE, 2021, pp. 583–590. [Online]. Available: <https://doi.org/10.1109/ICCD53106.2021.00094>
- [13] Y. Liu, Y. Liu, Y. Zhou, Y. Gao, Z. Qiao, and H. Wang, “A novel power analysis attack against crystals-dilithium implementation,” in *IEEE European Test Symposium, ETS 2024, The Hague, Netherlands, May 20-24, 2024*. IEEE, 2024, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/ETS61313.2024.10567325>
- [14] T. Tosun and E. Savas, “Zero-value filtering for accelerating non-profiled side-channel attack on incomplete ntt-based implementations of lattice-based cryptography,” *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 3353–3365, 2024. [Online]. Available: <https://doi.org/10.1109/TIFS.2024.3359890>
- [15] Y. Liu, Y. Zhou, S. Sun, T. Wang, R. Zhang, and J. Ming, “On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1868–1879, 2021. [Online]. Available: <https://doi.org/10.1109/TIFS.2020.3045904>
- [16] Z. Qiao, Y. Liu, Y. Zhou, J. Ming, C. Jin, and H. Li, “Practical public template attack attacks on crystals-dilithium with randomness leakages,” *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1–14, 2023. [Online]. Available: <https://doi.org/10.1109/TIFS.2022.3215913>
- [17] Z. Qiao, Y. Liu, Y. Zhou, M. Shao, and S. Sun, “When NTT meets SIS: efficient side-channel attacks on dilithium and kyber,” *IACR Cryptol. ePrint Arch.*, p. 1866, 2023. [Online]. Available: <https://eprint.iacr.org/2023/1866>
- [18] V. Lyubashevsky, “Fiat-shamir with aborts: Applications to lattice and factoring-based signatures,” in *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009, Proceedings*, ser. Lecture Notes in Computer Science, M. Matsui, Ed., vol. 5912. Springer, 2009, pp. 598–616. [Online]. Available: [https://doi.org/10.1007/978-3-642-10366-7\\_35](https://doi.org/10.1007/978-3-642-10366-7_35)
- [19] V. Lyubashevsky, “Lattice signatures without trapdoors,” in *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012, Proceedings*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, 2012, pp. 738–755. [Online]. Available: [https://doi.org/10.1007/978-3-642-29011-4\\_43](https://doi.org/10.1007/978-3-642-29011-4_43)
- [20] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “Crystals-dilithium: A lattice-based digital signature scheme,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 238–268, 2018. [Online]. Available: <https://doi.org/10.13154/tches.v2018.i1.238-268>
- [21] G. Barthe, S. Belaïd, T. Espitau, P. Fouque, B. Grégoire, M. Rossi, and M. Tibouchi, “Masking the GLP lattice-based signature scheme at any order,” in *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, ser. Lecture Notes in Computer Science, J. B. Nielsen and V. Rijmen, Eds., vol. 10821. Springer, 2018, pp. 354–384. [Online]. Available: [https://doi.org/10.1007/978-3-319-78375-8\\_12](https://doi.org/10.1007/978-3-319-78375-8_12)
- [22] A. Langlois and D. Stehlé, “Worst-case to average-case reductions for module lattices,” *Des. Codes Cryptogr.*, vol. 75, no. 3, pp. 565–599, 2015. [Online]. Available: <https://doi.org/10.1007/s10623-014-9938-4>
- [23] E. Kiltz, V. Lyubashevsky, and C. Schaffner, “A concrete treatment of fiat-shamir signatures in the quantum random-oracle model,” in *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, ser. Lecture Notes in Computer Science, J. B. Nielsen and V. Rijmen, Eds., vol. 10822. Springer, 2018, pp. 552–586. [Online]. Available: [https://doi.org/10.1007/978-3-319-78372-7\\_18](https://doi.org/10.1007/978-3-319-78372-7_18)
- [24] D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, “LWE with side information: Attacks and concrete security estimation,” in *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, ser. Lecture Notes in Computer Science, D. Micciancio and T. Ristenpart, Eds., vol. 12171. Springer, 2020, pp. 329–358. [Online]. Available: [https://doi.org/10.1007/978-3-030-56880-1\\_12](https://doi.org/10.1007/978-3-030-56880-1_12)