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

Anonymous Submission

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 operation 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 99.9999999478% 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.

Keywords: FALCON · Side channel · Post-quantum cryptography · NTRU · Key Generation · Lattice-based cryptography · Digital signature schemes

1 Introduction

Widely adopted encryption schemes such as RSA [RSA78] and elliptic curve cryptosystems [Kob87] rely on hard mathematical problems, including integer factorization [Mon94] and the discrete logarithm problem [DH76], which are traditionally regarded as computationally intractable for classical computers. However, quantum algorithms, particularly the Shor's algorithm [Sho99], offer exponential speedups for solving these problems, rendering classical cryptographic assumptions obsolete in the presence of a sufficiently powerful quantum computer. As a result, advances in quantum computing pose a significant threat to conventional public-key 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 [CCJ+16]. This effort aims to ensure long-term security in the face of rapidly advancing quantum computing technologies. As of now, NIST has selected three digital signature schemes for standardization: CRYSTALS-Dilithium [LDK+20], SPHINCS+[BHK+19], and FALCON[PFH+20]. Each of these schemes represents a different design philosophy and security foundation, providing diversity in cryptographic assumptions. FALCON, in particular, was selected for its highly compact signature size and efficient verification, making it especially attractive for use in embedded systems, mobile devices, and other bandwidth or memory-constrained environments. It is based on lattice cryptography

and leverages the Gentry-Peikert-Vaikuntanathan (GPV) framework with a fast Fourier sampling algorithm, enabling it to achieve high performance without sacrificing security.

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 [KJJ99]. An attacker can carry out such attacks using only a few side-channel measurements from the physical device [RRCB20, MKAA24, KDYA25]. 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 [NDGJ21]. Moreover, they often target critical and less-defended subroutines-such as key generation-which produce ephemeral secrets that are not reused, thereby reducing the opportunity for defensive monitoring or repeated validation. 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 [SY23, LLJ+24, LW23], 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. Example components include the number-theoretic transform (NTT) [PPM17a, PP19], polynomial multiplication [ATT+18a, BFM+18], message encoding/decoding [SKL+20, RBL24], cumulative distribution table (CDT) sampling [KH18, CHH24], and the Fujisaki-Okamoto transform [JNWD24]. 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. We provide a comprehensive comparison with the state-of-the-art attacks on FALCON in Subsection 2.2.

In this paper, we present a *novel* single-trace side-channel vulnerability in FALCON that is distinct from previously reported attacks [CHH24, GMRR22, ZLYW23, LZY+25, MWG+22, Sch24]. 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 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. 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 that is a widely used platform for side-channel evaluation. The microcontroller executes the reference and optimized implementations 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 99.9999999478% and a full key recovery rate higher than 99.99989309%. The attack is effective across

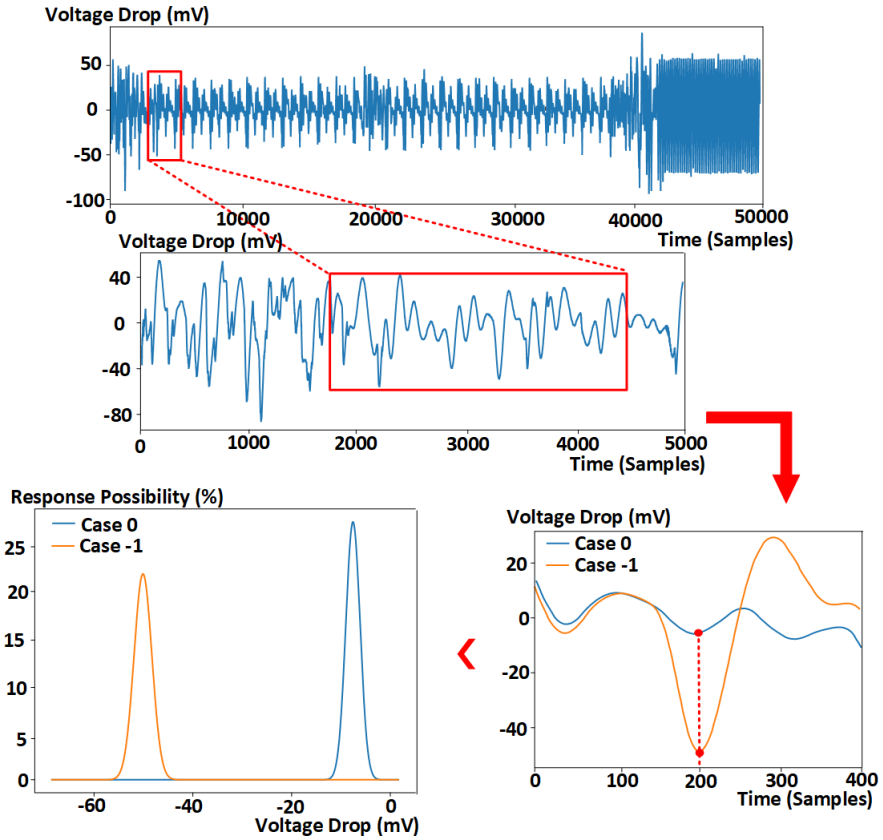

Figure 1: Visual Demonstration of the Vulnerability: The top figure illustrates the power consumption profile of the device during the key generation process. The middle figure provides a zoomed-in view of the power consumption during a specific vulnerable segment in the code. The bottom right figure 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.

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.

The remainder of the paper is organized as follows. Section 2 introduces essential background on FALCON and positions our work relative to the current state of the art, highlighting the gap our attack fills. Section 3 presents our foundational analysis, in which we identify a critical leaky operation within FALCON's discrete Gaussian sampling subroutine and show how it can be systematically exploited to recover the secret key with high reliability. Section 4 details our attack methodology, which combines fine-grained power analysis with profiling techniques to enable single-trace key recovery. Section 5 reports comprehensive experimental results, demonstrating not only the effectiveness of the attack but also how its success rate scales with the number of traces. In Section 6, we examine calibration factors, practical deployment challenges, and broader implications of our findings for the post-quantum cryptographic ecosystem. Finally, Section 7 summarizes

our key contributions and underscores the urgent need for robust countermeasures as post-quantum standards move toward real-world deployment.

2 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.

2.1 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 shortest integer solution (SIS) 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

$f, g, F, G \leftarrow \text{NTRUGen}(\phi, q)$ ▷ Solving the NTRU equation
$\mathbf{B} \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$
$\hat{\mathbf{B}} \leftarrow \text{FFT}(\mathbf{B})$ ▷ Compute the FFT for each of the 4 components $\{g, -f, G, -F\}$
$\mathbf{G} \leftarrow \hat{\mathbf{B}} \times \hat{\mathbf{B}}^{\star}$
$\mathbf{T} \leftarrow \text{ffLDL}^{\star}(\mathbf{G})$ ▷ Computing the LDL* tree
for each leaf of $\mathbf{T}$ **do** ▷ Loop over the leaves of $\mathbf{T}$
  leaf.value $\leftarrow \sigma / \sqrt{\text{leaf.value}}$ ▷ Normalization step
sk $\leftarrow (\hat{\mathbf{B}}, \mathbf{T})$
$h \leftarrow g f^{-1} \mod q$
pk $\leftarrow h$
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, f and g, 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 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 3.

Algorithm 2 NTRUGen(ϕ, q)

Require: A monic polynomial $\phi \in \mathbb{Z}[x]$ of degree n, a modulus q
Ensure: Polynomials f, g, F, G
1: $\sigma_{(f,g)} \leftarrow 1.17\sqrt{q}/2n$ ▷ $\sigma_{(f,g)}$ 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: $f \leftarrow \sum_i f_i x^i$ ▷ $f \in \mathbb{Z}[x]/(\phi)$
7: $g \leftarrow \sum_i g_i x^i$ ▷ $g \in \mathbb{Z}[x]/(\phi)$
...
8: $F, G \leftarrow \text{NTRUSolve}_n, q(f,g)$ ▷ Computing F, G such that $fG - gF = q \mod \phi$
9: **if** $(F, G) = 1$ **then**
10: **restart**
11: **end if**
12: **return** f, g, F, G

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:

$$f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots + f_{n-1} x^{n-1} \tag{1}$$

$$g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \cdots + g_{n-1} x^{n-1} \tag{2}$$

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.

$$fG - gF = q \mod \phi \tag{3}$$

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 converting the polynomials into floating-point representations, 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.

2.2 Comparison to Previous Attacks on FALCON

Previous studies investigating implementation attacks on FALCON include the work of Karabulut et al. [KA21], who presented the first multi-trace side-channel attack on the

scheme, and McCarthy et al. [MHS+19], 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. [GMRR22] analyzed leakage in the base sampling procedure, while Zhang et al. [ZLYW23] proposed improvements to this approach. More recently, Lin et al. [LZY+25] identified side-channel leakage originating from the half-Gaussian sampler used during FALCON's signature generation. The Gaussian sampler has recently been shown to be vulnerable due to floating-point inaccuracies, which can be exploited in fault attacks [LTYZ25]. Moreover, successful side-channel attacks have been demonstrated on masked implementations of FALCON, including variants like Mitaka [SD25]. 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. [KAA21] and later leveraged by Guerreau et al. [GR24]. 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.

Although prior research has predominantly focused on side-channel attacks targeting the signature generation process, the key generation algorithm, even though it is executed only once per key pair, remains vulnerable to single-trace attacks according to recent publications [PK25, BCM+25, YCY+24, PPM17b, BJL+14], making the exploration of vulnerabilities an active research topic. Identifying and mitigating such vulnerabilities during key generation thus represents a critical and ongoing challenge in securing cryptographic implementations against real-world adversaries.

2.3 Threat Model

We adopt the well-established threat model for single-trace side-channel attacks[KAA21, KA21, GMRR22, ATT+18b, BCM+25, LZY+25, BJL+14, PPM17b, MKK+23], assuming an adversary who has physical access to the target device or a device that is of the same make and model as the target device. The adversary can measure the device's 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.

The threat model we employ is practical in realistic deployment scenarios because the key generation (KeyGen) algorithm is typically executed only once per device or session. This limited execution frequency often results in its deprioritization or complete omission in the design of side-channel countermeasures. **However, this infrequent execution does not mitigate its susceptibility to single-trace side-channel attacks.** Our experimental results demonstrate that even a single power trace observed during KeyGen can leak sufficient information to recover the secret key, underscoring the urgent need for robust, dedicated countermeasures during the key generation phase.

Our experiments are conducted on the STM32F417 development board, which integrates an ARM Cortex-M4 processor-a widely adopted platform in side-channel analysis research, as evidenced by its frequent use in prior TCHES and related publications.[GMRR22, HCM24, RPJ+24, CC24, GR24, QWY+25, KA21, KAA21]. 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 [LZY+25, KDYA25, KAA21, GR24].

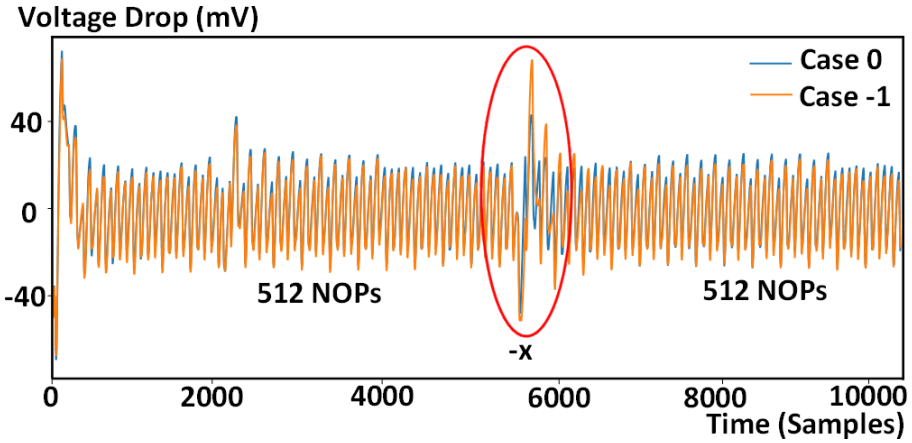

Figure 2: Power traces of the leaky operation under varying inputs are presented. The blue trace depicts the power consumption during the leaky operation when the secret is assigned the value '0', while the orange trace shows the power consumption when the secret is assigned '-1'. A distinct difference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

The design and implementation of countermeasures are outside the scope of this paper. We only provide a brief overview of possible countermeasures in Section 6 and we focus primarily on attack methodologies, following the approach taken in prior TCHES publications and publications at other venues [DG25, AHC+25, RPJ+24, KA21, KAA21]. Other attacks, such as fault injection [MMA25], are likewise considered out of scope.

3 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.

3.1 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. [KAA21]. Guerreau et al. [GR24] 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

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];
```

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 of 70 mV. Furthermore, the power spike is more pronounced in the '-1' case compared to the '0' case, demonstrating the vulnerability. We inserted `NOP` instructions to isolate the vulnerable operations as we try to identify the vulnerability and build our attack steps. However, the ultimate target implementation used in our attack was an unaltered version of FALCON submission package, without any inserted `NOP` instructions.

3.2 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`, which is set to 9 for FALCON-512. 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 32 returns `val`, which holds the generated secret coefficient. To extract the generated secret coefficient, we trace the changes to this variable back in this subroutine. Line 30 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 29, `v` is XORed with `-neg` and then added to `neg`, implying that the adversary also needs to know `neg` to determine `val`. We identify two critical points in the subroutine that enable the recovery of f and g. Specifically, line 27 reveals the value of `v`, and line 29 reveals `neg`, as highlighted in the listing.

Line 27 is our first attack point because the value of `-(t & (f ^ 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 18 and 26, which retains only the most significant bit (MSB). The negation of `t & (f ^ 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`.

Listing 2: Gaussian sampling implementation from NIST submission package

```
1  mkgauss(RNG_CONTEXT *rng, unsigned logn){
2      unsigned u, g;
3      int val;
4
5      g = 1U << (10 - logn);
6      val = 0;
7      for (u = 0; u < g; u ++)
8      {     /*
9            * Each iteration generates one value with the
10           * Gaussian distribution for N = 512 or N = 1024.
11           */
12         uint64_t r;
13         uint32_t f, v, k, neg;
14
15         r = get_rng_u64(rng);
16         neg = (uint32_t)(r >> 63);
17         r &= ~((uint64_t)1 << 63);
18         f = (uint32_t)((r - gauss_1024_12289[0]) >> 63);
19         v = 0;
20         r = get_rng_u64(rng);
21         r &= ~((uint64_t)1 << 63);
22         for (k = 1; k < (sizeof gauss_1024_12289)
23             / (sizeof gauss_1024_12289[0]); k ++)
24             {
25             uint32_t t;
26             t = (uint32_t)((r - gauss_1024_12289[k]) >> 63) ^ 1;
27              v |= k & -(t & (f ^1));
28             f |= t;}
29          v = (v ^-neg) + neg;
30         val += *(int32_t *)&v;
31         }
32     return val;
33     }
```

Line 29 is our second attack point because `-neg` can only take '0' or '-1'. This is due to the '63-bit shift' operation on line 16 of the subroutine. The negation of `neg` introduces a vulnerability analogous to the one described at line 27, 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.

4 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, profiling traces are collected under varying software inputs

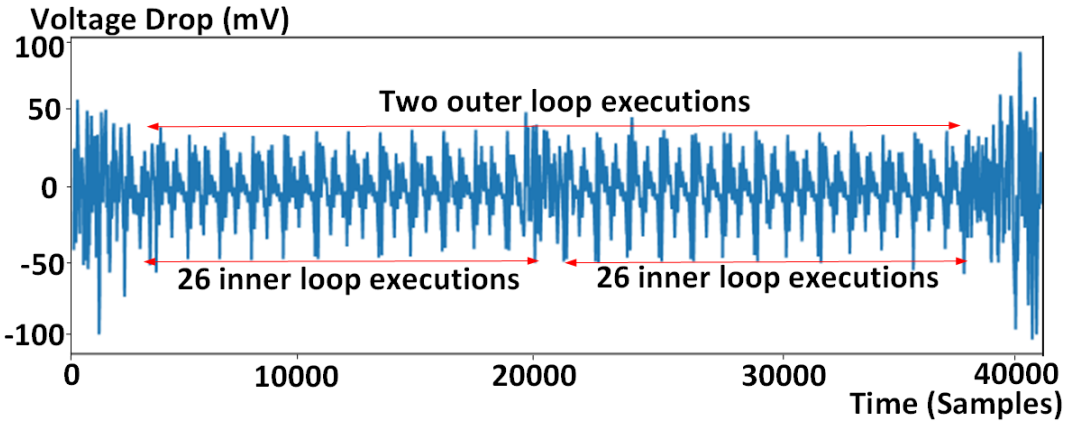

Figure 3: The full power trace of the subroutine running on the target device (an STM32F417 development board containing an ARM Cortex-M4 CPU) is shown. The two outer-loop executions and the 26 inner-loop executions are clearly observable.

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.

4.1 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 either 512 or 1024 times during key generation, the resulting power traces exhibit a consistent and easily identifiable structure.

4.2 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 profiling traces 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 `-(t & (f ^ 1))` in the first and third inner loop executions can be manually controlled. We then collect 500,000 traces, with `-(t & (f ^ 1))` set to '-1' in the first inner loop and '0' in the third inner loop for the first 250,000 traces. For the second 250,000 traces, `-(t & (f ^ 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 ^ 1))` in each case.

When `-(t & (f ^ 1))` is assigned the value '-1', it will have a Hamming weight (HW) of 64. When `-(t & (f ^ 1))` is assigned the value '0', it will have an HW of 0. Due to the power characteristics described in Section 3, this difference in Hamming weight is reflected in the power consumption of the device. This leads to a strong correlation between the device's power consumption and the predetermined value set at the timestamp when `-(t & (f ^ 1))` occurs in the first and third inner loop executions. **Consequently, the Pearson correlation coefficient peaks at these two timestamps.**

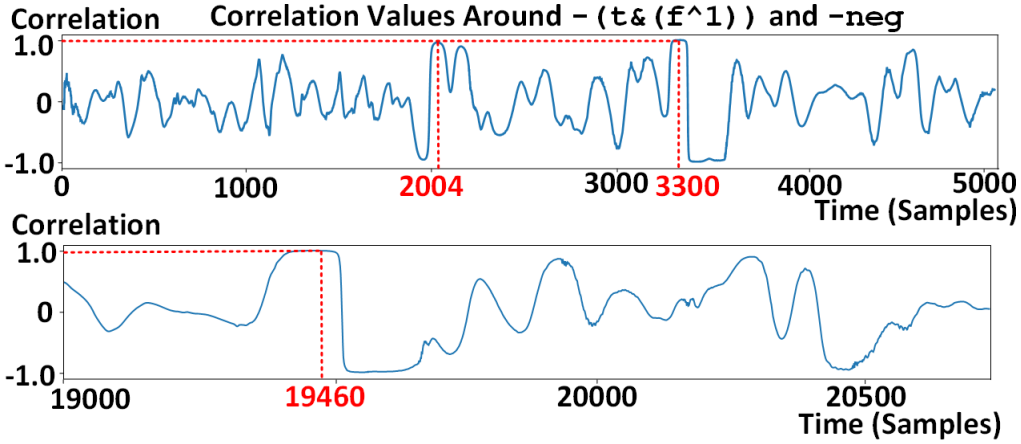

Figure 4: Correlation power analysis (CPA) results: For the first attack point, the leaky operation occurs around timestamps 2004 and 3300, corresponding to correlation values of 0.9997 and 0.9997. For the second point, the leaky operation occurs around timestamp 19460, corresponding to a correlation value of 0.9988.

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 around the end of the first outer 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 ^ 1))`, and Figure 4 (bottom) shows the result for the second leaky operation, `-neg`. We found that the timestamps with the highest correlation occur around samples 2004 and 3300, corresponding to when `-(t & (f ^ 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 5 demonstrates that the chosen attack points are sufficient to extract the full key with high accuracy. While the results are obtained from 500,000 traces, Subsection 5.3 shows that a consistently high correlation at the POI can be achieved using as few as 10 traces. To ensure the generalizability of our experimental results, we intentionally employ a larger number of traces during the evaluation phase.

5 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, we employ a univariate Gaussian template [CRR02a] with the selected point of interest (POI) to guarantee the success rate. To assess the profiling requirements, we examine how the attack performance evolves with varying numbers of available profiling traces. Finally, we analyze the impact of the proposed attack on the security of FALCON.

5.1 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

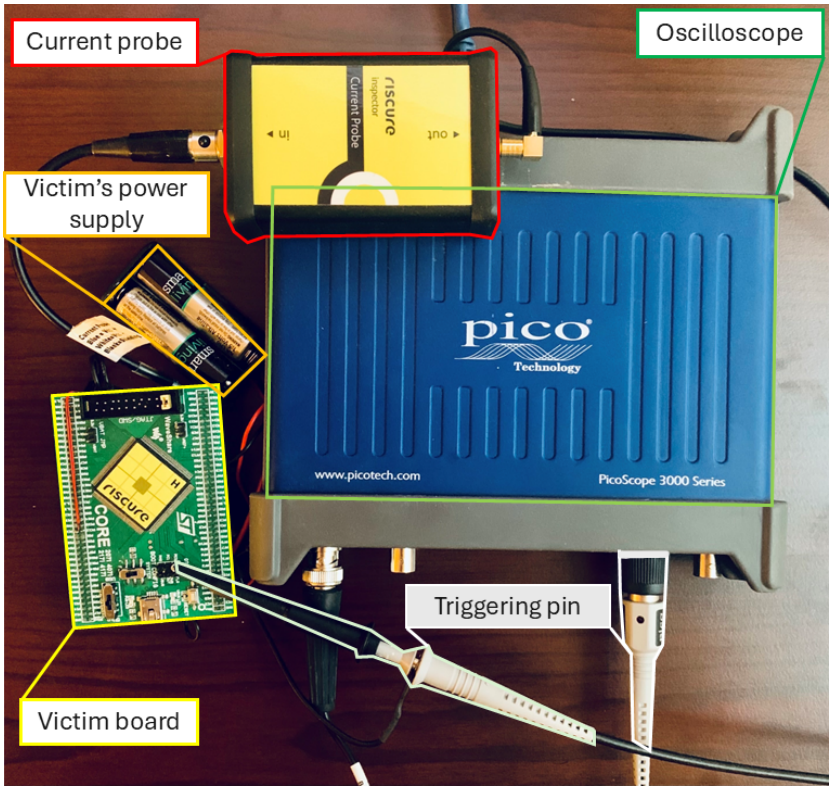

Figure 5: Equipment for trace acquisition includes devices for measuring the power consumption of the target system. A current probe is used to capture the power consumption, outputting a voltage proportional to the measured power. The data is then transmitted to an oscilloscope, which digitizes the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

applications according to previous publications [GMRR22, HCM24, RPJ+24, CC24, GR24, QWY+25, KA21, KAA21, ZLYW23, LZY+25]. 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.

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 [KDYA25, LZY+25, KAA21, GR24].

5.2 Attack Results

5.2.1 Results on `-(t & (f ^ 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 ^ 1))`. The horizontal axis represents time in samples, while the vertical axis indicates power consumption. Sample point 200 marks the POI identified in Section 4. The results indicate that the average power consumption for the '-1' case is higher than for the '0' case.

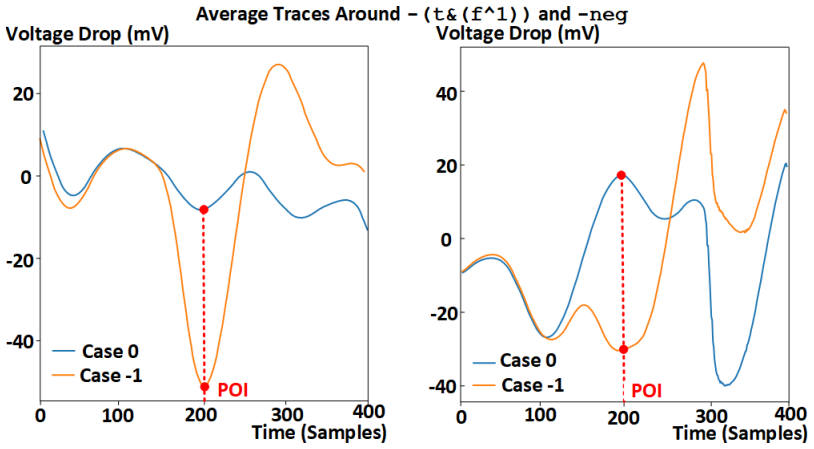

Figure 6: The average power trace around the leaky operations for the two attack points is depicted. The left figure corresponds to the first attack point, while the right figure represents the second attack point. At both attack points, only two possible cases exist: '0' and '-1'. The power consumption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

We 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 μ_i and variance of power v_i.

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

Using 500,000 profiling traces, 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 μ_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 an error rate of 10^{-9} and a success rate greater than 99.999999999%.

The obtained Gaussian model was applied to 500,000 real-world measurements to derive classification labels. A comparison with the ground truth shows a classification accuracy of 100% on real-world data. Although the results are based on 500,000 traces, Subsection 5.3 shows that a template can be built using as few as 10 traces. We intentionally use more traces during testing to ensure the generalizability of our results.

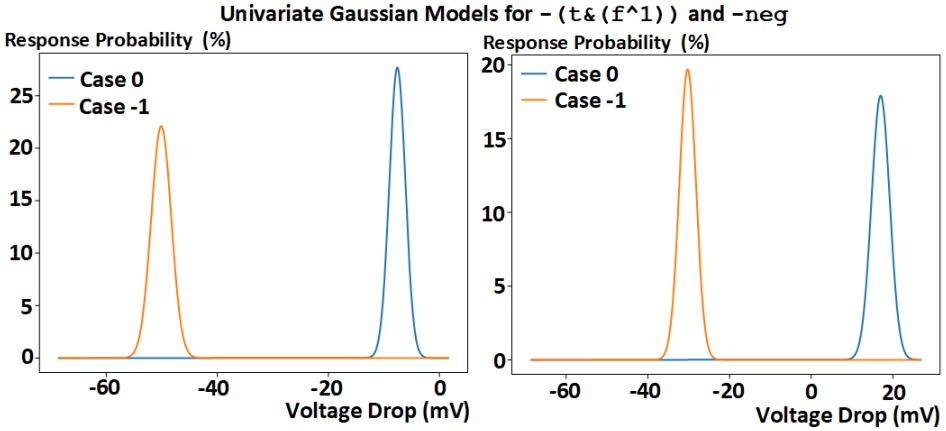

Figure 7: The final results for the two attack points are presented. The left figure illustrates the results for the first attack point, while the right figure shows the results for the second attack point. We applied a univariate Gaussian model. The results are clearly separated in both cases, with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

5.2.2 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 an error rate of 10^{-10} and classification accuracy higher than 99.9999999999%.

The Gaussian model was applied to 500,000 real-world measurements, and the resulting classifications matched the ground truth, yielding an accuracy of 100%. Similar to the previous case, we use 500,000 profiling traces to provide stronger empirical support for this attack. Subsection 5.3 shows that a successful template can be obtained using as few as 10 traces. We intentionally use more traces to ensure the generalizability of our results.

5.3 The Evolution of Attack Results with Number of Traces

As outlined in earlier sections, our attack begins with a profiling stage, during which the adversary constructs a power consumption profile of the target device. The number of power traces used in this stage can influence the practicality of the overall attack. While the testing results presented thus far are based on 500,000 power traces, this subsection analyzes how the effectiveness of the attack evolves as the number of traces changes. Our attack methodology consists of two primary steps: (1) identifying the leakage point through correlation power analysis (CPA), and (2) constructing a univariate Gaussian template based on two distinct groups of profiling power traces. The second step, which involves statistical modeling, can be intuitively interpreted by observing the separation in the average power consumption between the two groups. To this end, we analyze and visualize how both the correlation coefficient (used to identify leakage) and the difference in average

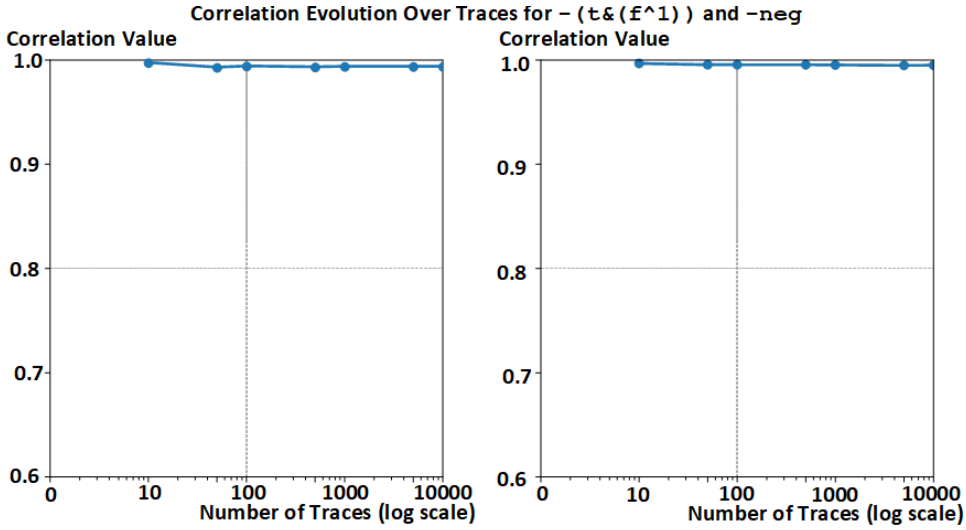

Figure 8: Evolution of the correlation value at the point of interest (POI) as the number of traces increases. The horizontal axis represents the number of traces considered (on a logarithmic scale), while the vertical axis shows the computed correlation values. The left panel corresponds to the attack point `-(t & (f ^1))`, and the right panel corresponds to `-neg`. A strong correlation, approaching 1.0, is observed with as few as 10 traces, indicating early leakage detectability at both POIs.

power levels (used to quantify distinguishability) change with varying numbers of traces. This analysis offers clearer insight into the trace requirements for successful profiling needs that leads to the ultimate attack execution with a single trace.

5.3.1 Evolution of Correlation Power Analysis Results w.r.t. Profiling Measurements

Figure 8 illustrates the evolution of correlation values as a function of the number of power traces used during the attack, evaluated at two distinct points of interest:`-(t & (f ^ 1))` and `-neg`. The horizontal axis is plotted on a logarithmic scale to show the progression from a small number to a large number of traces, while the vertical axis represents the corresponding correlation values computed between the fixed value set and the measured power traces. We initiate our evaluation using only 10 traces and observe that a stable point of interest (POI) emerges with as few as 10 traces. As the number of traces increases incrementally to 10,000, we track how the correlation values evolve at both POIs.

The left panel of the figure shows the correlation evolution for `-(t & (f ^ 1))`, while the right panel corresponds to `-neg`. Although the two panels look similar, we observe slightly different correlation values at the two attack points. For the attack point `-(t & (f ^ 1))`, we observe a correlation value of 0.9997 when using as few as 10 traces. The correlation value remains consistently high, approaching 1.0. The correlation for `-neg` yields 0.9988 with just 10 traces and remains stable as the number of traces increases, indicating highly distinguishable leakage at both points. These results demonstrate that a reliable leakage point can be identified using as few as 10 power traces.

5.3.2 Evolution of Average Power Value Separation w.r.t. Profiling Measurements

Figure 9 presents the evolution of average power differences at the POI corresponding to two attack points, `-(t & (f ^ 1))` and `-neg`, as a function of the number of power traces used. The left panel illustrates the results for `-(t & (f ^ 1))`, while the right

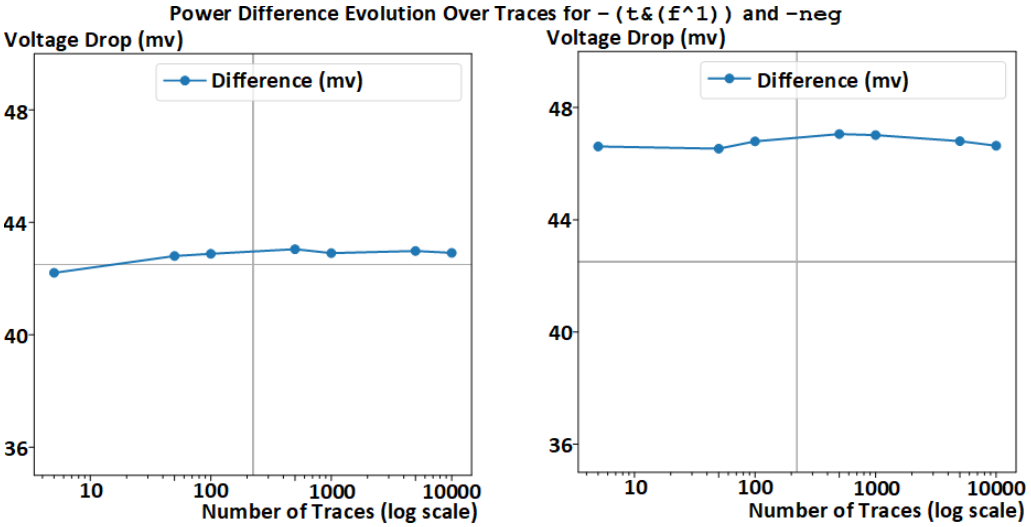

Figure 9: Evolution of the average power consumption difference between the two cases as the number of traces increases. The horizontal axis represents the number of traces considered (logarithmic scale), while the vertical axis shows the computed difference in average power consumption. The left panel depicts the difference at the attack point `-(t & (f ^ 1))`, while the right panel shows the difference at the attack point `-neg`. A clear and consistent separation is visible with as few as 5 traces.

panel shows the results for `-neg`. Although a small fluctuation is observed in the right panel, this variation is negligible and does not affect the overall analysis, as it occurs at a very small scale on the vertical axis. In both panels, the horizontal axis is plotted on a logarithmic scale to capture the progression across increasing trace counts, and the vertical axis represents the absolute difference in average power between the two cases.

Our analysis reveals that a clear and stable separation in average power values emerges with as few as 5 traces at both attack points. As the number of traces increases up to 10,000, this separation remains consistent, indicating that the observed leakage is robust and highly distinguishable. These results imply that an adversary can reliably classify new power measurements using only a minimal number of traces, as few as five, even based solely on average power. This underscores the vulnerability of the implementation, as it enables rapid and accurate identification of secret-dependent behavior with very limited profiling effort.

Our attack only need one power measurement trace to succeed in the attack phase. In the profiling stage, we significantly reduce the number of traces needed. In contrast to existing attacks on FALCON, which require a substantial amount of profiling data (e.g., 5,000 traces in [GMRR22], 220,000 traces in [ZLYW23], 6,500 traces in [LZY+25], and 10,000 traces in [GR24]), our attack significantly reduces the required profiling effort to as few as 10 traces. Furthermore, once the profiling phase is complete, the adversary needs only a single side-channel trace during the attack phase to successfully recover the secret key. This demonstrates the practicality and efficiency of our approach in exploiting single-trace leakage during key generation.

5.4 Cross-Device Attacks

We conducted the same set of experiments on two distinct STM32F417 development boards, each featuring an ARM Cortex-M4 microcontroller. The results were consistent across both platforms, closely aligned with the findings presented in the previous parts

of Section 5, with only negligible variations observed. As the outcomes are effectively identical, we omit redundant figures for clarity. This consistency across different hardware instances further validates the robustness and reproducibility of our methodology.

5.5 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. To recover all secret coefficients within a key generation session, the attacker must succeed at each individual attack point. Accordingly, the overall success rate is computed as the product of the individual success probabilities. Specifically, our attack on `-(t & (f ^ 1))` accomplished 99.999999999% accuracy (an error rate of 10^{-9}), while the attack on `-neg` accomplished 99.9999999999% accuracy (an error rate of 10^{-10}). Consequently, our attack for extracting one coefficient in FALCON is as follows:

$$(99.999999999\%)^{52} \times (99.9999999999\%)^{2} = 99.9999999478\%$$

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:

$$((99.9999999478\%)^{512})^{2} = 99.99994654\%$$

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

$$((99.9999999478\%)^{1024})^{2} = 99.99989309\%$$

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

Following established methodology, we assess the impact of the attack on FALCON's security level without demonstrating a full key recovery, in line with prior publications to maintain clarity and focus [GR24, GMRR22, KAA21, PPM17a]. Subsection 6.3 offers a detailed explanation and further insights into the rationale behind our approach.

6 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.

6.1 Potential Countermeasures

As in prior studies [DG25, AHC+25, RPJ+24, KDYA25, KA21, KAA21], this work demonstrates a practical side-channel 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 [TV04, Pla08, MRL14]. For example,

a switched-capacitor power supply can be employed as a hardware-level countermeasure [SG24]. 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. Therefore, the leakage is reduced. 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 [BMD22]. Another widely used method is shuffling. Shuffling aims to obscure the order of operations in cryptographic algorithms. This is done by randomly permuting the sequence in which secret dependent operations are executed. Shuffling introduces temporal noise into power or timing traces. It disrupts an attacker's ability to align and correlate traces across executions, thereby hindering statistical leakage exploitation [Pes16, CMM17].

6.2 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. For FALCON-1024, we obtained a final total success rate of 99.99989309%, which is sufficiently high to demonstrate the practical feasibility and reliability of the attack in real-world scenarios.

6.3 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 [ATT+18a, KAA21, OEKBN+24]. Earlier works have demonstrated single-trace side-channel attacks at even lower frequencies, such as 8 MHz in an attack on the NTT [PP19], 5 MHz in an attack on key encapsulation mechanisms [GJJ22], and 29.48 MHz for generic side-channel security evaluations [RPC22], as in previous TCHES and Latincrypt publications. 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 of prior works in TCHES [GR24, GMRR22, KAA21, PPM17a, CRR02b], 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 a coefficient of the secret key requires applying the same obtained template to power measurements at least 106 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 inference success** across varied conditions.

6.4 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 side-channel attacks [PK25, KAA21, KDYA25, LZY+25]. For attacks to succeed on devices of different makes and models (also known as heterogeneous 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 [KAP+20, AQH+21] 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 [PP19, BFM+18, KH18, RJH+18].

7 Conclusion

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

[AHC+25] Pierre Ayoub, Aurélien Hernandez, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. Phasesca: Exploiting phase-modulated emanations in side channels. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):392–419, 2025.

[AQH+21] Mahmoud Abdelkhalek, Jinyi Qiu, Michelle Hernandez, Alper Bozkurt, and Edgar 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)*, pages 7103–7107. IEEE, 2021.

[ATT+18a] 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)*, pages 81–88, 2018.

[ATT+18b] Aydin Aysu, Youssef Tobah, Mohit Tiwari, Andreas Gerstlauer, and Michael Orshansky. Horizontal side-channel vulnerabilities of post-quantum key exchange protocols. In *2018 IEEE international symposium on hardware oriented security and trust (HOST)*, pages 81–88. IEEE, 2018.

[BCM+25] Marcus Brinkmann, Chitchanok Chuengsatiansup, Alexander May, Julian Nowakowski, and Yuval Yarom. Leaky mceliece: Secret key recovery from highly erroneous side-channel information. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(2):94–125, 2025.

[BFM+18] Joppe W Bos, Simon Friedberger, Marco Martinoli, Elisabeth Oswald, and Martijn Stam. Assessing the feasibility of single trace power analysis of frodo. In *International Conference on Selected Areas in Cryptography*, pages 216–234. Springer, 2018.

[BHK+19] Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The sphincs+ signature framework. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2129–2146, 2019.

[BJL+14] Aurélie Bauer, Eliane Jaulmes, Victor Lomné, Emmanuel Prouff, and Thomas Roche. Side-channel attack against rsa key generation algorithms. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 223–241. Springer, 2014.

[BMD22] Martin Brisfors, Michail Moraitis, and Elena Dubrova. Side-channel attack countermeasures based on clock randomization have a fundamental flaw. Cryptology ePrint Archive, Paper 2022/1416, 2022.

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

[CCJ+16] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[CHH24] Keon-Hee Choi, Jaeseung Han, and Dong-Guk Han. Single trace analysis of visible vs. invisible leakage for comparison-operation-based cdt sampling. *Electronics*, 13(23):4681, 2024.

[CMM17] Núria Costa, Ramiro Martínez, and Paz Morillo. Proof of a shuffle for lattice-based cryptography. In *Nordic Conference on Secure IT Systems*, pages 280–296. Springer, 2017.

[CRR02a] Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 13–28. Springer, 2002.

[CRR02b] Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 13–28. Springer, 2002.

[DG25] Haiyue Dong and Qian Guo. Ot-pca: New key-recovery plaintext-checking oracle based side-channel attacks on hqc with offline templates. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):251–274, 2025.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976. Conference Name: IEEE Transactions on Information Theory.

[GJJ22] Qian Guo, Andreas Johansson, and Thomas Johansson. A key-recovery side-channel attack on classic mceliece implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 800–827, 2022.

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

[GR24] Morgane Guerreau and Mélissa Rossi. A not so discrete sampler: Power analysis attacks on hawk signature scheme. *Cryptology ePrint Archive*, 2024.

[HCM24] Suvadeep Hajra, Siddhartha Chowdhury, and Debdeep Mukhopadhyay. Estranet: An efficient shift-invariant transformer network for side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(1):336–374, 2024.

[JNWD24] Sönke Jendral, Kalle Ngo, Ruize Wang, and Elena Dubrova. Breaking sca-protected crystals-kyber with a single trace. In *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 70–73. IEEE, 2024.

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

[KAA21] Emre Karabulut, Erdem Alkim, and Aydin Aysu. Single-trace side-channel attacks on ω-small polynomial sampling: with applications to ntru, ntru prime, and crystals-dilithium. In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 35–45. IEEE, 2021.

[KAP+20] Priyank Kashyap, Furkan Aydin, Seetal Potluri, Paul Franzon, and Aydin 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*, pages 1–1, 2020.

[KDYA25] Ashley Kurian, Anuj Dubey, Ferhat Yaman, and Aydin Aysu. Tpuxtract: An exhaustive hyperparameter extraction framework. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):78–103, 2025.

[KH18] Suhri Kim and Seokhie Hong. Single trace analysis on constant time cdt sampler and its countermeasure. *Applied Sciences*, 8(10):1809, 2018.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual international cryptology conference*, pages 388–397. Springer, 1999.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[LDK+20] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. Crystals-dilithium. *Algorithm Specifications and Supporting Documentation*, 2020.

[LLJ+24] Jianqing Liu, Thinh Le, Tingxiang Ji, Ruozhou Yu, Demitry Farfurnik, Greg Bryd, and Daniel Stancil. The road to quantum internet: Progress in quantum network testbeds and major demonstrations. *Progress in Quantum Electronics*, 2024.

[LTYZ25] Xiuhan Lin, Mehdi Tibouchi, Yang Yu, and Shiduo Zhang. Do not disturb a sleeping falcon. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 213–244. Springer, 2025.

[LW23] Han Li and Thomas J Webster. Trends in nanomedicine. In *Nanomedicine*, pages 1–18. Elsevier, 2023.

[LZY+25] Xiuhan Lin, Shiduo Zhang, Yang Yu, Weijia Wang, Qidi You, Ximing Xu, and Xiaoyun Wang. Thorough power analysis on falcon gaussian samplers and practical countermeasure. *Cryptology ePrint Archive*, 2025.

[MHS+19] Sarah McCarthy, James Howe, Neil Smyth, Séamus Brannigan, and Máire OâĂŹNeill. Bearz attack falcon: implementation attacks with countermeasures on the falcon signature scheme. *Cryptology ePrint Archive*, 2019.

[MKAA24] Arsalan Ali Malik, Emre Karabulut, Amro Awad, and Aydin Aysu. Enabling secure and efficient sharing of accelerators in expeditionary systems. *Journal of Hardware and Systems Security*, 8(2):94–112, 2024.

[MKK+23] Soundes Marzougui, Ievgen Kabin, Juliane Krämer, Thomas Aulbach, and Jean-Pierre Seifert. On the feasibility of single-trace attacks on the gaussian sampler using a cdt. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 149–169. Springer, 2023.

[MMA25] Arsalan Ali Malik, Harshvadan Mihir, and Aydin Aysu. Craft: Characterizing and root-causing fault injection threats at pre-silicon. *arXiv preprint arXiv:2503.03877*, 2025.

[Mon94] Peter L Montgomery. A survey of modern integer factorization algorithms. *CWI quarterly*, 7(4):337–366, 1994.

[MRL14] Matthew A Morrison, Nagarajan Ranganathan, and Jay Ligatti. Design of adiabatic dynamic differential logic for dpa-resistant secure integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(8):1381–1389, 2014.

[MWG+22] Soundes Marzougui, Nils Wisiol, Patrick Gersch, Juliane Krämer, and Jean-Pierre 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*, pages 1–11, 2022.

[NDGJ21] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked ind-cca secure saber kem implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 676–707, 2021.

[OEKBN+24] Daniel Owens, Rabih El Khatib, Mojtaba Bisheh-Niasar, Reza Azarderakhsh, and Mehran Mozaffari Kermani. Efficient and side-channel resistant ed25519 on arm cortex-m4. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.

[Pes16] Peter Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In *International Conference on Cryptology in India*, pages 153–170. Springer, 2016.

[PFH+20] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon. *Post-Quantum Cryptography Project of NIST*, 2020.

[PK25] Eric Chun-Yu Peng and Markus G Kuhn. Adaptive template attacks on the kyber binomial sampler. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(3):470–492, 2025.

[Pla08] André Platzer. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41(2):143–189, 2008.

[PP19] Peter Pessl and Robert Primas. More practical single-trace attacks on the number theoretic transform. In *International Conference on Cryptology and Information Security in Latin America*, pages 130–149. Springer, 2019.

[PPM17a] Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 513–533, 2017.

[PPM17b] Robert Primas, Peter Pessl, and Stefan 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*, pages 513–533. Springer, 2017.

[QWY+25] Shipei Qu, Yuxuan Wang, Jintong Yu, Chi Zhang, and Dawu Gu. Trace copilot: Automatically locating cryptographic operations in side-channel traces by firmware binary instrumenting. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):128–159, 2025.

[RBL24] Tomáš Rabas, Jiří Buček, and Róbert Lórencz. Single-trace side-channel attacks on ntru implementation. *SN Computer Science*, 5:239, 2024.

[RJH+18] Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Side-channel assisted existential forgery attack on dilithium-a nist pqc candidate. *IACR Cryptol. ePrint Arch.*, 2018:821, 2018.

[RPC22] Anca Rădulescu, Pantelimon G Popescu, and Marios O Choudary. Ge vs gm: Efficient side-channel security evaluations on full cryptographic keys. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 886–905, 2022.

[RPJ+24] Prasanna Ravi, Thales Paiva, Dirmanto Jap, Jan-Pieter DâĂŹanvers, and Shivam Bhasin. Defeating low-cost countermeasures against side-channel attacks in lattice-based encryption. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024.

[RRCB20] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on cca-secure lattice-based pke and kems. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 307–335, 2020.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sch24] Markus Schönauer. *On the Physical Security of Falcon*. PhD thesis, Technische Universität Wien, 2024.

[SD25] Vladimir Sarde and Nicolas Debande. Improvement of side-channel attacks on mitaka. *Cryptology ePrint Archive*, 2025.

[SG24] Shreyas Sen and Archisman Ghosh. Circuit-level techniques for side-channel attack resilience: A tutorial. *IEEE Solid-State Circuits Magazine*, 16(4):96–108, 2024.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[SKL+20] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Taeho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk 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.

[SY23] Kiarash Sedghighadikolaei and Attila Altay Yavuz. A comprehensive survey of threshold signatures: Nist standards, post-quantum cryptography, exotic techniques, and real-world applications. *arXiv preprint arXiv:2311.05514*, 2023.

[TV04] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 1, pages 246–251. IEEE, 2004.

[YCY+24] Tianrun Yu, Chi Cheng, Zilong Yang, Yingchen Wang, Yanbin Pan, and Jian Weng. Hints from hertz: Dynamic frequency scaling side-channel analysis of number theoretic transform in lattice-based kems. *Cryptology ePrint Archive*, 2024.

[ZLYW23] Shiduo Zhang, Xiuhan Lin, Yang Yu, and Weijia Wang. Improved power analysis attacks on falcon. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 565–595. Springer, 2023.