

When Masking Preimage Computation Isn't Enough: A Power Analysis Attack on Masked Implementations of Falcon

Keng-Yu Chen¹ [0009-0007-1352-4582]

EPFL, Lausanne, Switzerland keng-yu.chen@epfl.ch

Abstract. In this paper, we present a novel power analysis attack targeting a FALCON implementation protected by a masked preimage computation. By analyzing the ratio between the real and imaginary parts of the public hash polynomial, we demonstrate that the leakage during the share recombination phase can be exploited to recover the secret key. We propose two attack variants: a chosen-message attack, where the adversary controls input messages to force extreme coefficient ratios, and a non-chosen-message attack that filters for naturally occurring vulnerable hashes. Through experimental evaluations using the ELMO leakage simulator for an ARM Cortex-M0 architecture, our experiments achieve a 98.3% success rate with 4,000 traces. Finally, while a fully masked implementation of Falcon prevents our attack, we propose a practical countermeasure based on rejection sampling to avoid the prohibitive computational overhead of a fully masked Gaussian sampler.

Keywords: FALCON · Side-Channel Analysis · Post-Quantum Cryptography · Floating-Point Arithmetic · Masking

1 Introduction

Since Shor's seminal work [29] on finding the period of a function using large-scale quantum computers, the security of widely used public-key cryptographic schemes such as RSA [27] and ECC [15] has been threatened. To prepare for the quantum era, the National Institute of Standards and Technology (NIST) initiated a standardization process for post-quantum cryptography (PQC) in 2016 [30]. After a rigorous evaluation process, NIST announced several winners, including CRYSTALS-Kyber/ML-KEM [28] and HQC [1] for key-encapsulation mechanisms (KEM) and CRYSTALS-Dilithium/ML-DSA [19], FALCON/FN-DSA [24], and SPHINCS⁺/SLH-DSA [14] for digital signatures. ML-KEM, ML-DSA, and SLH-DSA have been formally standardized, while the standardization of HQC and FALCON is still in progress.

In this work, we focus on FALCON, which is a lattice-based digital signature scheme based on the GPV Hash-then-Sign framework [10] and the NTRU lattice [12]. It is designed to be compact and efficient, yielding signature sizes of approximately 666 bytes and 1280 bytes for NIST security levels 1 and 5, respectively.

Despite its efficiency and security, FALCON has been shown to be vulnerable to side-channel attacks. These attacks exploit the physical leakage of a cryptographic implementation, such as the execution time, power consumption, or electromagnetic radiation,

to recover the secret key. Early works [4,9,23,21] demonstrated that the running time of FALCON can be leveraged to break its security. To mitigate this issue, an isochronous Gaussian distribution sampler [32,13] was subsequently developed. Regarding power consumption and electromagnetic radiation, several prior works [16,11,7,6] show that the preimage computation in FALCON can leak information about the secret key, while others [11,31,18] highlight the vulnerabilities in the subsequent Gaussian distribution sampling stage. Furthermore, recent studies [25,26] have demonstrated that the key generation process and the Number Theoretic Transform (NTT) of FALCON can also leak secret information through power analysis.

To defend against these power and electromagnetic threats, masking is a widely used technique that randomizes the intermediate values of a cryptographic implementation to prevent leakage of information about the secret key. A software masked implementation of FALCON was first introduced in [5], which focuses on masking the preimage computation, and a hardware design was later proposed in [17]. However, these implementations protect only the preimage computation and leave the subsequent Gaussian sampling stage unprotected.

In [2], masked floating-point floor and inverse functions were developed, which yields a full masking scheme for FALCON when combined with the work of [5]. However, the high overhead of masked floating-point inversion, which requires 55 masked floating-point additions, renders the full masking scheme of FALCON impractical. A fully 2-shared masked FALCON-512 implementation on their laptop took roughly 3.2 seconds even without signature rejection, with Gaussian sampling accounting for 99% of the runtime. Given this high overhead, a hybrid solution that masks only the preimage computation while leaving Gaussian sampling unmasked offers a substantially lower-overhead alternative. Still, it remains unclear whether such a hybrid solution is secure against power analysis attacks.

1.1 Our Contributions

In this paper, we show that this hybrid masking approach is vulnerable to a power analysis attack. Concretely, we present the following contributions:

- We propose a novel power analysis attack on a FALCON implementation with a masked preimage computation. More concretely, we present a chosen-message attack that assumes the attacker can control the input message and knows the random salt used in hashing it. Our attack can be extended to a non-chosen-message attack, where the attacker cannot control the input message or know the random salt in advance, by filtering for signatures and power traces that satisfy the conditions of the chosen-message attack. Our approach builds on the work of [6]; however, while their attack only works for an implementation where floating-point multiplication is masked, our attack applies to implementations where both floating-point multiplication and addition are masked.
- We present a theoretical analysis of our attack to guide parameter selection. In particular, we follow the strategy of [6] to model the distribution of the secret key and public hashes as continuous distributions.

- We validate the effectiveness of our attack through experiments on the leakage simulator ELMO [20]. We present the results of our attack across different parameters. Our experiment achieved a success rate of 98.3% using 4,000 traces for the chosen-message attack, and 96.9% using 298,062 traces for the non-chosen-message attack.
- We propose a countermeasure against our attack by rejecting the generated hash polynomial if it qualifies for the chosen-message attack. We provide different trade-offs between resistance to our attack and the probability of accepting a hash polynomial that does not qualify for the chosen-message attack.

1.2 Organization

The rest of the paper is organized as follows. In Section 2, we provide the necessary background for introducing our attack, including an overview of FALCON and the power analysis attack proposed in [6]. In Section 3, we present our attack in detail. In Section 4, we provide a theoretical analysis of our attack to guide the choice of parameters. We validate the effectiveness of our attack through simulated experiments using ELMO in Section 5. Potential countermeasures against our attack are discussed in Section 6. Finally, we conclude our work and discuss limitations and directions for future research in Section 7.

2 Preliminaries

2.1 Notation

In this paper, let $n = 2^\kappa$ for some integer $\kappa \in \{9, 10\}$, and let $q = 12289$ be a prime number. For a polynomial $f(X) \in \mathbb{Z}[X]/(X^n + 1)$, we denote its degree- k coefficient as $f[k]$; that is, $f(X) = f[0] + f[1]X + \dots + f[n-1]X^{n-1}$. We may also represent f as an $n \times n$ matrix, where the i -th row consists of the coefficients of $(X^i f \bmod (X^n + 1))$ for $0 \leq i < n$. Vector and matrix additions and multiplications naturally map to the corresponding polynomial operations in the ring $\mathbb{Z}[X]/(X^n + 1)$.

2.2 Discrete Gaussian Distribution

For real numbers $\sigma > 0$ and c , the Gaussian function is defined as

$$\rho_{\sigma,c}(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right).$$

The *discrete Gaussian distribution* over integers is defined by its probability mass function:

$$D_{\mathbb{Z},\sigma,c}(z) = \frac{\rho_{\sigma,c}(z)}{\sum_{x \in \mathbb{Z}} \rho_{\sigma,c}(x)}.$$

The parameters σ and c represent the standard deviation and the center of the distribution, respectively.

Let $\mathbf{B}$ be an $m \times m$ matrix for some integer m . The lattice $\mathcal{L}(\mathbf{B})$ generated by $\mathbf{B}$ is the set of vectors

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{zB} \mid \mathbf{z} \in \mathbb{Z}^m\}.$$

For a center vector $\mathbf{c} \in \mathbb{R}^m$, we analogously define the function $\rho_{\sigma, \mathbf{c}}(\mathbf{v}) = \exp\left(-\frac{\|\mathbf{v}-\mathbf{c}\|^2}{2\sigma^2}\right)$. The discrete Gaussian distribution over a lattice $\mathcal{L}(\mathbf{B})$ is defined by the following probability mass function:

$$D_{\mathcal{L}(\mathbf{B}), \sigma, \mathbf{c}}(\mathbf{w}) = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{w})}{\sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \rho_{\sigma, \mathbf{c}}(\mathbf{v})}.$$

In FALCON, each coefficient of the secret polynomials f and g is drawn from a discrete Gaussian distribution over integers. Additionally, the signature is a vector drawn from a discrete Gaussian distribution over a lattice generated by f and g .

2.3 Falcon Signature Scheme

In this section, we briefly recall the FALCON signature scheme. We refer the reader to the NIST round-3 FALCON specification [24] for more details.

FALCON is based on the GPV signature scheme [10] instantiated over NTRU lattices. In the GPV framework, the public key is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and the secret key is a small matrix $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$ that forms a basis of the lattice $\mathcal{L}(\mathbf{B}) = \{\mathbf{zB} \mid \mathbf{z} \in \mathbb{Z}^m\}$ and satisfies $\mathbf{BA}^T = \mathbf{0}$. The signature is a short vector $\mathbf{s}$ such that $\mathbf{sA}^T = \mathbf{c}$, where $\mathbf{c}$ is the hash of the message. The security of the scheme relies on the hardness of finding a vector in the lattice $\mathcal{L}(\mathbf{B})$ that is exceptionally close to $\mathbf{c}$.

In FALCON, the secret key consists of four polynomials $f, g, F, G \in \mathbb{Z}_q[X]/(X^n + 1)$. Each coefficient of f and g is drawn from a discrete Gaussian distribution $D_{\mathbb{Z}, \sigma_{\{f, g\}}, 0}$, where $\sigma_{\{f, g\}} = 1.17\sqrt{q/2n}$. The polynomials F and G are computed such that they satisfy the NTRU equation:

$$fG - gF = q \pmod{X^n + 1}.$$

The public key is the polynomial $h = gf^{-1} \pmod{X^n + 1, q}$. We also represent the public key and the secret key in matrix form as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & | & h \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} g & | & -f \\ G & | & -F \end{bmatrix}.$$

To sign a message, we first compute the hash polynomial of the message $c = H(r||m) \in \mathbb{Z}_q[X]/(X^n + 1)$ using a hash function H , where r is a random salt. Next, we compute the *preimage* vector $\mathbf{t}$ such that $\mathbf{tB} = \mathbf{c} := (c, 0)$:

$$\mathbf{t} = \mathbf{cB}^{-1} = (c, 0) \times \mathbf{B}^{-1}.$$

In order to compute $\mathbf{t}$ efficiently, FALCON applies the fast Fourier transform to c , f , and F to evaluate $\mathbf{t}$ in the FFT domain, which we describe in Section 2.4. The *fast Fourier nearest plane* algorithm [8] is then applied to find an integer vector $\mathbf{z}$ such that

$\mathbf{zB}$ follows the discrete Gaussian distribution $D_{\mathcal{L}(\mathbf{B}), \sigma, \mathbf{c}}$ for some parameter σ . The signature is $\mathbf{s} = (\mathbf{t} - \mathbf{z})\mathbf{B} = \mathbf{c} - \mathbf{zB}$. To minimize the signature size, the final output consists of the salt r and only the second polynomial component s_2 of $\mathbf{s}$. To verify the signature, the verifier recovers $c = H(r||m)$ and s_1 by $s_1 + s_2h = c$ and checks that $\mathbf{s}$ is short.

The full signing algorithm Sign is given in Algorithm 1. Note that the secret key $\mathbf{sk}$ includes the matrix $\mathbf{B}$ and a FALCON tree T , which is utilized in the fast Fourier nearest plane algorithm.

Algorithm 1 Sign [24]

input: A message m , a secret key $\mathbf{sk}$, and bound $\lfloor \beta^2 \rfloor$

output: Signature sig of m

```

1:  $r \leftarrow \{0, 1\}^{320}$ 
2:  $c \leftarrow H(r||m)$ 
3:  $\mathbf{t} \leftarrow \left(-\frac{1}{q}\text{FFT}(c) \odot \text{FFT}(F), \frac{1}{q}\text{FFT}(c) \odot \text{FFT}(f)\right)$  ▷ preimage computation
4: do
5:   do
6:      $\mathbf{z} \leftarrow \text{ffSampling}(\mathbf{t}, T)$  ▷ fast Fourier nearest plane algorithm
7:      $\mathbf{s} \leftarrow (\mathbf{t} - \mathbf{z})\hat{\mathbf{B}}$  ▷  $\hat{\mathbf{B}}$  is the fast Fourier transform of  $\mathbf{B}$ 
8:     while  $\|\mathbf{s}\|^2 > \lfloor \beta^2 \rfloor$ 
9:        $(s_1, s_2) \leftarrow \text{invFFT}(\mathbf{s})$ 
10:       $\mathbf{s} \leftarrow \text{Compress}(s_2)$ 
11: while  $\mathbf{s} = \perp$ 
12:  $\text{sig} \leftarrow (r, \mathbf{s})$ 
13: return  $\text{sig}$ 
```

2.4 Fast Fourier Transform

To sign a message efficiently, FALCON applies the fast Fourier transform to compute the preimage $\mathbf{t}$ and executes the fast Fourier nearest plane algorithm. Let Ω_n be the set of all complex roots $\left\{\exp\left(\frac{i\pi(2k+1)}{n}\right) \mid 0 \leq k < n\right\}$ of $X^n + 1$, the fast Fourier transform of a polynomial f of degree $n - 1$ is defined as

$$\hat{f} = \text{FFT}(f) = (f(\xi))_{\xi \in \Omega_n}.$$

Note that the k -th coefficient is given by:

$$\hat{f}[k] = \sum_{m=0}^{n-1} f[m] \exp\left(\frac{i\pi \cdot (2k+1) \cdot m}{n}\right).$$

We say f is in the *coefficient domain* and $\hat{f}$ is in the *FFT domain*.

Polynomial additions and multiplications in the coefficient domain can be efficiently computed by performing the corresponding pointwise operations on each complex

coefficient in the FFT domain. That is, for two polynomials c and f , we have

$$\text{FFT}(c + f) = \hat{c} + \hat{f}, \quad \text{FFT}(c \cdot f) = \hat{c} \odot \hat{f},$$

where $\odot$ denotes the pointwise multiplication of two vectors. To transform a polynomial $\hat{f}$ in the FFT domain back to the coefficient domain, we apply the inverse fast Fourier transform (invFFT):

$$f = \text{invFFT}(\hat{f}), \quad f[k] = \frac{1}{n} \sum_{m=0}^{n-1} \hat{f}[m] \exp\left(\frac{-i\pi \cdot (2m+1) \cdot k}{n}\right).$$

When the coefficients of f are real numbers, which is the case in FALCON, observe that:

$$\begin{aligned} \hat{f}[n-1-k] &= \sum_{m=0}^{n-1} f[m] \exp\left(\frac{i\pi \cdot (2(n-1-k)+1) \cdot m}{n}\right) \\ &= \sum_{m=0}^{n-1} f[m] \exp\left(\frac{-i\pi m(2k+1)}{n}\right) \\ &= \overline{\hat{f}[k]}, \end{aligned}$$

where $\overline{\hat{f}[k]}$ is the complex conjugate of $\hat{f}[k]$. Therefore, the second half of the coefficients of $\hat{f}$ can be derived from the first half. Consequently, to recover the full vector $\hat{f}$ and subsequently f , it suffices to recover only the first half of its coefficients.

2.5 Floating-Point Numbers and Operations

Implementing FALCON requires performing multiplications and additions over the complex field. A complex number $a = a_1 + a_2i$ is represented by two floating-point numbers a_1 and a_2 , corresponding to its real and imaginary parts. To ensure sufficient precision, FALCON follows the IEEE-754 standard and uses 64-bit floating-point numbers, each of which is composed of a 1-bit sign, an 11-bit exponent, and a 53-bit mantissa with a leading 1 implicit in storage. The exponent is biased by adding 1023 to simplify floating-point comparisons. We follow [5] to represent a 64-bit floating-point number x as a tuple (s, e, m) , where $s \in \{0, 1\}$, $e \in [0, 2^{11})$, and $m \in [2^{52}, 2^{53})$ are three non-negative integers, and the numerical value of x is given by

$$x = (-1)^s \cdot 2^{e-1023} \cdot (m \cdot 2^{-52}).$$

The detailed algorithms for floating-point multiplication and addition are omitted, as they are not related to our primary focus. We refer the reader to [5] for a comprehensive description of these operations.

2.6 Previous Attack on Floating-Point Product

We recall the power analysis attack on FALCON proposed by Chen et al. [6], which forms the foundation of our work. The authors target an implementation where the floating-point multiplications are masked while additions remain unmasked. This specific threat

model is considered motivated by the severe performance overhead of masking floating-point additions. Let c and f be the public hash and the secret polynomial, respectively, and let $\hat{c}$ and $\hat{f}$ be their FFT representations. The preimage computation in FALCON involves evaluating the pointwise complex multiplication $\hat{c} \cdot \hat{f}$ in the FFT domain; that is,

$$(\hat{c}_1 \odot \hat{f}_1 - \hat{c}_2 \odot \hat{f}_2) + (\hat{c}_1 \odot \hat{f}_2 + \hat{c}_2 \odot \hat{f}_1)i,$$

where $\hat{c}_1$ and $\hat{c}_2$ denote the real and imaginary parts of $\hat{c}$, and $\hat{f}_1$ and $\hat{f}_2$ denote the real and imaginary parts of $\hat{f}$. Because floating-point addition is left unmasked, the implementation must eventually recombine the shares of the following four intermediate floating-point products for each coefficient index k :

$$\hat{c}_1[k] \cdot \hat{f}_1[k], \hat{c}_2[k] \cdot \hat{f}_2[k], \hat{c}_1[k] \cdot \hat{f}_2[k], \hat{c}_2[k] \cdot \hat{f}_1[k].$$

Since c is public, an adversary can conduct Correlation Power Analysis (CPA) [3] on the recombination of these products. More concretely, let (s_c, e_c, m_c) and (s_f, e_f, m_f) denote the two operands (i.e., $\hat{c}_1[k]$ or $\hat{c}_2[k]$, and $\hat{f}_1[k]$ or $\hat{f}_2[k]$, respectively), and let (s, e, m) represent their floating-point product. The attack sequentially recovers the sign bit, mantissa, and exponent of $\hat{f}_1[k]$ or $\hat{f}_2[k]$ using the following relations, which hold with high probability:

$$\begin{aligned} s &= s_c \oplus s_f \\ e &= e_c + e_f + \mathbb{1}_{m_c \times m_f \geq 2^{105}} \\ m^{[53:53-h]} &= \begin{cases} (m_c \times m_f)^{[105:105-h]} & \text{if } m_c \times m_f < 2^{105} \\ (m_c \times m_f)^{[106:106-h]} & \text{if } m_c \times m_f \geq 2^{105} \end{cases} \end{aligned}$$

Here, the superscript $[i : j]$ indicates the bit range from i to j , inclusive, with the least significant bit (LSB) indexed as 1. Building on the techniques of [11], this attack targets only the first $h + 1$ most significant bits of the mantissa (including the implicit leading 1), where h is a tunable parameter that influences the success rate and efficiency of the attack. The authors demonstrated that recovering $h + 1 = 12$ bits of the mantissa is sufficient to reconstruct the full secret coefficients in the coefficient domain with overwhelming probability.

Moreover, they showed that the exponents of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ are highly concentrated around a few specific values. We recall the probabilities for each exponent observed by Chen et al. in Table 1. Based on this distribution and the union-bound analysis of [6], restricting the attack to exponents within $[1016, 1031]$ covers all exponents with probability over 95% for $n = 512$ and over 90% for $n = 1024$. Furthermore, the authors demonstrated that targeting exponents outside this narrow range significantly increases the number of power traces required for the attack.

3 Proposed Attack

3.1 Threat Model

We assume a threat model similar to that of [6], where the adversary can record the power traces of a device running FALCON with an unknown secret key and observe the resulting signatures. However, unlike their target implementation where only floating-point

Table 1: Occurrence probability (%) of exponents of FALCON’s secret key f after the fast Fourier transform examined in [6].

Exponent	<1013	1013	1014	1015	1016	1017	1018
Probability	0.001	0.001	0.002	0.004	0.009	0.019	0.038
Exponent	1019	1020	1021	1022	1023	1024	1025
Probability	0.076	0.153	0.307	0.615	1.230	2.457	4.899
Exponent	1026	1027	1028	1029	1030	1031	>1031
Probability	9.669	18.342	29.800	27.529	4.832	0.007	$3 \cdot 10^{-12}$

multiplications are masked, we assume the device executes a fully masked preimage computation that encompasses both multiplication and addition, such as the masking scheme in [5]. Our attack targets the recombination of shares after the preimage computation. Since no practically efficient masking scheme for the subsequent Gaussian sampling stage currently exists, such share recombination step is necessary.

We consider two variants: a chosen-message attack and a non-chosen-message attack. For the chosen-message attack, we assume that the adversary can control the input message and knows the random salt used during the signature generation. This may occur when the randomness is generated beforehand or derived from a counter or a pseudo-random number generator (PRNG) with a known state. More generally, the salt may be exposed through compromised PRNG states, random buffers, or auxiliary side-channel information. We emphasize that this represents an auxiliary-information setting rather than the standard FALCON signing interface.

For the non-chosen-message attack, we assume the adversary can only record power traces of the signing process without controlling the input message or knowing the randomness beforehand. In this scenario, the adversary filters the collected traces to isolate those that inadvertently satisfy the chosen-message conditions described below and then applies the chosen-message attack.

3.2 Attack Overview

We detail the chosen-message attack. As noted, the non-chosen-message attack is a direct extension derived by filtering for power traces corresponding to hashes that satisfy the chosen-message criteria. Naturally, this filtering approach requires a significantly larger number of power traces. We estimate the required trace count for the non-chosen-message attack theoretically in Section 4 and experimentally in Section 5.

Let c and f be the public hash and the secret polynomial, respectively, and let $\hat{c}$ and $\hat{f}$ be their FFT representations. The preimage computation in FALCON involves evaluating the pointwise complex multiplication $\hat{c} \cdot \hat{f}$ in the FFT domain; that is,

$$\begin{aligned}\hat{t}_1 &= \hat{c}_1 \odot \hat{f}_1 - \hat{c}_2 \odot \hat{f}_2, \\ \hat{t}_2 &= \hat{c}_1 \odot \hat{f}_2 + \hat{c}_2 \odot \hat{f}_1.\end{aligned}$$

Our attack exploits the following observation. Let $B \geq 1$ and $D \geq B$ be two integers. For each coefficient index $k \in \{0, 1, \dots, n-1\}$:

- Assuming that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < B$ and $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$, we have

$$\begin{aligned} |\hat{t}_1[k]| &\leq |\hat{c}_1[k] \cdot \hat{f}_1[k]| + |\hat{c}_2[k] \cdot \hat{f}_2[k]| < \left(1 + \frac{B}{D}\right) |\hat{c}_2[k] \cdot \hat{f}_2[k]|, \\ |\hat{t}_1[k]| &\geq -|\hat{c}_1[k] \cdot \hat{f}_1[k]| + |\hat{c}_2[k] \cdot \hat{f}_2[k]| > \left(1 - \frac{B}{D}\right) |\hat{c}_2[k] \cdot \hat{f}_2[k]|. \end{aligned}$$

Therefore, $|\hat{t}_1[k]|$ is within $(1 \pm \frac{B}{D}) |\hat{c}_2[k] \cdot \hat{f}_2[k]|$. Since $|\hat{c}_1[k] \cdot \hat{f}_1[k]| < |\hat{c}_2[k] \cdot \hat{f}_2[k]|$, $\hat{t}_1[k]$ is within $(1 \pm \frac{B}{D}) \cdot (-\hat{c}_2[k] \cdot \hat{f}_2[k])$.

- Assuming that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > \frac{1}{B}$ and $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$, we have

$$\begin{aligned} |\hat{t}_1[k]| &\leq |\hat{c}_1[k] \cdot \hat{f}_1[k]| + |\hat{c}_2[k] \cdot \hat{f}_2[k]| < \left(1 + \frac{B}{D}\right) |\hat{c}_1[k] \cdot \hat{f}_1[k]|, \\ |\hat{t}_1[k]| &\geq |\hat{c}_1[k] \cdot \hat{f}_1[k]| - |\hat{c}_2[k] \cdot \hat{f}_2[k]| > \left(1 - \frac{B}{D}\right) |\hat{c}_1[k] \cdot \hat{f}_1[k]|. \end{aligned}$$

Therefore, $|\hat{t}_1[k]|$ is within $(1 \pm \frac{B}{D}) |\hat{c}_1[k] \cdot \hat{f}_1[k]|$. Since $|\hat{c}_2[k] \cdot \hat{f}_2[k]| < |\hat{c}_1[k] \cdot \hat{f}_1[k]|$, $\hat{t}_1[k]$ is within $(1 \pm \frac{B}{D}) \cdot (\hat{c}_1[k] \cdot \hat{f}_1[k])$.

- Similarly, assuming that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < B$ and $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$, $\hat{t}_2[k]$ is within $(1 \pm \frac{B}{D}) \cdot (\hat{c}_1[k] \cdot \hat{f}_2[k])$.
- Similarly, assuming that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > \frac{1}{B}$ and $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$, $\hat{t}_2[k]$ is within $(1 \pm \frac{B}{D}) \cdot (\hat{c}_2[k] \cdot \hat{f}_1[k])$.

Assuming D is sufficiently large and B is sufficiently small, we can adapt the attack methodology of [6]. Specifically, for coefficient indices k where the ratio $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \in [\frac{1}{B}, B]$, one term in the complex multiplication dominates. We can thus approximate the intermediate values as follows:

- If $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$, we approximate $\hat{t}_1[k]$ as $-\hat{c}_2[k] \cdot \hat{f}_2[k]$ and recover $\hat{f}_2[k]$.
- If $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$, we approximate $\hat{t}_1[k]$ as $\hat{c}_1[k] \cdot \hat{f}_1[k]$ and recover $\hat{f}_1[k]$.
- If $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$, we approximate $\hat{t}_2[k]$ as $\hat{c}_1[k] \cdot \hat{f}_2[k]$ and recover $\hat{f}_2[k]$.
- If $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$, we approximate $\hat{t}_2[k]$ as $\hat{c}_2[k] \cdot \hat{f}_1[k]$ and recover $\hat{f}_1[k]$.

3.3 Parameter Selection

The precision of our approximation heavily relies on the ratio $\frac{B}{D}$. With a small $\frac{B}{D}$, the value $\hat{t}_1[k]$ or $\hat{t}_2[k]$ closely approximates $-\hat{c}_2[k] \cdot \hat{f}_2[k]$, $\hat{c}_1[k] \cdot \hat{f}_1[k]$, $\hat{c}_1[k] \cdot \hat{f}_2[k]$, or $\hat{c}_2[k] \cdot \hat{f}_1[k]$, allowing us to recover $\hat{f}_1[k]$ or $\hat{f}_2[k]$ with high probability.

A naive strategy is to find B such that the condition $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \in [\frac{1}{B}, B]$ holds for all indices $k \in \{0, 1, \dots, n-1\}$. We would then select a D such that $\frac{B}{D}$ is sufficiently

small for our desired precision, and search for messages and random salts that yield $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$ or $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$. However, since D scales proportionally with B , finding a valid message-salt pair with a large D is computationally prohibitive. For example, applying Theorem 1 in Section 4 with the union bound for $n = 512$, we have to set $B = 1087$ to ensure that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \in [\frac{1}{B}, B]$ holds for all k with a probability of at least 70%. This would require $D \geq 17392$ to achieve $\frac{B}{D} \leq 2^{-4}$. By Theorem 2 in Section 4, the probability that an index k satisfies $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$ or $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$ is around $7 \cdot 10^{-5}$. Consequently, an attacker would need to run $7 \cdot 10^7$ trials on average just to find 100 message-salt pairs satisfying the necessary condition on $\hat{c}$.

To avoid the bottleneck of a large D , we relax our conditions: we select a smaller B such that $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \in [\frac{1}{B}, B]$ holds for *most* indices k , and treat the outliers differently. Specifically, for indices where $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \notin [\frac{1}{B}, B]$, one part of the secret is rather small compared to the other, and we recover the large part while ignoring the small part. That is:

- If $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < \frac{1}{B}$, then $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < B$ in particular. Therefore, we can still recover $\hat{f}_2[k]$ from $\hat{t}_1[k]$ (if $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$) or from $\hat{t}_2[k]$ (if $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$). We then simply approximate $\hat{f}_1[k]$ to be 0.
- If $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > B$, then $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > \frac{1}{B}$ in particular. Therefore, we can still recover $\hat{f}_1[k]$ from $\hat{t}_1[k]$ (if $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$) or from $\hat{t}_2[k]$ (if $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$). We then simply approximate $\hat{f}_2[k]$ to be 0.

In Table 2, we show the empirical probability that we fail to recover the exact secret polynomials in the coefficient domain using our approximation strategy. To evaluate this, we generated 10^6 samples of f , computed $\hat{f} = \text{FFT}(f)$, and defined the approximation $\hat{f}'$ of $\hat{f}$ using $B = 11$ by the following equation:

$$\hat{f}'[k] = \begin{cases} \hat{f}_1[k], & |\hat{f}_1[k]| > B|\hat{f}_2[k]|, \\ \hat{f}_2[k], & |\hat{f}_2[k]| > B|\hat{f}_1[k]|, \\ \hat{f}[k], & \text{otherwise} \end{cases} \quad (1)$$

Next, for each index k , we retained only the $h + 1$ most significant bits of the mantissa of $\hat{f}'[k]$ and set the rest of the bits to 0, according to the strategy outlined in [11,6]. Finally, we applied the inverse fast Fourier transform of the approximated $\hat{f}'$ and rounded it to the nearest integer.

Our experiments indicate that setting $B = 11$ and $h = 6^1$ yields an overwhelming success probability for both $n = 512$ and 1024. However, we note that the selection of h also depends on multiple practical factors, such as the precision of the recovered mantissa, as discussed in [6, Section 3.3]. In our experiments in Section 5, we empirically set $B = 11$ and $h = 10$ to ensure a negligible probability of recovery failure, accommodate higher noise levels in the traces, and permit more aggressive mantissa approximations.

¹ While the theoretical bound in [6] suggests $h \geq 11$, our empirical observations show this bound to be conservative.

Table 2: The empirical probability of successful recovery for different values of h . The probability is estimated by generating 10^6 samples of f and applying the approximation of Equation (1) with $B = 11$ followed by only retaining the $h + 1$ most significant bits of mantissa.

h	3	4	5	6	7	8
$n = 512$	0.014%	94.676%	99.992%	100.000%	100.000%	100.000%
$n = 1024$	28.446%	99.997%	100.000%	100.000%	100.000%	100.000%

3.4 Recovery of Exponents

The approximation strategy described above only recovers the sign bits and mantissas of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ effectively. This is because the hashes $\hat{c}_1[k]$ and $\hat{c}_2[k]$ on condition that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ are very large and lack sufficient variance in their exponents to facilitate a reliable CPA.

Let (s_1, m_1) and (s_2, m_2) denote the recovered sign bits and mantissas of $\hat{f}_1[k]$ and $\hat{f}_2[k]$, respectively. To recover the exponents of $\hat{f}_1[k]$ and $\hat{f}_2[k]$, we record an additional set of power traces using random unfiltered hashes; that is, without restrictions on the message-salt pairs. We then perform a search over all possible pairs (e_1, e_2) of both exponents of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ by calculating the hypothetical Hamming weights of the recombined shares for the two intermediate values:

$$\begin{aligned} & (s_{c_1}, e_{c_1}, m_{c_1}) \cdot (s_1, e_1, m_1) - (s_{c_2}, e_{c_2}, m_{c_2}) \cdot (s_2, e_2, m_2) \\ & (s_{c_1}, e_{c_1}, m_{c_1}) \cdot (s_2, e_2, m_2) + (s_{c_2}, e_{c_2}, m_{c_2}) \cdot (s_1, e_1, m_1) \end{aligned}$$

Here, $(s_{c_1}, e_{c_1}, m_{c_1})$ and $(s_{c_2}, e_{c_2}, m_{c_2})$ are the known sign bits, exponents, and mantissas of the public $\hat{c}_1[k]$ and $\hat{c}_2[k]$. We select the exponent pair that yields the maximum correlation between these modeled Hamming weights and the recorded power traces. As established in [6], exponents fall within the range $[1016, 1031]$ with a probability of at least 99.9%. Therefore, we only need to enumerate among $16^2 = 256$ pairs of exponents for each index.

In Section 4, we demonstrate that $\hat{f}_1[k]$ and $\hat{f}_2[k]$ are statistically independent, and their joint probability is approximately given by

$$P(x, y) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad \sigma = \sigma_{\{f, g\}} \cdot \sqrt{\frac{n}{2}}. \quad (2)$$

One can exploit this information to further optimize the recovery of exponents. For instance, it is highly improbable that $\hat{f}_1[k]$ and $\hat{f}_2[k]$ would both have small (< 1020) or large (> 1030) exponents simultaneously as suggested by Table 1. If multiple pairs of exponents yield similar high correlation values in CPA, the attacker can pick the pair with the largest probability. In our experimental evaluation in Section 5, we first isolate a candidate set of exponents by identifying the maximum successive difference among the sorted correlation values. From this filtered set, we select the pair with the

highest theoretical probability. While this specific heuristic yielded a high success rate in our experiments, the exact method for integrating the joint probability is flexible and depends on the trace quality and the attacker’s precise knowledge about the leakage model.

3.5 Recovery of Sign Bits of Small Coefficients

We note that for coefficients $\hat{c}[k]$ with small indices k , the value $|\hat{c}_2[k]|$ is significantly larger than $|\hat{c}_1[k]|$ with high probability. As demonstrated in the proof of Theorem 2 in Section 4, the expected value of $\hat{c}_1[k]$ is $\frac{q-1}{2} = 6,144$, whereas the expected value of $\hat{c}_2[k]$ is $\frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right)$. For example, when $k = 0$, the expected value of $|\hat{c}_2[0]|$ is around 2,002,626, which leads to a negligible probability of finding a hash where $|\hat{c}_1[0]| > D|\hat{c}_2[0]|$ for a large D , or where $\hat{c}_2[0] < 0$. Practically, this means we cannot recover the sign bits of $\hat{f}_1[0]$ or $\hat{f}_2[0]$ because all the public hash components we use in the attack (i.e., only $\hat{c}_2[0]$ and no $\hat{c}_1[0]$) will be positive, providing no information about the sign bits of the secret.

Since this issue only affects a few small indices, we can mitigate it by simply enumerating the missing sign bits for these coefficients. For instance, at $k = 8$, the expected value of $|\hat{c}_2[8]|$ drops to approximately 117,695, and the probability that $|\hat{c}_1[8]| > D|\hat{c}_2[8]|$ is no longer negligible, allowing the standard recovery method to succeed.

3.6 Summary

Our complete attack proceeds as follows. We pick the parameters $B = 11$, $D = B \cdot 64 = 704$, and $h = 10$. We emphasize that the choice of D depends on the trace quality and the computational resources of the attacker. A smaller D would allow us to find more valid message-salt pairs but reduce the precision of the approximation. Similarly, the choice of h also depends on the power traces and impacts the probability of recovery failure in the coefficient domain.

Let N_D be the number of power traces used for recovering the sign bits and mantissas for each index, and let N_e be the number of unfiltered traces used to recover the exponents.

Preparation We assume the chosen-message threat model, where the adversary can control the input message and know the random salt of the target device. For each index $k \in \{0, 1, \dots, \frac{n}{2} - 1\}^2$, we find N_D message-salt pairs satisfying $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$ or $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$. We then record power traces of running FALCON using these specific pairs and the unknown secret key. Let T_1 and T_2 denote the sets of power traces corresponding to the conditions $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} < \frac{1}{D}$ and $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$, respectively. We also record N_e power traces using random message-salt pairs without filtering. Let T_e denote this set of traces.

² Recall that the second half of coefficients can be derived from the complex conjugates of the first half.

Attack For each index $k \in \{0, 1, \dots, \frac{n}{2} - 1\}$, we apply the attack of [6] to the power traces collected in the preparation step. Let $\hat{t}_1[k] = \hat{c}_1[k] \cdot \hat{f}_1[k] - \hat{c}_2[k] \cdot \hat{f}_2[k]$ and $\hat{t}_2[k] = \hat{c}_1[k] \cdot \hat{f}_2[k] + \hat{c}_2[k] \cdot \hat{f}_1[k]$ be the unmasked floating-point values after the recombination of shares.

- To recover $\hat{f}_1[k]$, we approximate $\hat{t}_1[k]$ as $\hat{c}_1[k] \cdot \hat{f}_1[k]$ and use the trace set T_2 ; we also approximate $\hat{t}_2[k]$ as $\hat{c}_2[k] \cdot \hat{f}_1[k]$ and use the trace set T_1 . We extract the sign bit and mantissa suggested by the maximum correlation between the two, and let this maximum correlation value be ρ_1 .
- To recover $\hat{f}_2[k]$, we approximate $\hat{t}_1[k]$ as $-\hat{c}_2[k] \cdot \hat{f}_2[k]$ and use the trace set T_1 ; we also approximate $\hat{t}_2[k]$ as $\hat{c}_1[k] \cdot \hat{f}_2[k]$ and use the trace set T_2 . We extract the sign bit and mantissa suggested by the maximum correlation between the two, and let this maximum correlation value be ρ_2 .

If either correlation ρ_1 or ρ_2 exceeds a predefined empirical threshold, we accept the recovered values; otherwise, we assume the corresponding real or imaginary part is insignificant compared to the other part and approximate it as 0.

Next, we recover the exponents of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ by enumerating over all possible pairs of exponents within $[1016, 1031] \times [1016, 1031]$ and selecting the pair that yields the maximum correlation with the unfiltered traces in T_e . Note that during this stage, we compute the hypothetical values using the sign bits and mantissas recovered in the previous step. If the real or imaginary part of the coefficient was previously rejected and set to 0, we only run the exponent search of the other part over $[1016, 1031]$. As noted, the joint probability in Equation (2) can also be leveraged to further optimize the recovery.

We repeat the entire process for all indices $k \in \{0, \dots, \frac{n}{2} - 1\}$ to recover $\hat{f}_1$ and $\hat{f}_2$. We then apply the inverse fast Fourier transform and round the coefficients to the nearest integers to reconstruct f in the coefficient domain. Note that the secret polynomial g can be recovered from f and the public key h , and F and G can be recovered from f and g subsequently. If f or g is not a small polynomial, we brute-force the missing sign bits for small indices $k < 8$, as outlined in Section 3.5, until we find the correct f and g .

4 Theoretical Analysis

Let f be the FALCON secret polynomial, and let $\hat{f} = \text{FFT}(f) = \hat{f}_1 + \hat{f}_2 i$, where $\hat{f}_1$ and $\hat{f}_2$ are the real and the imaginary parts of $\hat{f}$, respectively. Let c be the public hash polynomial, and let $\hat{c} = \text{FFT}(c) = \hat{c}_1 + \hat{c}_2 i$, where $\hat{c}_1$ and $\hat{c}_2$ are the real and the imaginary parts of $\hat{c}$, respectively. In this section, we provide a theoretical analysis of the parameters used in our attack. Specifically, we show the following results:

- Theorem 1 gives the probability that $|\hat{f}_1[k]| > B|\hat{f}_2[k]|$ or $|\hat{f}_2[k]| > B|\hat{f}_1[k]|$ for each index k . This suggests that applying a large B intended to satisfy the ratio condition across *all* coefficients is impractical. This formally justifies our strategy of approximating the smaller real or imaginary part of $\hat{f}[k]$ to be 0 when one part is significantly larger than the other.

- Theorem 2 gives the probability that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ for each index k . This allows us to estimate the complexity in the preparation step of our chosen message attack to find effective hashes c .

Following the approach in [6], our analysis approximates the discrete distributions of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ for each index k using continuous Gaussian distributions. That is, we assume

$$\hat{f}_1[k] \sim \mathcal{N}(0, \sigma), \quad \hat{f}_2[k] \sim \mathcal{N}(0, \sigma),$$

where $\sigma = \sigma_{\{f,g\}} \cdot \sqrt{n/2} = 0.585 \cdot \sqrt{q}$.

Lemma 1. *For each index k , the distributions of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ are independent.*

Proof. Recall the fast Fourier transform of f is defined as

$$\hat{f}_1[k] = \sum_{m=0}^{n-1} f[m] \cdot \cos\left(\frac{\pi(2k+1)m}{n}\right), \quad \hat{f}_2[k] = \sum_{m=0}^{n-1} f[m] \cdot \sin\left(\frac{\pi(2k+1)m}{n}\right).$$

By the bilinearity of covariance, we have

$$\text{Cov}[\hat{f}_1[k], \hat{f}_2[k]] = \sum_{m_1, m_2=0}^{n-1} \text{Cov}[f[m_1], f[m_2]] \cos\left(\frac{\pi(2k+1)m_1}{n}\right) \sin\left(\frac{\pi(2k+1)m_2}{n}\right).$$

For each index $m_1 \neq m_2$, $f[m_1]$ and $f[m_2]$ are independent and $\text{Cov}[f[m_1], f[m_2]] = 0$. Therefore, the above formula is reduced to

$$\begin{aligned} & \sum_{m=0}^{n-1} \text{Var}(f[m]) \cos\left(\frac{\pi(2k+1)m}{n}\right) \sin\left(\frac{\pi(2k+1)m}{n}\right) \\ &= \frac{\sigma_{\{f,g\}}^2}{2} \sum_{m=0}^{n-1} \sin\left(\frac{2\pi(2k+1)m}{n}\right) = 0. \end{aligned}$$

Since $\hat{f}_1[k]$ and $\hat{f}_2[k]$ are jointly Gaussian, they are independent if their covariance is zero. □

Theorem 1. *Given an integer $B \geq 1$. For each index k , the probability that $|\hat{f}_1[k]| > B|\hat{f}_2[k]|$ or $|\hat{f}_2[k]| > B|\hat{f}_1[k]|$ is roughly $\frac{4}{\pi} \cdot \arctan(\frac{1}{B})$.*

Proof. Let $\sigma = \sigma_{\{f,g\}} \cdot \sqrt{n/2}$. By Lemma 1, the density function of the joint distribution of $\hat{f}_1[k]$ and $\hat{f}_2[k]$ is

$$P(x, y) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

The probability that $|\hat{f}_1[k]| > B|\hat{f}_2[k]|$ is

$$\iint_{|x| > B|y|} P(x, y) dx dy = \frac{1}{2\pi\sigma^2} \iint_{|x| > B|y|} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) dx dy.$$

We change the Cartesian coordinates to the polar coordinates. Let $x = r \cos \theta$ and $y = r \sin \theta$. Also, let $\alpha = \arctan(\frac{1}{B})$. The condition $|x| > B|y|$ is equivalent to $\theta \in [-\alpha, \alpha] \cup [\pi - \alpha, \pi + \alpha]$. We only consider the case $\theta \in [-\alpha, \alpha]$ since the other case is symmetric. Therefore, we have

$$\begin{aligned} \iint_{|x| > B|y|} P(x, y) dx dy &= \frac{2}{2\pi\sigma^2} \int_{-\alpha}^{\alpha} \int_0^{\infty} \exp\left(-\frac{r^2}{2\sigma^2}\right) r dr d\theta \\ &= \frac{2}{\pi\sigma^2} \int_0^{\alpha} d\theta \int_0^{\infty} r \cdot \exp\left(-\frac{r^2}{2\sigma^2}\right) dr \\ &= \frac{2\alpha}{\pi} \cdot \int_0^{\infty} \exp(-u) du \quad (u = \frac{r^2}{2\sigma^2}) \\ &= \frac{2\alpha}{\pi} \cdot \Gamma(1) = \frac{2\alpha}{\pi}, \end{aligned}$$

where Γ is the Gamma function. Similarly, the probability that $|\hat{f}_2[k]| > B|\hat{f}_1[k]|$ is also $\frac{2\alpha}{\pi}$. Therefore, the probability that $|\hat{f}_1[k]| > B|\hat{f}_2[k]|$ or $|\hat{f}_2[k]| > B|\hat{f}_1[k]|$ is $\frac{4\alpha}{\pi}$. $\square$

To validate the continuous Gaussian approximation, we sampled 10,000 FALCON secret polynomials f and compared the empirical probabilities with the theoretical estimates. They closely agree, with a relative error of at most 4% for all tested values of $B \leq 2048$.

By Theorem 1, we can upper bound the probability that there exists an index k such that $|\hat{f}_1[k]| > B|\hat{f}_2[k]|$ or $|\hat{f}_2[k]| > B|\hat{f}_1[k]|$ by $\frac{n}{2} \cdot \frac{4}{\pi} \cdot \arctan(\frac{1}{B})$. Note that we only evaluate half of the coefficients since the other half are simply their complex conjugates. For example, when $n = 512$, the probability of encountering an index k such that $|\hat{f}_1[k]| > 1087|\hat{f}_2[k]|$ or $|\hat{f}_2[k]| > 1087|\hat{f}_1[k]|$ is bounded above by 0.3. As we describe in Section 3, setting $B = 1087$ leads to a correspondingly large D and impractical complexity in the preparation step of our attack. Therefore, we adopt the approximation strategy by ignoring the real or imaginary part of $\hat{f}[k]$ whenever the ratio between them is heavily skewed.

Next, we estimate the probability about public hash c . We assume each coefficient of c is uniformly distributed over $[0, q)$.

Lemma 2. *For each index k , the covariance of $\hat{c}_1[k]$ and $\hat{c}_2[k]$ is 0.*

Proof. By the bilinearity of covariance, we have

$$\text{Cov}[\hat{c}_1[k], \hat{c}_2[k]] = \sum_{m_1, m_2=0}^{n-1} \text{Cov}[c[m_1], c[m_2]] \cos\left(\frac{\pi(2k+1)m_1}{n}\right) \sin\left(\frac{\pi(2k+1)m_2}{n}\right).$$

For each index $m_1 \neq m_2$, $c[m_1]$ and $c[m_2]$ are independent and $\text{Cov}[c[m_1], c[m_2]] = 0$. Therefore, the above formula is reduced to

$$\begin{aligned} & \sum_{m=0}^{n-1} \text{Var}(c[m]) \cos\left(\frac{\pi(2k+1)m}{n}\right) \sin\left(\frac{\pi(2k+1)m}{n}\right) \\ &= \frac{q^2-1}{12} \sum_{m=0}^{n-1} \frac{1}{2} \sin\left(\frac{2\pi(2k+1)m}{n}\right) = 0. \end{aligned}$$

□

Theorem 2. Given an integer $D \geq 1$. For each index k , the probability that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ is roughly

$$\phi_\rho(\mu'_1, \mu'_2) + \phi_\rho(-\mu'_1, -\mu'_2) + \phi_\rho(\mu''_1, \mu''_2) + \phi_\rho(-\mu''_1, -\mu''_2),$$

where ϕ_ρ is the cumulative distribution function of the bivariate Gaussian distribution with mean 0, variance 1, and covariance $\rho = \frac{1-D^2}{1+D^2}$, and

$$\begin{aligned} \mu'_1 &= \frac{\frac{q-1}{2} - D \cdot \frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right)}{\sqrt{\frac{n(q^2-1)}{24}(1+D^2)}}, & \mu'_2 &= \frac{\frac{q-1}{2} + D \cdot \frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right)}{\sqrt{\frac{n(q^2-1)}{24}(1+D^2)}}, \\ \mu''_1 &= \frac{\frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right) - D \cdot \frac{q-1}{2}}{\sqrt{\frac{n(q^2-1)}{24}(1+D^2)}}, & \mu''_2 &= \frac{\frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right) + D \cdot \frac{q-1}{2}}{\sqrt{\frac{n(q^2-1)}{24}(1+D^2)}}. \end{aligned}$$

Proof. Fix the index k . We consider the bivariate random variable $(\hat{c}_1[k], \hat{c}_2[k])$. By the multivariate central limit theorem, the distribution of $(\hat{c}_1[k], \hat{c}_2[k])$ can be approximated by a bivariate Gaussian distribution. Since the covariance of $\hat{c}_1[k]$ and $\hat{c}_2[k]$ is 0 by Lemma 2, the bivariate Gaussian distribution is reduced to two independent Gaussian distributions:

$$\hat{c}_1[k] \sim \mathcal{N}(\mu_1, \sigma_c^2), \quad \hat{c}_2[k] \sim \mathcal{N}(\mu_2, \sigma_c^2),$$

where

$$\begin{aligned} \sigma_c^2 &= \sum_{m=0}^{n-1} \text{Var}[c[m]] \cos^2\left(\frac{\pi(2k+1)m}{n}\right) = \sum_{m=0}^{n-1} \text{Var}[c[m]] \sin^2\left(\frac{\pi(2k+1)m}{n}\right) \\ &= \frac{n(q^2-1)}{24}, \end{aligned}$$

and

$$\mu_1 = \frac{q-1}{2} \sum_{m=0}^{n-1} \cos\left(\frac{\pi(2k+1)m}{n}\right), \quad \mu_2 = \frac{q-1}{2} \sum_{m=0}^{n-1} \sin\left(\frac{\pi(2k+1)m}{n}\right)$$

Let $\theta = \frac{\pi(2k+1)}{n}$. We have $\sum_{m=0}^{n-1} \exp(i\theta m) = \frac{1-\exp(i\theta n)}{1-\exp(i\theta)} = \frac{2}{1-\exp(i\theta)}$. Moreover,

$$\frac{1}{1-\exp(i\theta)} = \frac{1}{2} + \frac{i}{2} \frac{\sin \theta}{1-\cos \theta} = \frac{1}{2} + \frac{i}{2} \cot\left(\frac{\theta}{2}\right).$$

Therefore, by matching the real and imaginary parts, we obtain:

$$\mu_1 = \frac{q-1}{2}, \quad \mu_2 = \frac{q-1}{2} \cot\left(\frac{\pi(2k+1)}{2n}\right).$$

Now, let $X = \hat{c}_1[k]$ and $Y = \hat{c}_2[k]$. We first compute the probability that $|X| > D|Y|$, or equivalently, $X - DY > 0 \wedge X + DY > 0$ or $X - DY < 0 \wedge X + DY < 0$. Let $Z_1 = X - DY$ and $Z_2 = X + DY$. The joint distribution of Z_1 and Z_2 is also a bivariate Gaussian distribution. We have

$$\begin{aligned} \mathbb{E}[Z_1] &= \mu_1 - D\mu_2, & \mathbb{E}[Z_2] &= \mu_1 + D\mu_2 \\ \text{Var}[Z_1] &= \text{Var}[Z_2] = \text{Var}[X] + D^2\text{Var}[Y] = (1+D^2)\sigma_c^2, \\ \text{Cov}[Z_1, Z_2] &= \text{Var}(X) - D^2\text{Var}(Y) = (1-D^2)\sigma_c^2 \end{aligned}$$

Our goal is to compute $\Pr[Z_1 > 0 \wedge Z_2 > 0] + \Pr[Z_1 < 0 \wedge Z_2 < 0]$. By standardizing Z_1 and Z_2 , we define $Z'_1 = \frac{Z_1 - \mathbb{E}[Z_1]}{\sqrt{\text{Var}[Z_1]}}$ and $Z'_2 = \frac{Z_2 - \mathbb{E}[Z_2]}{\sqrt{\text{Var}[Z_2]}}$. The joint distribution of Z'_1 and Z'_2 is a bivariate Gaussian distribution with mean 0, variance 1, and covariance

$$\rho = \frac{\text{Cov}[Z_1, Z_2]}{\sqrt{\text{Var}[Z_1] \cdot \text{Var}[Z_2]}} = \frac{1-D^2}{1+D^2}.$$

Define $\mu'_1 = \frac{\mathbb{E}[Z_1]}{\sqrt{\text{Var}[Z_1]}}$ and $\mu'_2 = \frac{\mathbb{E}[Z_2]}{\sqrt{\text{Var}[Z_2]}}$. We have

$$\begin{aligned} \Pr[Z_1 > 0 \wedge Z_2 > 0] &= \Pr[Z'_1 > -\mu'_1 \wedge Z'_2 > -\mu'_2] = \Pr[Z'_1 < \mu'_1 \wedge Z'_2 < \mu'_2] \\ \Pr[Z_1 < 0 \wedge Z_2 < 0] &= \Pr[Z'_1 < -\mu'_1 \wedge Z'_2 < -\mu'_2] \end{aligned}$$

Let ϕ_ρ be the cumulative distribution function of the bivariate Gaussian distribution with mean 0, variance 1, and covariance ρ . The desired probability is $\phi_\rho(\mu'_1, \mu'_2) + \phi_\rho(-\mu'_1, -\mu'_2)$.

Following the same analysis, we can also compute the probability that $|Y| > D|X|$. Let $\mu''_1 = \frac{\mu_2 - D\mu_1}{\sqrt{(1+D^2)\sigma_c^2}}$ and $\mu''_2 = \frac{\mu_2 + D\mu_1}{\sqrt{(1+D^2)\sigma_c^2}}$. The probability that $|Y| > D|X|$ is $\phi_\rho(\mu''_1, \mu''_2) + \phi_\rho(-\mu''_1, -\mu''_2)$. Note that the two events $|X| > D|Y|$ and $|Y| > D|X|$ are mutually exclusive because $D \geq 1$.

□

We similarly evaluated Theorem 2 over 100,000 randomly generated hash polynomials. The empirical probabilities closely match the theoretical estimates, with a relative error of at most 10% for $D \leq 176$ and 20% for $D \leq 704$ across all coefficients. For larger D , the relative error increases as the event $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ becomes extremely rare, resulting in fewer observations among the sampled polynomials.

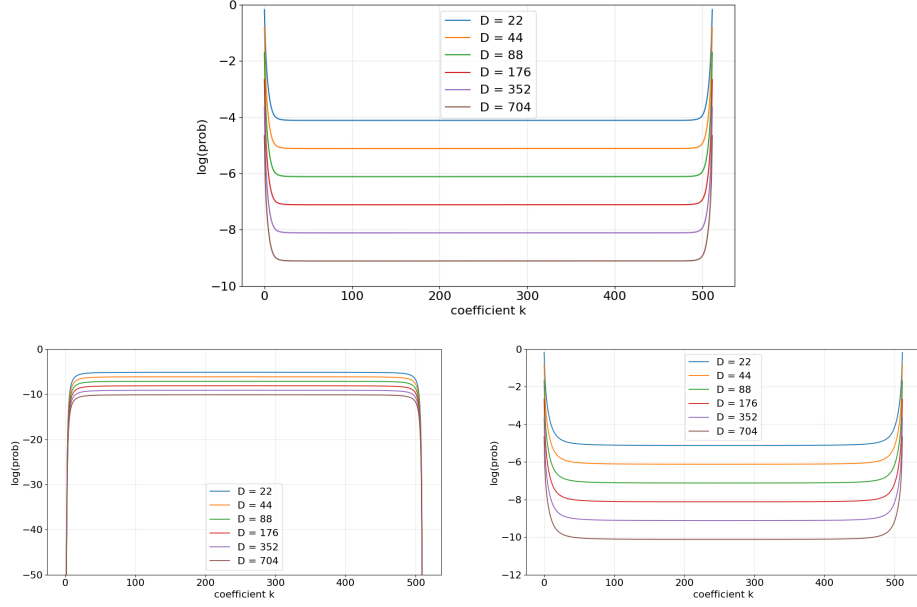


Fig. 1: Logarithm of probability that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ (top), probability that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ (bottom left), and probability that $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ (bottom right) for each coefficient index k when $n = 512$ and $D \in 11 \cdot \{2, 4, \dots, 64\}$.

Note that the probability established in Theorem 2 varies for different indices k . Moreover, we can split the probability into two disjoint cases:

$$\begin{aligned} p_{12}(n, D, k) &:= \phi_\rho(\mu'_1, \mu'_2) + \phi_\rho(-\mu'_1, -\mu'_2) \approx \Pr[|\hat{c}_1[k]| > D|\hat{c}_2[k]|], \\ p_{21}(n, D, k) &:= \phi_\rho(\mu''_1, \mu''_2) + \phi_\rho(-\mu''_1, -\mu''_2) \approx \Pr[|\hat{c}_2[k]| > D|\hat{c}_1[k]|]. \end{aligned}$$

In Fig. 1, we plot the total probability that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ (i.e., $p_{12}(n, D, k) + p_{21}(n, D, k)$), as well as the probabilities for $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ (i.e., $p_{12}(n, D, k)$) and $|\hat{c}_2[k]| > D|\hat{c}_1[k]|$ (i.e., $p_{21}(n, D, k)$), for each index k when $n = 512$. These plots are generated for $D \in B \cdot \{2, 4, \dots, 64\}$, with B fixed to 11. As illustrated in the first subfigure, all probability curves are strictly above 2^{-10} across all coefficients. Consequently, one can discover an effective c in fewer than 2^{10} randomly generated trials on average. However, the second subfigure reveals that the probability p_{12} , where $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$, is extremely small for indices $k < 8$. Therefore, it is computationally infeasible to find an effective c such that $|\hat{c}_1[k]| > D|\hat{c}_2[k]|$ for these specific indices, which justifies the need for enumerating the sign bits for them, as described in Section 3.5.

While we only present the results for $n = 512$, we note that the distributions for $n = 1024$ exhibit very similar behavior.

5 Experimental Evaluations

To validate the effectiveness of our proposed attack, we implemented the masked preimage computation of FALCON in [5] in C.³ Specifically, we implemented the masked complex multiplication using the masked floating-point multiplications and additions over two shares and recombined the shares of the output complex products. We compiled the program using `arm-none-eabi-gcc 10.3.1` and utilized ELMO [20] to record the power traces of the above process. ELMO is a cycle-accurate leakage simulator that estimates the instruction-level power consumption of an ARM Cortex-M0 processor. It is widely utilized in the literature and has previously been employed to evaluate side-channel vulnerabilities in the standardized post-quantum digital signature scheme ML-DSA [22].

Throughout this section, we fix the parameters $B = 11$ and $h = 10$. We conducted our experiments for $n = 512$, while the attack results for $n = 1024$ are similar since the theoretical distributions of the coefficients in the FFT domain remain the same ($\hat{f}_1[k], \hat{f}_2[k] \sim \mathcal{N}(0, 0.585 \cdot \sqrt{q})$) based on our analysis in Section 4. Because the trace filtering process of the polynomials c for the non-chosen-message attack depends on the degree n , we also present the estimated number of traces for $n = 1024$ in that scenario.

For each experiment, we randomly generated a secret polynomial f using the sampler algorithm specified in [24]. We picked $D \in B \cdot \{4, 16, 64\}$ and generated random polynomials c whose coefficients were uniformly distributed in $[0, q)$. Each candidate c was then evaluated to check whether $|\hat{c}_1[k]| > D \cdot |\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D \cdot |\hat{c}_1[k]|$ held for each index k . The generation process continued until we accumulated N_D qualifying polynomials for all the coefficients. We also generated N_e random polynomials c used for the exponent recovery. We then executed the masked complex multiplication for each generated polynomial and the secret key and collected the power traces using ELMO. Finally, we applied the attack in Section 3 to sequentially recover the sign bits, mantissas, and exponents for each coefficient.

To leverage the joint probability distribution of the real and imaginary parts during the exponent recovery, we employed the following heuristic. First, we identified the top five candidate exponent pairs that yielded the highest correlation values during the CPA. Next, we applied a maximum successive difference strategy to find the largest gap between the sorted correlation values. Finally, among the candidate pairs whose correlation scores fell above this gap, we determined the recovered exponents by selecting the pair with the highest theoretical joint probability (Equation 2).

For an index $k \geq 8$, we define a coefficient $\hat{f}_1[k]$ or $\hat{f}_2[k]$ to be successfully recovered by the following:

- If $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} \in [\frac{1}{B}, B]$, both the sign bits and exponents must be correctly recovered, and the relative error of the recovered mantissas must be strictly less than 2^{-5} . As demonstrated in Table 2, this guarantees an overwhelming probability of a complete recovery in the coefficient domain.

³ Our source code is publicly available at <https://github.com/kengyuchen/masked-falcon-power-analysis>.

- If $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > B$, then $\hat{f}_1[k]$ is considered successfully recovered if its sign bit and exponent are correct, and the relative error of its mantissa is less than 2^{-5} . We omit $\hat{f}_2[k]$ in this case.
- If $\frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < \frac{1}{B}$, then $\hat{f}_2[k]$ is considered successfully recovered if its sign bit and the exponent are correct, and the relative error of its mantissa is less than 2^{-5} . We omit $\hat{f}_1[k]$ in this case.

For a small index $k < 8$, we define a successful recovery similarly while assuming the sign bits are always correctly recovered. The overall success rate is defined as the number of successfully recovered coefficient parts divided by the total number of effective real and imaginary parts; that is, $512 - \#\{k : \frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} > B \vee \frac{|\hat{f}_1[k]|}{|\hat{f}_2[k]|} < \frac{1}{B}\}$.

Fig. 2 illustrates the relationship between the success rate and the number of traces, assuming $N_D = N_e$. We achieve a success rate of 98.3% for $D = 704$ when using 4,000 ($N_D + N_e$) traces, and 96.9% for $D = 176$ using the same number of traces. In contrast, the success rate for $D = 44$ reaches only 79.4%. Our experiment confirms that a larger D leads to a higher success rate, consistent with our analysis since a lower ratio $\frac{B}{D}$ ensures that the true intermediate complex products $\hat{t}_1[k]$ and $\hat{t}_2[k]$ more closely approximate the target floating-point products $\hat{c}_1[k] \cdot \hat{f}_1[k]$, $\hat{c}_1[k] \cdot \hat{f}_2[k]$, $\hat{c}_2[k] \cdot \hat{f}_1[k]$, and $-\hat{c}_2[k] \cdot \hat{f}_2[k]$. Nevertheless, a 98.3% success rate is insufficient for complete recovery of the secret key, as even a single incorrectly recovered coefficient part can affect all coefficients in the coefficient domain. Based on the experimental results in [6], full key recovery in the coefficient domain requires around a 99.8% success rate and substantially more traces. We capped our experiments at 4,000 traces due to the long signing time and large trace size of FALCON with masked preimage computation.

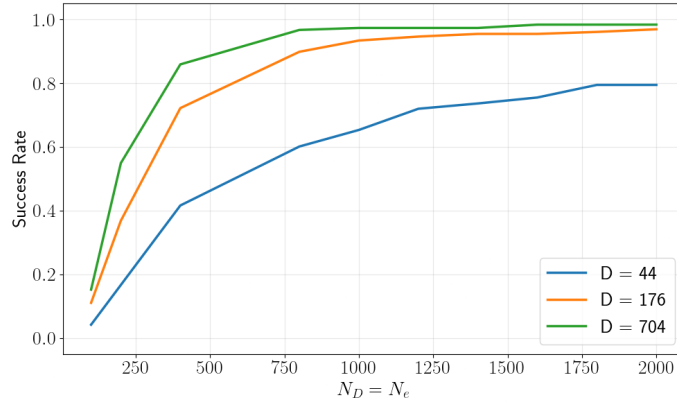


Fig. 2: The success rate for different values of $D \in B \cdot \{4, 16, 64\}$ relative to the number of traces $N_D = N_e$.

Next, we examine the practical costs of the non-chosen-message attack. Fig. 3 illustrates the relationship between the success rate and the *total* number N_{total} of generated polynomials, which include all the discarded polynomials that do not satisfy the condition $|\hat{c}_1[k]| > D \cdot |\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D \cdot |\hat{c}_1[k]|$ for some indices k and N_e polynomials used for exponent recovery. For example, discovering a qualifying c is easier for $D = 44$ than for $D = 176$ or 704 and thus requires fewer raw traces. This metric reflects the real-world trace complexity for a non-chosen-message attack, where the adversary cannot control the input message or random salt.

While the chosen-message attack achieves a higher success rate with fewer traces when $D = 704$, as previously shown in Fig. 2, it requires much more traces in Fig. 3. This is because the probability of randomly encountering a useful polynomial c for $D = 704$ is much smaller than for $D = 176$ and 44 . However, while generating useful polynomials c for $D = 44$ is the easiest, the attack’s precision degrades, and the success rate does not seem to converge to 100% within 10^6 traces. Consequently, $D = 176$ emerges as the optimal parameter for a non-chosen-message attack, which achieves a 96.9% success rate with 298,062 traces. Unlike the chosen-message scenario where a higher D strictly improves performance, the non-chosen-message attack requires balancing trace filtering costs against the precision of the approximation.

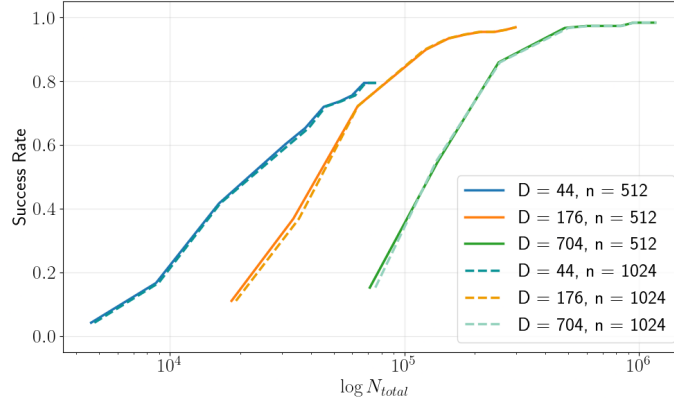


Fig. 3: The success rate for different values of $D \in B \cdot \{4, 16, 64\}$ relative to the total number of traces N_{total} . The dashed lines indicate the estimated success rate for $n = 1024$.

We also estimated the number of traces required for $n = 1024$ in the non-chosen-message attack. Specifically, we generated random polynomials c of degree 1023 and counted how many iterations were necessary to accumulate N_D polynomials satisfying the condition $\hat{c}_1[k] > D \cdot \hat{c}_2[k]$ or $\hat{c}_2[k] > D \cdot \hat{c}_1[k]$ for all indices $k \in \{0, \dots, 511\}$ ⁴. We plotted these projections by assuming that the success rate for a given N_D remains the same between $n = 512$ and $n = 1024$. The results, depicted by the dashed lines in Fig. 3,

⁴ Note that only half of the coefficients are required due to the conjugate symmetry in the FFT domain.

reveal no significant difference in trace complexity between the two security levels. This is consistent with our theoretical analysis. The overall trace number requirement is bottlenecked primarily by the specific index k that yields the smallest probability derived in Theorem 2. For $D = 176$, this minimum probability is roughly 0.0072339 for $n = 512$ and 0.0072341 for $n = 1024$, which are nearly identical.

Finally, we compare the individual success rates for recovering the sign bits, exponents, and mantissas for $D = 176$ in Fig. 4. The success criteria for the sign bits and mantissas remain the same as defined previously. However, to isolate the exponent-recovery metric from the performance of the recovery approaches for sign bits and mantissas, we define its success rate by conducting the attack using the ground-truth sign bits and the $h + 1 = 11$ most significant bits of the ground-truth mantissas. Hence, the performance of the exponent recovery is independent of D . As shown in Fig. 4, the success rate for the sign bits reaches 100% with merely 400 traces. In contrast, the success rates for the exponents and mantissas are lower and reach 97.9% and 99.2%, respectively, at 2,000 traces. Because these two curves exhibit similar trends, recovering the exponents and mantissas constitutes the empirical bottleneck of the attack and requires comparable numbers of traces (i.e., $N_D \approx N_e$). Nevertheless, we emphasize that generating N_D useful polynomials c in the context of a non-chosen-message attack is far more demanding than generating N_e random polynomials for exponent recovery.

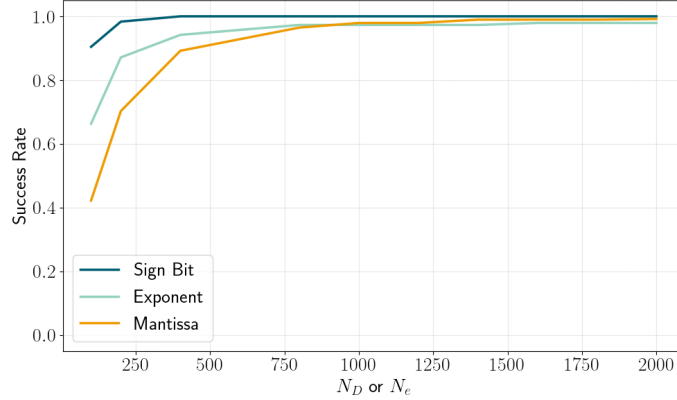


Fig. 4: The individual success rates for sign bit, exponent, and mantissa recovery at $D = 176$ relative to the number of traces N_D (for sign bit and mantissa) or N_e (for exponent).

6 Countermeasures

In this section, we discuss potential countermeasures to mitigate our proposed attack.

The chosen-message attack can be thwarted simply by preventing the leakage of the random salt before signing. This can be achieved either by using a true random num-

ber generator (TRNG) or a cryptographically secure pseudo-random number generator (PRNG) with a secret seed. To resist the non-chosen-message attack, we restrict the allowable range of the ratio $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|}$ for each index k . A naive approach would be to reject any generated hash polynomial c if there exists a single index k such that $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$ or $< \frac{1}{D}$. However, enforcing this global condition would require an impractically large number of rejection sampling trials to generate a valid polynomial and significantly increase the signing time. Therefore, we propose a relaxed countermeasure that checks this condition on only a specific subset of the coefficients.

Table 3 presents the empirical probability that no index k within the range $[t, \frac{n}{2} - t)$ satisfies $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$ or $< \frac{1}{D}$ for $n = 512$ and 1024 . These probabilities are estimated by generating 10^4 samples of c for each trial. The parameter t determines the number of coefficients left vulnerable to the non-chosen-message attack, and D determines the required precision of the CPA. For instance, when $n = 512$, $t = 240$, and $D = 64$, the probability that no index $k \in [240, 256)$ satisfies $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > 64$ or $< \frac{1}{64}$ is approximately 61.4%. Consequently, one can expect to generate a compliant, attack-resistant polynomial c for the protected range $[240, 256)$ within 5 trials with a probability over 99%.

Table 3: The empirical probability of generating a polynomial c such that no index k in the range $[t, \frac{n}{2} - t)$ satisfies $\frac{|\hat{c}_1[k]|}{|\hat{c}_2[k]|} > D$ or $< \frac{1}{D}$. Results are shown for $n \in \{512, 1024\}$ across various values of t and D . The probabilities are estimated by generating 10^4 samples of c per trial.

t		144	160	176	192	208	224	240
$n = 512$	$D = 32$	0.6%	1.3%	2.5%	4.9%	9.9%	19.2%	37.2%
	$D = 64$	8.2%	11.65%	17.1%	22.6%	32.3%	44.8%	61.4%
	$D = 128$	29.7%	35.7%	40.2%	47.2%	57.2%	67.4%	79.6%
t		400	416	432	448	464	480	496
$n = 1024$	$D = 32$	0.6%	1.2%	2.2%	4.5%	8.6%	16.3%	31.7%
	$D = 64$	8.2%	11.1%	15.5%	20.9%	30.2%	41.5%	57.0%
	$D = 128$	28.3%	33.4%	39.1%	46.9%	55.8%	65.4%	76.2%

As demonstrated by our experiments in Section 5, the required number of traces for the non-chosen-message attack is already exceptionally high. Table 3 shows that even if an adversary acquires an infinite number of traces, we can still protect a subset of coefficients by this rejection sampling strategy, assuming that the attack utilizing a small D always fails. The adversary would be forced to brute-force the real and imaginary parts of the protected coefficients in the range $[t, \frac{n}{2} - t)$. Assuming $h = 5$, each protected part requires $2 \times 16 \times 2^5 = 2^{10}$ guesses for sign bits in $\{0, 1\}$, 16 exponents in $[1016, 1031]$, and 2^5 mantissas in $2^{52} + 2^{52-5} \cdot [0, 2^5)$. However, we do not rule out the possibility of

a more sophisticated algebraic attack that can recover the full secret key using only the unprotected partial coefficients in the FFT domain.

Finally, a comprehensive masking scheme that covers the entire signing process can prevent both the chosen-and non-chosen-message attacks. A fully masked implementation of FALCON can be constructed by combining a masked preimage computation [5] and a masked Gaussian sampling [2]. However, implementing such a full masking scheme currently introduces severe computational overhead. According to the benchmarks in [2], a fully masked FALCON spends approximately 99% of its execution time on the subsequent Gaussian sampling stage, and the current implementation is still impractical for real-world deployment. Therefore, designing a more efficient masking scheme for Gaussian sampling remains a critical research problem to fully secure FALCON against our proposed attack without sacrificing reasonable signing performance.

7 Conclusion

In this work, we proposed a power analysis attack targeting a FALCON implementation protected by a masked preimage computation. Specifically, we developed a chosen-message attack that assumes the attacker can control the input messages and random salts. Our attack can naturally be extended to a non-chosen-message setting by filtering for power traces whose corresponding public hash polynomials c satisfy the condition $|\hat{c}_1[k]| > D \cdot |\hat{c}_2[k]|$ or $|\hat{c}_2[k]| > D \cdot |\hat{c}_1[k]|$ for some indices k . To resist the chosen-message attack, implementations should avoid using predictable randomness. In cases where the random salts cannot be hidden before signing, or to resist the non-chosen-message attack, we proposed a low-overhead rejection sampling countermeasure that enforces these ratio bounds on a predefined subset of coefficients. Another solution would be to employ a full masking scheme over the entire signing process, which is still practically infeasible due to the extreme computational overhead of masking the subsequent Gaussian sampling stage.

We conclude by noting several limitations of our study and directions for future work. First, we only conducted our experiments on the ELMO leakage simulator. While ELMO provides highly accurate instruction-level power estimations, validating the empirical trace complexity on a real device remains a critical next step, particularly on architectures other than the ARM Cortex-M0. Second, due to the long signing times and massive trace sizes of the masked preimage computation, our experiments were capped at 4,000 traces for each coefficient across three distinct values of D . Profiling the exact trace requirements across a wider spectrum of parameter values requires further investigation. Finally, our proposed rejection sampling countermeasure assumes that no efficient attacks can recover the secret key using only partial, unprotected coefficients in the FFT domain. Since the feasibility of such an algebraic attack remains an open question, designing more lightweight and robust countermeasures is an important area for future research.

Acknowledgments. We thank the anonymous reviewers of CHES 2027 for their constructive comments and suggestions, which helped us improve the presentation and clarity of this work.

References

1. Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G., Bos, J., Dion, A., Lacan, J., Robert, J.M., Veron, P.: HQC. Tech. rep., National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions>
2. Berthet, P.A., Paillet, J., Tavernier, C., Bossuet, L., Colombier, B.: Masked computation of the floor function and its application to the FALCON signature. *IACR Communications in Cryptology (CiC)* **1**(4), 9 (2024). <https://doi.org/10.62056/ay73zl7s>
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.J. (eds.) *Cryptographic Hardware and Embedded Systems – CHES 2004*. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer Berlin Heidelberg, Germany, Cambridge, Massachusetts, USA (Aug 11–13, 2004). https://doi.org/10.1007/978-3-540-28632-5_2
4. Bruinderink, L.G., Hülsing, A., Lange, T., Yarom, Y.: Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In: Gierlichs, B., Poschmann, A.Y. (eds.) *Cryptographic Hardware and Embedded Systems – CHES 2016*. Lecture Notes in Computer Science, vol. 9813, pp. 323–345. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–19, 2016). https://doi.org/10.1007/978-3-662-53140-2_16
5. Chen, K.Y., Chen, J.P.: 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). <https://doi.org/10.46586/tches.v2024.i2.276-303>
6. Chen, K.Y., Ching, M.Q., Chen, J.P., Yang, B.Y.: When masking multiplication isn’t enough: Exploiting floating-point leakage in falcon’s pre-image computation. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2026**(2), 686–708 (Apr 2026). <https://doi.org/10.46586/tches.v2026.i2.686-708>, <https://tches.iacr.org/index.php/TCHES/article/view/12905>
7. Chen, S., Shao, H., Ming, J., Liu, Y., Gao, Y., Zhou, Y.: Serp-sca: A strengthened side-channel attack framework for falcon signature. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **45**(4), 2028–2040 (2026). <https://doi.org/10.1109/TCAD.2025.3608056>
8. Ducas, L., Prest, T.: Fast fourier orthogonalization. In: *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*. pp. 191–198 (2016)
9. Espitau, T., Fouque, P.A., Gérard, B., Tibouchi, M.: Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) *ACM CCS 2017: 24th Conference on Computer and Communications Security*. pp. 1857–1874. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017). <https://doi.org/10.1145/3133956.3134028>
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) *40th Annual ACM Symposium on Theory of Computing*. pp. 197–206. ACM Press, Victoria, BC, Canada (May 17–20, 2008). <https://doi.org/10.1145/1374376.1374407>
11. Guerreau, M., Martinelli, A., Ricosset, T., Rossi, M.: The hidden parallelepiped is back again: Power analysis attacks on falcon. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2022**(3), 141–164 (2022). <https://doi.org/10.46586/tches.v2022.i3.141-164>
12. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: *Third Algorithmic Number Theory Symposium (ANTS)*. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (Jun 1998)

13. Howe, J., Prest, T., Ricosset, T., Rossi, M.: Isochronous gaussian sampling: From inception to implementation. In: Ding, J., Tillich, J.P. (eds.) *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*. pp. 53–71. Springer, Cham, Switzerland, Paris, France (Apr 15–17, 2020). https://doi.org/10.1007/978-3-030-44223-1_4
14. Hülsing, A., Bernstein, D.J., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., Kampanakis, P., Kölbl, S., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Aumasson, J.P., Westerbaan, B., Beullens, W.: SPHINCS+. Tech. rep., National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
15. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ecdsa). *International journal of information security* **1**(1), 36–63 (2001)
16. Karabulut, E., Aysu, A.: Falcon down: Breaking falcon post-quantum signature scheme through side-channel attacks. In: *Proceedings of the 58th Annual ACM/IEEE Design Automation Conference*. p. 691–696. DAC '21, IEEE Press (2022). <https://doi.org/10.1109/DAC18074.2021.9586131>, <https://doi.org/10.1109/DAC18074.2021.9586131>
17. Karabulut, E., Aysu, A.: Masking FALCON's floating-point multiplication in hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2024**(4), 483–508 (2024). <https://doi.org/10.46586/tches.v2024.i4.483-508>
18. Lin, X., Zhang, S., Yu, Y., Wang, W., You, Q., Xu, X., Wang, X.: Thorough power analysis on falcon gaussian samplers and practical countermeasure. In: Jager, T., Pan, J. (eds.) *PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part I*. Lecture Notes in Computer Science, vol. 15674, pp. 229–258. Springer, Cham, Switzerland, Røros, Norway (May 12–15, 2025). https://doi.org/10.1007/978-3-031-91820-9_8
19. Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D., Bai, S.: CRYSTALS-DILITHIUM. Tech. rep., National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
20. McCann, D., Whitnall, C., Oswald, E.: ELMO: Emulating leaks for the ARM Cortex-M0 without access to a side channel lab. *Cryptology ePrint Archive*, Report 2016/517 (2016), <https://eprint.iacr.org/2016/517>
21. McCarthy, S., Howe, J., Smyth, N., Brannigan, S., O'Neill, M.: BEARZ attack FALCON: Implementation attacks with countermeasures on the FALCON signature scheme. *Cryptology ePrint Archive*, Report 2019/478 (2019), <https://eprint.iacr.org/2019/478>
22. Migliore, V., Gérard, B., Tibouchi, M., Fouque, P.A.: Masking Dilithium - efficient implementation and side-channel evaluation. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) *ACNS 2019: 17th International Conference on Applied Cryptography and Network Security*. Lecture Notes in Computer Science, vol. 11464, pp. 344–362. Springer, Cham, Switzerland, Bogota, Colombia (Jun 5–7, 2019). https://doi.org/10.1007/978-3-030-21568-2_17
23. Pessl, P., Bruinderink, L.G., Yarom, Y.: To BLISS-B or not to be: Attacking strongSwan's implementation of post-quantum signatures. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) *ACM CCS 2017: 24th Conference on Computer and Communications Security*. pp. 1843–1855. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017). <https://doi.org/10.1145/3133956.3134023>
24. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON. Tech. rep., National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
25. Qiu, J., Aysu, A.: SHIFT SNARE: Uncovering secret keys in FALCON via single-trace analysis. *Cryptology ePrint Archive*, Report 2025/146 (2025), <https://eprint.iacr.org/2025/146>

26. Qiu, J., Aysu, A.: Ntt-peel: Bit shift side-channel in falcon's number theoretic transform. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2026**(2), 709–735 (Apr 2026). <https://doi.org/10.46586/tches.v2026.i2.709-735>, <https://tches.iacr.org/index.php/TCHES/article/view/12906>
27. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery* **21**(2), 120–126 (Feb 1978). <https://doi.org/10.1145/359340.359342>
28. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D., Ding, J.: CRYSTALS-KYBER. Tech. rep., National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
29. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**(5), 1484–1509 (oct 1997). <https://doi.org/10.1137/s0097539795293172>, <https://doi.org/10.1137%2Fs0097539795293172>
30. of Standards, N.I., Technology.: Post-quantum cryptography standardization, <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>
31. Zhang, S., Lin, X., Yu, Y., Wang, W.: Improved power analysis attacks on falcon. In: Hazay, C., Stam, M. (eds.) *Advances in Cryptology – EUROCRYPT 2023, Part IV. Lecture Notes in Computer Science*, vol. 14007, pp. 565–595. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30634-1_19
32. Zhao, R.K., Steinfeld, R., Sakzad, A.: Facct: Fast, compact, and constant-time discrete gaussian sampler over integers. *IEEE Trans. Comput.* **69**(1), 126–137 (Jan 2020). <https://doi.org/10.1109/TC.2019.2940949>, <https://doi.org/10.1109/TC.2019.2940949>