

# SHIFT SNARE: Uncovering Secret Keys in FALCON via Single-Trace Analysis

Jinyi Qiu<sup>†</sup>, Aydin Aysu<sup>†</sup>

<sup>†</sup>Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA

**Abstract**—This paper presents a novel *single-trace* side-channel attack on FALCON, a lattice-based post-quantum digital signature protocol recently approved for standardization by NIST. We target the discrete Gaussian sampling operation within FALCON’s key generation scheme and demonstrate that a single power trace is sufficient to mount a successful attack. Notably, negating the results of a 63-bit right-shift operations on 64-bit secret values leaks critical information about the assignment of ‘-1’ versus ‘0’ to intermediate coefficients during sampling. These leaks enable full recovery of the secret key.

We demonstrate a ground-up approach to the attack on an ARM Cortex-M4 microcontroller executing both the reference and optimized implementations from FALCON’s NIST round 3 software package. We successfully recovered all of the secret polynomials in FALCON. We further quantify the attacker’s success rate using a univariate Gaussian template model, providing generalizable guarantees. Statistical analysis with over 500,000 tests reveals a per-coefficient success rate of 99.999999478% and a full-key recovery rate of 99.99994654% for FALCON-512. We verify that this vulnerability is present in all implementations included in FALCON’s NIST submission package. This highlights the vulnerability of current software implementations to single-trace attacks and underscores the urgent need for single-trace-resilient software in embedded systems.

**Index Terms**—Side-channel attacks, Post-quantum cryptography, NTRU, FALCON, Key generation, Lattice-based cryptography, Digital signature schemes

## I. INTRODUCTION

Widely adopted encryption schemes such as RSA [1] and elliptic curve cryptosystems [2] rely on hard mathematical problems, including integer factorization [3] and the discrete logarithm problem [4], which are traditionally regarded as computationally intractable for classical computers. However, quantum algorithms offer exponential speedups for solving these problems [5]. As a result, advances in quantum computing pose a significant threat to conventional encryption schemes and underscore the critical need to design cryptographic systems that can resist quantum attacks.

To address this issue, the National Institute of Standards and Technology (NIST) initiated a standardization process for post-quantum cryptographic schemes, also referred to as quantum-resistant algorithms, designed to withstand quantum cryptanalysis [6]. As of now, this process has selected three digital signature schemes for standardization: CRYSTALS-Dilithium [7], SPHINCS+ [8], and FALCON [9]. NIST selected FALCON in part because of its small signature size, making it particularly suitable for embedded and bandwidth-constrained systems.

While the algorithms chosen are expected to be mathematically robust, their practical implementations may remain vulnerable to side-channel attacks. These attacks exploit physical characteristics of implementations, such as execution time, power consumption, and electromagnetic emissions, to extract secret information [10]. An attacker can carry out such attacks using only a few side-channel measurements from the physical device [11], [12], [13]. The most severe form of these attacks, known as *single-trace* attacks or simple power analysis, enables adversaries to recover secret data from just a single execution of the program. Single-trace attacks are particularly dangerous because they can bypass commonly deployed defenses such as masking [14]. Furthermore, they can target sensitive subroutines such as key generation, which produces a new secret on each invocation. FALCON’s suitability for embedded deployment makes it a prime target for side-channel exploitation. Given the imminent real-world deployment of NIST’s post-quantum cryptographic standards [15], [16], [17], there is a critical need to uncover such vulnerabilities and guide the development of effective countermeasures.

Previous work on *single-trace* side-channel analysis of lattice-based cryptosystems has identified several vulnerable components. Examples include the number-theoretic transform (NTT) [18], [19], polynomial multiplication [20], [21], message encoding/decoding [22], [23], cumulative distribution table (CDT) sampling [24], [25], and the Fujisaki–Okamoto transform [26]. Although FALCON incorporates some of these components, it also introduces unique elements such as fast Fourier sampling and floating-point arithmetic. Existing attacks cannot be directly applied to these FALCON-specific operations, highlighting the need for further investigation into vulnerabilities specific to FALCON.

In this paper, we present a *novel* single-trace side-channel vulnerability in FALCON that is distinct from previously reported attacks [25], [27], [28], [29], [30], [31]. Figure 1 provides a visual summary of the vulnerability and emphasizes its practical relevance. The top panel displays a power trace recorded during the discrete Gaussian sampling phase of FALCON key generation. The middle panel highlights the region of interest where data-dependent leakage is apparent. The bottom right panel presents a magnified view, illustrating the average power consumption corresponding to two secret value assignments: ‘0’ and ‘-1’. The bottom left panel shows the output of a univariate Gaussian model evaluated at the point of maximum leakage, clearly demonstrating a separation between the two classes. This separation confirms the presence

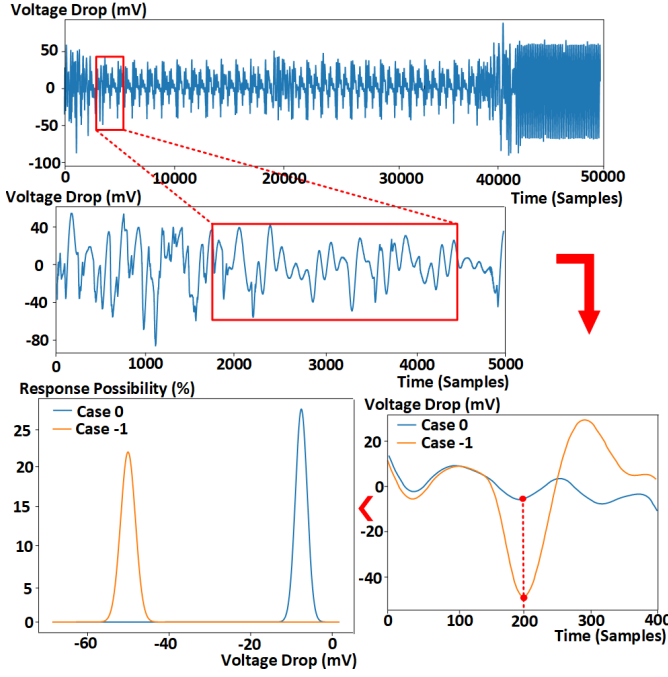


Fig. 1. Visual Demonstration of the Vulnerability: The top figure illustrates the power consumption profile of the device during the key generation process. The middle figure provides a zoomed-in view of the power consumption during a specific vulnerable segment in the code. The bottom right figure offers a further magnified view and the average power consumption for two distinct cases of secret value assignment ('0' versus '-1'). The bottom left figure presents the results of a univariate Gaussian model distinguishing these two classes over 500,000 trials. The analysis reveals that different assignments of secret intermediate values cause significant variations in power consumption. This demonstrates the practicality of mounting an attack with a substantial success rate.

of an exploitable vulnerability in FALCON's implementation. The contributions of this paper are as follows:

- We demonstrate a *new* side-channel vulnerability in FALCON's key generation process that enables full recovery of the secret key from a single power measurement.
- We present a custom method developed from first principles to recover the entire secret key. A theoretical model is constructed to provide a generalizable guarantee on the success rate.
- We apply the attack to an off-the-shelf device featuring an ARM Cortex-M4 microcontroller—a widely used platform for side-channel evaluation. The microcontroller executes the reference implementation from FALCON's NIST submission package. Our practical experiments achieve a 100% success rate, while the theoretical model predicts a per-variable success rate of 99.999999478% and a full key recovery rate higher than 99.99989309%. The attack is effective across all implementations within the NIST submission package, as the same leakage is consistently observed in both the reference and optimized versions.
- To support reproducibility and future research, we will publicly release our power traces and attack code upon acceptance of this paper.

## II. BACKGROUND

This section provides an overview of FALCON and explains the role of its secret polynomials. We also highlight the differences between our attack and previous side-channel attacks on FALCON. Finally, we present our threat model.

### A. The Generation of FALCON's Secret Polynomial

FALCON (Fast Fourier Lattice-based Compact Signature over NTRU) is a digital signature scheme designed for the post-quantum era. It provides security against quantum adversaries, as even quantum computers would require infeasible computational resources to break the mathematical trapdoors underlying the scheme. This resilience is derived from the Ring Learning with Errors (R-LWE) problem, which FALCON incorporates within the NTRU lattice framework. In comparison to other post-quantum algorithms, FALCON features relatively small key and signature sizes, enhancing its efficiency and making it well-suited for deployment in low-power devices and bandwidth-constrained environments.

FALCON comprises three main stages: key generation, signature generation, and signature verification. This work focuses on the key generation step, specifically the computation of the secret polynomials  $f$  and  $g$ .

---

#### Algorithm 1 FALCON Key Generation

---

**Require:** A monic polynomial  $\phi \in \mathbb{Z}[x]$ , a modulus  $q$

**Ensure:** A secret key  $sk$ , a public key  $pk$

---

- 1:  $f, g, F, G \leftarrow \text{NTRUGen}(\phi, q)$  ▷ Solving the NTRU equation
  - 2:  $\mathbf{B} \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$
  - 3:  $\hat{\mathbf{B}} \leftarrow \text{FFT}(\mathbf{B})$  ▷ Compute the FFT for each of the 4 components  $\{g, -f, G, -F\}$
  - 4:  $\mathbf{G} \leftarrow \hat{\mathbf{B}} \times \hat{\mathbf{B}}^*$
  - 5:  $\mathbf{T} \leftarrow \text{fftLDL}^*(\mathbf{G})$  ▷ Computing the LDL\* tree
  - 6: **for each** leaf of  $\mathbf{T}$  **do** ▷ Loop over the leaves of  $\mathbf{T}$
  - 7:   leaf.value  $\leftarrow \sigma / \sqrt{\text{leaf.value}}$  ▷ Normalization step
  - 8:  $sk \leftarrow (\hat{\mathbf{B}}, \mathbf{T})$
  - 9:  $h \leftarrow gf^{-1} \bmod q$
  - 10:  $pk \leftarrow h$
  - 11: **return**  $sk, pk$
- 

Algorithm 1 illustrates the steps involved in FALCON's key generation process. It begins with a predefined parameter  $n$  (set to either 512 or 1024) and a modulus  $q$  (set to 12289 in the NIST submission package), which define the size of the ring. Initially, the base NTRU lattice components,  $f$  and  $g$ , are generated. These components are then used to deterministically derive both the public and secret keys—without relying on randomness or additional secret variables. This deterministic derivation implies that **the base NTRU lattice components,  $g$  and  $f$ , are critically important, since both keys are fixed once  $f$  and  $g$  are generated.** Although the public key  $h$  is derived from these components, the process is non-invertible, meaning an adversary cannot reconstruct the

---

**Algorithm 2**  $NTRUGen(\phi, q)$ 

---

**Require:** A monic polynomial  $\phi \in \mathbb{Z}[x]$  of degree  $n$ , a modulus  $q$

**Ensure:** Polynomials  $f, g, F, G$

```
1:  $\sigma_{(f,g)} \leftarrow 1.17\sqrt{q}/2n$   $\triangleright \sigma_{(f,g)}$  is chosen so that  
    $\mathbb{E}[\|(f,g)\|] = 1.17\sqrt{q}$   
2: for  $i$  from 0 to  $n-1$  do  
3:    $f_i \leftarrow D_{\mathbb{Z}, \sigma_{(f,g)}}$   
4:    $g_i \leftarrow D_{\mathbb{Z}, \sigma_{(f,g)}, 0}$   
5: end for  
6:  $f \leftarrow \sum_i f_i x^i$   $\triangleright f \in \mathbb{Z}[x]/(\phi)$   
7:  $g \leftarrow \sum_i g_i x^i$   $\triangleright g \in \mathbb{Z}[x]/(\phi)$   
   ...  
8:  $F, G \leftarrow NTRUSolve_{n,q}(f, g)$   $\triangleright$  Computing  $F, G$  such  
   that  $fG - gF = q \pmod{\phi}$   
9: if  $(F, G) = 1$  then  
10:  restart  
11: end if  
12: return  $f, g, F, G$ 
```

---

base lattice components  $f$  and  $g$  from the public key  $h$ . The remainder of this subsection provides a brief explanation of how FALCON generates the public key  $pk$  and secret key  $sk$ , to contextualize the attack described in section III.

Algorithm 2,  $NTRUGen()$ , is the first step in generating the base NTRU lattice components  $f$  and  $g$  based on a Gaussian distribution. Lines 2 through 7 of the algorithm define  $f$  and  $g$  as polynomials of degree  $n$ , following the structure described in the equation below:

$$f(x) = f_0 + f_1x + f_2x^2 + f_3x^3 + \dots + f_{n-1}x^{n-1} \quad (1)$$

$$g(x) = g_0 + g_1x + g_2x^2 + g_3x^3 + \dots + g_{n-1}x^{n-1} \quad (2)$$

In the formula above, the coefficients of  $f$  and  $g$  are sampled discretely from a Gaussian distribution. The mean of this distribution is zero, and the standard deviation is determined by the degree  $n$  and the parameter  $q$ , as specified in line 1 of Algorithm 2. For FALCON-512,  $n$  is set to 512, whereas for FALCON-1024,  $n$  is set to 1024. In both parameter sets, the modulus  $q$  is fixed at 12289.

In line 8 of Algorithm 2, after generating the lattice base components, the key generation process solves the NTRU equation (Equation 3) using  $f$  and  $g$  to derive the secret key and the public key.

$$fG - gF = q \pmod{\phi} \quad (3)$$

To obtain  $F$  and  $G$ , which are essential for generating the public and secret keys, the algorithm recursively reduces the polynomials  $f$  and  $g$  into two polynomials whose degrees are half those of their predecessors. When the polynomial degree is reduced to 1, the base elements of  $F$  and  $G$  are obtained by finding the greatest common divisor (GCD) of  $f$  and  $g$ . Once the base elements of  $F$  and  $G$  are established, they are recursively combined, doubling the degree of the resulting polynomial at each step. This process continues until the polynomials  $F$  and  $G$  are reconstructed at degree  $n$ .

The remainder of the key generation process involves computing the fast Fourier transform (FFT), constructing the Gram matrix, and generating the FALCON tree based on this matrix. Notably, this portion of the process is deterministic and does not introduce additional randomness, provided that the NTRU lattice components  $f$ ,  $F$ ,  $g$ , and  $G$  are known.

### B. Comparison to Previous Attacks on FALCON

Previous studies investigating side-channel attacks on FALCON include the work of Karabulut et al. [32], who presented the first multi-trace side-channel attack on the scheme, and McCarthy et al. [33], who introduced the first fault injection attack. FALCON's vulnerability to single-trace attacks has also been examined at the algorithmic level; for example, Guerreau et al. [27] analyzed leakage in the base sampling procedure, while Zhang et al. [28] proposed improvements to this approach. More recently, Lin et al. [29] identified side-channel leakage originating from the half-Gaussian sampler used during FALCON's signature generation. The vulnerability we identify is at a different subroutine and thus is orthogonal to these single-trace FALCON attacks.

We discovered a novel side-channel vulnerability in a previously unexplored stage of the FALCON signature scheme—key generation. The root cause of this vulnerability arises from the leaky negation operations, initially identified by Karabulut et al. [34] and later leveraged by Guerreau et al. [35]. However, our identification of how this vulnerability manifests in the targeted subroutine and our corresponding exploitation techniques for full secret-key recovery constitute the novel contributions of this work.

### C. Threat Model

We adopt the well-established threat model for single-trace side-channel attacks [29], [34], [35], assuming an adversary who has physical access to the target device and can measure its power consumption during cryptographic operations. The adversary is assumed to possess knowledge of the executing software, to approximate the timing of specific computations, and to intercept communication channels to capture exchanged public messages. Our attack uses a profiling phase. During this phase, the attacker can supply random inputs and analyze the software's power consumption behavior using known values. However, at runtime, the adversary is restricted to capturing a single power trace in an attempt to deduce the entire secret key.

We conduct our experiments on the STM32F417 development board, which features an ARM Cortex-M4 processor, one of the most widely used platforms in side-channel analysis research [27], [28], [29], [32], [34], [35]. Although evaluating the vulnerability across multiple hardware platforms is a reasonable direction for future work, we focus exclusively on this target device, following the approach taken in prior studies [13], [29], [34], [35].

The design and implementation of countermeasures are outside the scope of this paper, which focuses primarily on attack methodologies, following the approach taken in prior

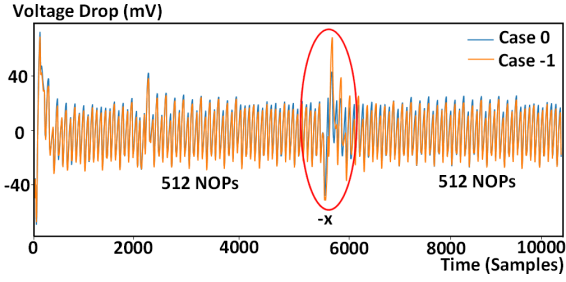


Fig. 2. Power traces of the leaky operation under varying inputs are presented. The blue trace depicts the power consumption during the leaky operation when the secret is assigned the value ‘0’, while the orange trace shows the power consumption when the secret is assigned ‘-1’. A distinct difference in power consumption between the two cases is observable.

work [32], [34]. Other attacks, such as fault injection [36], are likewise considered out of scope.

### III. UNDERLYING MECHANISM OF THE ATTACK

In this section, we first demonstrate how negating the result of the target bit-shift operation leaks intermediate secret variables. We also provide proof-of-concept demonstrations that confirm this vulnerability. Subsequently, we explain how these leaked variables enable the full recovery of FALCON’s secret polynomial.

#### A. The Operations That Leak

This section describes how negating the ‘63-bit right-shift’ operation can leak whether the result is ‘0’ or ‘-1’, and we present preliminary results to substantiate this claim.

For a 64-bit variable, the ‘63-bit right-shift’ operation (expressed as ‘ $x \gg 63$ ’ in software) shifts the most significant bit (MSB) to the least significant bit (LSB) and clears all other bits. This operation produces only two possible outcomes: ‘0’ or ‘1’. Negating the result yields either ‘0’ or ‘-1’, which are represented in 64-bit two’s complement as all 0s and all 1s, respectively. When the processor writes the result, the ‘-1’ case exhibits a Hamming weight (HW) of 64, while the ‘0’ case has an HW of 0. **Consequently, the ‘-1’ case results in higher power consumption compared to the ‘0’ case.** This leakage was first identified by Karabulut et al. [34]. Guerreau et al. [35] extended the analysis of this leakage. **However, this is the first study to identify this vulnerability in FALCON’s official software submission package.**

Figure 2 presents experimental results validating the observed power difference caused by this operation. The experiment involves performing a negation operation on a 64-bit value using the assembly code shown in Listing 1, aligning with the FALCON implementation. The `sbc.w` instruction is employed for two’s-complement negation (sign inversion) of the 64-bit value. The power consumption of the operation  $-(x)$  was measured, with  $x$  taking a value of either ‘1’ or ‘0’ (the result of  $x \gg 63$ ). Assembly NOP instructions were inserted around the operation to isolate it. The resulting traces were overlaid for comparison. The blue trace corresponds to  $x$  is ‘0’ and shows a peak voltage drop of 40 mV, whereas the orange trace, corresponding to  $x$  is ‘1’, exhibits a drop

Listing 1. Assembly instructions corresponding to  $-(x)$

```
1 negs    r2, r2;
2 sbc.w   r3, r3, r3, lsl #1;
3 strd    r2, r3, [r7, #24];
```

Listing 2. Gaussian sampling implementation from NIST submission package

```
1 mkgauss(RNG_CONTEXT *rng, unsigned logn){
2     ...
3     for (u = 0; u < g; u++) {
4         ...
5         r = get_rng_u64(rng);
6         neg = (uint32_t)(r >> 63);
7         r &= ~(uint64_t)1 << 63;
8         f = (uint32_t)((r -
9             gauss_1024_12289[0]) >> 63);
10        v = 0;
11        r = get_rng_u64(rng);
12        r &= ~(uint64_t)1 << 63;
13        for (k = 1; k < (sizeof
14            gauss_1024_12289)
15            / (sizeof gauss_1024_12289[0])
16            ; k++){
17            uint32_t t;
18            t = (uint32_t)((r -
19                gauss_1024_12289[k]) >> 63)
20                ^ 1;
21            v |= k & -(t & (f ^ 1));
22            f |= t;
23            v = (v ^ neg) + neg;
24            val += *(int32_t *)&v;
25        }
26        return val;
27    }
```

of 70 mV. Furthermore, the power spike is more pronounced in the ‘-1’ case compared to the ‘0’ case, demonstrating the vulnerability.

#### B. Only Two Variables Needed

This subsection explains how an adversary can recover FALCON’s base components  $f$  and  $g$  using only two intermediate variables within the discrete Gaussian sampling subroutine. Recall that the coefficients of  $f$  and  $g$  are generated using a discrete Gaussian sampling process. The reference C implementation of this process from the NIST submission package is outlined in Listing 2. This subroutine is called  $n$  times to generate values for a degree- $n$  polynomial forming the NTRU base components. The parameter  $n$  is the degree of the polynomial specified by the user. In the NIST submission package,  $n$  is configured as 512 or 1024. The implementation consists of an outer and an inner loop. The number of outer loop executions depends on the value of the variable `logn`. The number of inner loop executions depends on both the dimensions and contents of the predefined matrix `Gauss_1024_12289`. Static analysis reveals that the outer loop executes twice, and the inner loop executes 26 times per outer iteration. We subsequently audited the source code to identify how this behavior leads to side-channel leakage.

Within Listing 2, line 20 returns `val`, which holds the generated secret coefficient. To extract the generated secret coefficient, we trace the changes to this variable back in

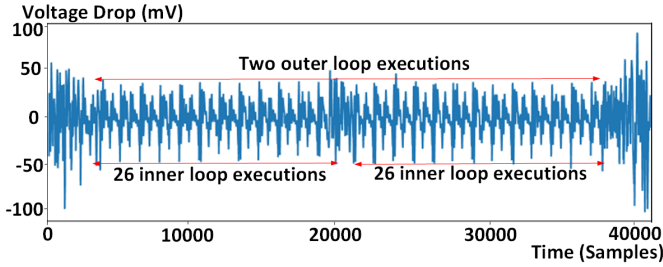


Fig. 3. The full power trace of the subroutine running on the target device (an STM32F417 development board containing an ARM Cortex-M4 CPU) is shown. The two outer-loop executions and the 26 inner-loop executions are clearly observable.

this subroutine. Line 19 performs an update that, despite involving different access patterns, simplifies to  $val = val + v$ . Since  $val$  is initialized to 0, the adversary only needs to infer the value of  $v$  for each outer loop to deduce the returned result. In Line 18,  $v$  is XORed with  $neg$  and then added to  $neg$ , implying that the adversary also needs to know  $neg$  to determine  $val$ . Therefore, we identify two critical points in the subroutine that enable the recovery of  $f$  and  $g$ . Specifically, line 16 reveals the value of  $v$ , and line 18 reveals  $neg$ , as highlighted in the listing.

Line 16 is our first attack point because the value of  $-(t \& (f \wedge 1))$  can only evaluate to ‘0’ or ‘-1’. The reason is that  $t$  and  $f$ —which are local variables distinct from the base component  $f$ —can only take ‘0’ or ‘1’ as a result of the ‘63-bit shift’ operation on lines 8 and 15, which retains only the most significant bit (MSB). The negation of  $t \& (f \wedge 1)$  thus simplifies to the negation of ‘0’ or ‘1’, respectively. The negation result, expressed in two’s complement, will be all zeros (when the result is ‘0’) or all ones (when the result is ‘-1’). **This will cause a significant difference in power consumption in the target device because the Hamming weight (HW) of these two results differs by 64.** Additionally, in line 16,  $k$  represents the iteration index of the inner loop and ranges from 1 to 26. An adversary can count recurring power peaks to infer the value of  $k$ .

Line 18 is our second attack point because  $neg$  can only take ‘0’ or ‘-1’. This is due to the ‘63-bit shift’ operation on line 6 of the subroutine. The negation of  $neg$  introduces a vulnerability analogous to the one described at line 16, due to the stark difference in Hamming weight (HW). An adversary can infer the value of  $neg$  by exploiting this vulnerability.

Since both points of exploitation are located within a loop, each loop iteration depends on the value of  $v$  generated during previous iterations. Therefore, the attack must achieve a high success rate to recover the secret correctly. Any incorrect inference of an intermediate value may result in a deviation from the correct output.

#### IV. EXPLOITING THE FOUND VULNERABILITY

This section presents the proposed attack strategy to recover the secret polynomials in FALCON. The power trace is analyzed to identify points of interest associated with information leakage. As noted previously, the attack begins with a profiling

stage, where the adversary has physical access to the target device and knowledge of the software implementation. In this stage, power measurements are collected under varying software inputs to construct a leakage profile to determine *when* leakage occurs. However, after profiling is complete, the adversary can recover the secret polynomial from a single power measurement.

##### A. Inspecting the Power Trace

Figure 3 shows the full power trace obtained when executing the discrete Gaussian sampling subroutine. Two outer loop executions and 26 inner loop executions per outer iteration are clearly distinguishable, reflecting the structure of the code in Listing 2. Each inner loop iteration produces a recurring pattern every 700 samples, while each outer loop spans 19000 samples. Because the discrete Gaussian sampling subroutine is executed  $n$  times during key generation, the resulting power traces exhibit a consistent and easily identifiable structure.

##### B. Pinpointing the Point of Interest

We use the following approach to apply correlation power analysis (CPA) to identify the point-of-interest (POI). The Pearson correlation coefficients are computed between the power measurements and a predetermined value set over time. First, we assign values to the variable  $r$  and adjust the values in the matrix `gauss_1024_12289` so that the value of  $r$  in the first and third inner loop executions can be manually controlled. We then collect 500,000 measurements, with  $-(t \& (f \wedge 1))$  set to ‘-1’ in the first inner loop and ‘0’ in the third inner loop for the first 250,000 measurements. For the second 250,000 measurements,  $-(t \& (f \wedge 1))$  is set to ‘0’ in the first inner loop and ‘-1’ in the third inner loop. The predetermined value set also contains 500,000 numbers. The first 250,000 values are set to 64, and the remaining 250,000 are set to 0, corresponding to the assigned values of  $-(t \& (f \wedge 1))$  in each case.

When  $-(t \& (f \wedge 1))$  is assigned the value ‘-1’, it will have a Hamming weight (HW) of 64. When  $-(t \& (f \wedge 1))$  is assigned the value ‘0’, it will have an HW of 0. Due to the power characteristics described in section III, this difference in Hamming weight is reflected in the device’s power consumption. This leads to a strong correlation between the device’s power consumption and the predetermined value set at the timestamp when  $-(t \& (f \wedge 1))$  occurs in the first and third inner loop executions. **Consequently, the Pearson correlation coefficient peaks at these two timestamps.**

Similarly, for the second leaky operation, we controlled the conditions under which  $neg$  is computed and applied Pearson correlation between the power traces and the predetermined value set. We focused on the power trace segment between the end of the last inner loop and the start of the second outer loop execution. The Pearson correlation coefficient peaks at the timestamp when  $neg$  is computed.

Figure 4 (top) illustrates the Pearson correlation observed over time for the first leaky operation,  $-(t \& (f \wedge 1))$ , and Figure 4 (bottom) shows the result for the second leaky

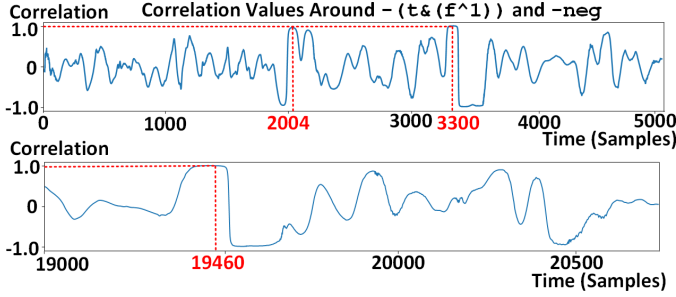


Fig. 4. Correlation power analysis (CPA) results: For the first attack point, the leaky operation occurs around timestamps 2004 and 3300, corresponding to correlation values of 0.996 and 0.977. For the second point, the leaky operation occurs around timestamp 19460, corresponding to a correlation value of 0.992.



Fig. 5. Equipment for trace acquisition includes devices for measuring the power consumption of the target system. A current probe is used to capture the power consumption, outputting a voltage proportional to the measured power. The data is then transmitted to an oscilloscope, which digitizes the output for further analysis.

operation,  $-neg$ . We found that the timestamps with the highest correlation occur around samples 2004 and 3300, corresponding to when  $-(t \& (f \wedge 1))$  was computed in the first and third inner loop executions. For  $-neg$ , we found that the highest correlation timestamps occur around sample 19460. Although a few other timestamps also exhibit elevated correlation, section V demonstrates that the chosen attack points are sufficient to extract the full key with high accuracy.

## V. EVALUATING THE SINGLE-TRACE VULNERABILITY

This section begins with a description of our measurement setup, followed by an analysis of the collected side-channel information and presentation of the results. We begin by demonstrating that the attack distinguishes intermediate value assignments through graphical illustrations. We then quantify our attack success rate using a theoretical model. Specifically,

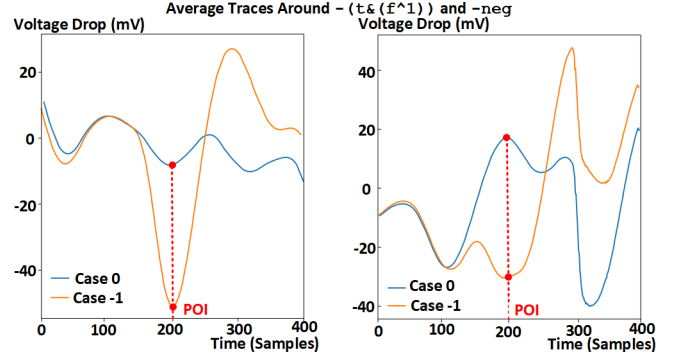


Fig. 6. The average power trace around the leaky operations for the two attack points is depicted. The left figure corresponds to the first attack point, while the right figure represents the second attack point. At both attack points, only two possible cases exist: ‘0’ and ‘-1’. The power consumption for the ‘-1’ case is observed to be higher than for the ‘0’ case.

we employ a univariate Gaussian template [37] with the selected point of interest (POI) to guarantee the success rate. Finally, we analyze the impact of the proposed attack on the security of FALCON.

### A. Measurement Setup

Figure 5 shows our measurement setup. The target device is an ARM Cortex-M4F CPU operating at 30 MHz, which is a canonical setting for side-channel testing in embedded applications according to previous publications [27], [28], [29], [34], [35]. We selected the lowest supported frequency to reduce noise, as lower frequencies typically yield cleaner power traces. Therefore, our measurements are not inherently limited by environmental noise. We utilized the submission package from the NIST reference software implementation. Measurements were captured using a PicoScope 3206D oscilloscope set to a sampling rate of 250 MHz. A Tektronix CT1 passive current probe was used, offering a bandwidth of 1–1,000 MHz at 3 dB. No external amplification was applied to enhance the measured signals.

All experiments presented in this work were performed on an STM32F417 series board equipped with an ARM Cortex-M4 processor, a platform widely adopted in side-channel research due to its accessibility and relevance. Although extending the analysis to additional hardware targets would be a logical next step, we limit our evaluation to a single device to maintain experimental focus and reproducibility. This decision is consistent with the methodology employed in several prior studies [13], [29], [34], [35].

### B. Attack Results

#### 1) Results on $-(t \& (f \wedge 1))$

The left panel in Figure 6 shows the average power trace of the 400 samples around the point-of-interest (POI) when attacking  $-(t \& (f \wedge 1))$ . The horizontal axis represents time in samples, while the vertical axis indicates power consumption. Sample point 200 marks the POI identified in section IV. The results indicate that the average power consumption for the ‘-1’ case is higher than for the ‘0’ case.

We then quantify the attack success rate by modeling the power distribution to derive a theoretical estimate. Since FALCON executes the targeted discrete Gaussian sampling step once to generate a single secret coefficient, we conduct a single-trace template attack.

We selected the point of maximum correlation as the POI to build our univariate Gaussian template, though multiple POIs could be chosen to enhance the success rate on noisier platforms. At this POI, we computed the mean  $\mu_i$  and variance of power  $v_i$ .

$$P_k = \sum_{j=0}^k \log \mathcal{N}(t_{j,s_i}, \mu_i, v_i) , \quad (4)$$

Using 500,000 measurements, we constructed the template by applying a normal probability density function (NPDF) at the POI. The computation incorporates the observed trace values  $t_{j,s_i}$ , along with the mean  $\mu_i$ , and variance  $v_i$  obtained during profiling. To mitigate precision issues caused by extreme NPDF values, we computed the sum of log-likelihoods under the normal distribution  $\mathcal{N}$ , as shown in Equation 4. The index of the matrix  $P_k$  with the highest value corresponds to the predicted coefficient.

The left panel of Figure 7 illustrates the Gaussian model derived from the data at the POI. The horizontal axis represents power consumption measured by voltage drop, while the vertical axis represents the probability density as derived from the template. The two bell-shaped curves correspond to the cases where the intermediate value is ‘-1’ or ‘0’. For this attack point, the two distributions are well separated, with overlapping area accounting for  $2.56 \times 10^{-9}\%$  of the total area under the two curves, indicating a success rate greater than 99.999999999%. The obtained Gaussian model was applied to 500,000 measurements to derive classification labels. A comparison with the ground truth shows a classification accuracy of 100% on real-world data.

## 2) Results on *-neg*

We followed the same template-building approach to evaluate the attack results on *-neg*. The right panel of Figure 6 illustrates the separation between cases ‘0’ and ‘-1’ cases, as observed in the average power traces. The horizontal axis represents time samples, while the vertical axis reflects power consumption. These average traces show a clear distinction between the two cases.

The right panel of Figure 7 illustrates the Gaussian model obtained from the first step of our attack. We selected the point of maximum correlation as the POI to build our univariate Gaussian template. The horizontal axis represents power consumption, and the vertical axis denotes the response probability from the constructed template. The two bell-shaped curves illustrate the two cases, which are well separated at this attack point. The overlap accounts for only  $2.55 \times 10^{-10}\%$  of the total area under the curves, indicating classification accuracy higher than 99.999999999%. The Gaussian model was applied to 500,000 measurements, and the resulting classifications matched the ground truth, yielding an accuracy of 100%.

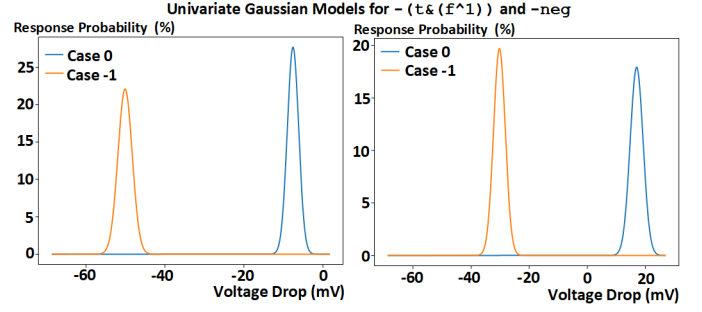


Fig. 7. The final results for the two attack points are presented. The left figure illustrates the results for the first attack point, while the right figure shows the results for the second attack point. We applied a univariate Gaussian model. The results are clearly separated in both cases, with no overlap, indicating clear and successful classification.

## C. The Impact on FALCON’s Security

Based on the generalized success rate for each secret value assignment described above, we analyze the impact on FALCON’s security scheme. Each discrete Gaussian sampling subroutine execution involves running the outer loop twice and running the inner loop 52 times. Our attack on  $-(t \& (f \wedge 1))$  accomplished 99.999999999% accuracy, while the attack on *-neg* accomplished 99.999999999% accuracy. Consequently, our attack for extracting one coefficient in FALCON is as follows:

$$(99.999999999\%)^{52} \times (99.999999999\%)^2 = 99.999999478\%$$

Based on the success rates described above, the overall success rate for recovering the secret polynomials  $f$  and  $g$ , in the FALCON-512 parameter set is as follows:

$$((99.999999478\%)^{512})^2 = 99.99994654\%$$

For FALCON-1024, the overall success rate for recovering the secret polynomials  $f$  and  $g$  is as follows:

$$((99.999999478\%)^{1024})^2 = 99.99989309\%$$

We assert that this vulnerability represents a significant compromise of the FALCON cryptographic scheme.

## VI. DISCUSSIONS

In this section, we briefly discuss potential defense methods and calibration parameters considered in our experimental setup. We also analyze the limitations of the proposed attack.

### A. Defense Methods

As in prior studies [13], [32], [34], this work demonstrates a practical single-trace attack to highlight real-world risks associated with vulnerabilities in FALCON and to promote awareness within the developer community. We offer a brief overview of potential countermeasures without emphasizing any specific approach. The design and implementation of these measures are considered outside the scope of this paper.

Defenses against single-trace side-channel vulnerabilities can be implemented at both the hardware and software levels. On the hardware side, constant-power designs aim to flatten the power profile of cryptographic operations, thereby preventing power traces from revealing data-dependent behavior. Such designs can be implemented through custom circuitry that ensures uniform power consumption. For example, a switched-capacitor power supply can be employed as a hardware-level countermeasure [38]. This technique charges a bank of capacitors during non-sensitive operations and uses the stored energy to power the device during sensitive computations—such as the discrete Gaussian sampling subroutine—thereby reducing leakage. As a result, the device draws minimal or constant power from the external supply during these critical periods, reducing observable leakage. While effective, these approaches often require significant changes to the hardware architecture and may incur performance and area overheads.

On the software side, a widely used class of countermeasures is known as hiding, which seeks to obscure the relationship between internal computations and side-channel leakage. This can be achieved by inserting dummy operations, introducing random delays, or reordering independent instructions, thus reducing the temporal correlation between power consumption and specific intermediate values [39].

#### *B. Applicability to Other Implementations*

The proposed attack also applies to other algorithms that negate the result of a 63-bit right shift applied to 64-bit intermediate variables containing secret data. The identified vulnerability exists in all of FALCON’s software implementations. This includes both the baseline and optimized versions provided in the NIST submission package, as the discrete Gaussian sampling subroutines remain unchanged across all implementations. Although the attack was demonstrated on FALCON-512, it applies equally to FALCON-1024, which employs the same sampling subroutine.

#### *C. Calibration Factors*

The platform’s noise level decreases as the device’s operating frequency is lowered. To minimize measurement noise, we configured the development board to operate at its minimum supported frequency of 30 MHz, in line with prior studies [20], [34], [40]. Earlier works have demonstrated single-trace side-channel attacks at even lower frequencies, such as 8 MHz in attacks on the NTT [19]. Analyzing higher frequencies may require more sophisticated power measurement equipment, additional probes for near-field electromagnetic leakage detection, or amplification and post-processing for noise reduction.

Although full key recovery demonstrations can be informative, our evaluation focuses on real-world recovery of intermediate coefficients at two identified leakage points. We then use theoretical modeling to guarantee the overall success rate of full key extraction. This follows a common practice in prior work [21], [34], [41], where full key recovery is omitted due to the repetitive nature of applying the template to power measurements and the limited analytical value it

provides. In the case of FALCON, fully recovering the secret key requires applying the same obtained template to power measurements at least 104 times, providing limited insights. More importantly, while strong performance on a specific test set may demonstrate success in an individual instance, only a theoretical model can provide generalizable guarantees for future measurement success and inference across varied conditions.

#### *D. Drawbacks of Our Attack*

Template attacks have well-known limitations and challenges, including processing time constraints. In our scenario, selecting a single POI resulted in a high success rate and required only a few minutes to construct the profiling templates. On devices with more complex power profiles, incorporating additional POIs could further improve attack effectiveness, though it would also incur higher computational overhead.

Our paper follows the same method used in prior demonstrations of single-device attacks [29], [34], [35]. For attacks to succeed on devices of different makes and models (also known as cross-device attacks), it is essential to develop device-specific power profiles that consider architectural features such as pipelining and out-of-order execution. Overcoming these challenges may require advanced machine-learning-based profiling techniques [42], [43] or reconstructing the profiling template on newly tested device types. It is important to recognize that the challenge of cross-device single-trace side-channel attacks on post-quantum cryptosystems remains an open problem. Our paper adopts the method used in prior demonstrations of single-device attacks [19], [21], [24], [44].

## VII. CONCLUSIONS

While lattice-based cryptography offers strong post-quantum security with relatively low computational overhead, it relies on specialized operations that have not been extensively scrutinized for side-channel leakage. In this work, we revealed a critical, new vulnerability in software implementations of FALCON, specifically related to the negation of a value obtained via a 63-bit right shift.

Our analysis demonstrated that FALCON’s discrete Gaussian sampling routine leaks intermediate value assignments, enabling full recovery of the secret key. The attack is validated on an off-the-shelf embedded device running both the reference and optimized implementations from FALCON’s NIST submission package, confirming its practicality. Notably, the uncovered vulnerability differs from known single-trace attacks and is inherently distinct from multi-trace approaches. Consequently, existing countermeasures designed for other types of leakage may be ineffective and needs to be re-evaluated in light of these findings and modified if needed. This paper, therefore, presents a concrete attack, with the primary goal of highlighting the specific risks of using FALCON’s unprotected reference software in key generation and informing the developers of this leakage.

## REFERENCES

- [1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.
- [3] P. L. Montgomery, "A survey of modern integer factorization algorithms," *CWI quarterly*, vol. 7, no. 4, pp. 337–366, 1994.
- [4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976, conference Name: IEEE Transactions on Information Theory. [Online]. Available: <https://ieeexplore.ieee.org/document/1055638>
- [5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.
- [6] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016, vol. 12.
- [7] V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, and S. Bai, "Crystals-dilithium," *Algorithm Specifications and Supporting Documentation*, 2020.
- [8] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2129–2146.
- [9] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon," *Post-Quantum Cryptography Project of NIST*, 2020.
- [10] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual international cryptology conference*. Springer, 1999, pp. 388–397.
- [11] P. Ravi, S. S. Roy, A. Chattopadhyay, and S. Bhasin, "Generic side-channel attacks on cca-secure lattice-based pke and kems," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 307–335, 2020.
- [12] A. A. Malik, E. Karabulut, A. Awad, and A. Aysu, "Enabling secure and efficient sharing of accelerators in expeditionary systems," *Journal of Hardware and Systems Security*, vol. 8, no. 2, pp. 94–112, 2024.
- [13] A. Kurian, A. Dubey, F. Yaman, and A. Aysu, "Tpuxtract: An exhaustive hyperparameter extraction framework," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2025, no. 1, pp. 78–103, 2025.
- [14] K. Ngo, E. Dubrova, Q. Guo, and T. Johansson, "A side-channel attack on a masked ind-cca secure saber kem implementation," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 676–707, 2021.
- [15] K. Sedghighadikolaei and A. A. Yavuz, "A comprehensive survey of threshold signatures: Nist standards, post-quantum cryptography, exotic techniques, and real-world applications," *arXiv preprint arXiv:2311.05514*, 2023.
- [16] J. Liu, T. Le, T. Ji, R. Yu, D. Farfurnik, G. Bryd, and D. Stancil, "The road to quantum internet: Progress in quantum network testbeds and major demonstrations," *Progress in Quantum Electronics*, 2024.
- [17] H. Li and T. J. Webster, "Trends in nanomedicine," in *Nanomedicine*. Elsevier, 2023, pp. 1–18.
- [18] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," in *International Conference on Cryptographic Hardware and Embedded Systems*, 2017, pp. 513–533.
- [19] P. Pessl and R. Primas, "More practical single-trace attacks on the number theoretic transform," in *International Conference on Cryptology and Information Security in Latin America*. Springer, 2019, pp. 130–149.
- [20] A. Aysu, Y. Tobah, M. Tiwari, A. Gerstlauer, and M. Orshansky, "Horizontal side-channel vulnerabilities of post-quantum key exchange protocols," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 81–88.
- [21] J. W. Bos, S. Friedberger, M. Martinoli, E. Oswald, and M. Stam, "Assessing the feasibility of single trace power analysis of frodo," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 216–234.
- [22] B.-Y. Sim, J. Kwon, J. Lee, I.-J. Kim, T. Lee, J. Han, H. Yoon, J. Cho, and D.-G. Han, "Single-trace attacks on the message encoding of lattice-based kems," *Cryptology ePrint Archive*, Report 2020/992, 2020, <https://eprint.iacr.org/2020/992>.
- [23] T. Rabas, J. Buček, and R. Lórencz, "Single-trace side-channel attacks on ntru implementation," *SN Computer Science*, vol. 5, p. 239, 2024.
- [24] S. Kim and S. Hong, "Single trace analysis on constant time cdt sampler and its countermeasure," *Applied Sciences*, vol. 8, no. 10, p. 1809, 2018.
- [25] K.-H. Choi, J. Han, and D.-G. Han, "Single trace analysis of visible vs. invisible leakage for comparison-operation-based cdt sampling," *Electronics*, vol. 13, no. 23, p. 4681, 2024.
- [26] S. Jendral, K. Ngo, R. Wang, and E. Dubrova, "Breaking sca-protected crystals-kyber with a single trace," in *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2024, pp. 70–73.
- [27] M. Guerreau, A. Martinelli, T. Ricosset, and M. Rossi, "The hidden parallelepiped is back again: Power analysis attacks on falcon," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 141–164, 2022.
- [28] S. Zhang, X. Lin, Y. Yu, and W. Wang, "Improved power analysis attacks on falcon," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2023, pp. 565–595.
- [29] X. Lin, S. Zhang, Y. Yu, W. Wang, Q. You, X. Xu, and X. Wang, "Thorough power analysis on falcon gaussian samplers and practical countermeasure," *Cryptology ePrint Archive*, 2025.
- [30] S. Marzougui, N. Wisiol, P. Gersch, J. Krämer, and J.-P. Seifert, "Machine-learning side-channel attacks on the galactics constant-time implementation of bliss," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–11.
- [31] M. Schönauer, "On the physical security of falcon," Ph.D. dissertation, Technische Universität Wien, 2024.
- [32] E. Karabulut and A. Aysu, "Falcon down: Breaking falcon post-quantum signature scheme through side-channel attacks," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 691–696.
- [33] S. McCarthy, J. Howe, N. Smyth, S. Brannigan, and M. O'Neill, "Bearz attack falcon: implementation attacks with countermeasures on the falcon signature scheme," *Cryptology ePrint Archive*, 2019.
- [34] E. Karabulut, E. Alkim, and A. Aysu, "Single-trace side-channel attacks on  $\omega$ -small polynomial sampling: with applications to ntru, ntru prime, and crystals-dilithium," in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2021, pp. 35–45.
- [35] M. Guerreau and M. Rossi, "A not so discrete sampler: Power analysis attacks on hawk signature scheme," *Cryptology ePrint Archive*, 2024.
- [36] A. A. Malik, H. Mihir, and A. Aysu, "Craft: Characterizing and root-causing fault injection threats at pre-silicon," *arXiv preprint arXiv:2503.03877*, 2025.
- [37] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 13–28.
- [38] S. Sen and A. Ghosh, "Circuit-level techniques for side-channel attack resilience: A tutorial," *IEEE Solid-State Circuits Magazine*, vol. 16, no. 4, pp. 96–108, 2024.
- [39] M. Brisfors, M. Moraitis, and E. Dubrova, "Side-channel attack countermeasures based on clock randomization have a fundamental flaw," *Cryptology ePrint Archive*, Paper 2022/1416, 2022. [Online]. Available: <https://eprint.iacr.org/2022/1416>
- [40] D. Owens, R. El Khatib, M. Bisheh-Niasar, R. Azarderakhsh, and M. M. Kermani, "Efficient and side-channel resistant ed25519 on arm cortex-m4," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.
- [41] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 13–28.
- [42] P. Kashyap, F. Aydin, S. Potluri, P. Franzon, and A. Aysu, "2deep: Enhancing side-channel attacks on lattice-based key-exchange via 2d deep learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2020.
- [43] M. Abdelkhalik, J. Qiu, M. Hernandez, A. Bozkurt, and E. Lobaton, "Investigating the relationship between cough detection and sampling frequency for wearable devices," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 7103–7107.
- [44] P. Ravi, M. P. Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin, "Side-channel assisted existential forgery attack on dilithium-a nist pqc candidate," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 821, 2018.