Improved Power Analysis Attacks on Falcon

Shiduo Zhang[1], Xiuhan Lin[2], Yang Yu[3,4,5[0000−0003−0120−0648](✉)], and Weijia Wang[2[0000−0001−6982−2537]]

[1] Institute for Advanced Study, Tsinghua University, Beijing, China
zsd19@mails.tsinghua.edu.cn
[2] School of Cyber Science and Technology, Shandong University, Qingdao, China
xhlin@mail.sdu.edu.cn, wjwang@sdu.edu.cn
[3] BNRist, Tsinghua University, Beijing, China
yu-yang@mail.tsinghua.edu.cn
[4] Zhongguancun Laboratory, Beijing, China
[5] National Financial Cryptography Research Center, Beijing, China

Abstract. Falcon is one of the three post-quantum signature schemes selected for standardization by NIST. Due to its low bandwidth and high efficiency, Falcon is seen as an attractive option for quantum-safe embedded systems. In this work, we study Falcon's side-channel resistance by analysing its Gaussian samplers. Our results are mainly twofold.

The first result is an improved key recovery exploiting the leakage within the base sampler investigated by Guerreau et al. (CHES 2022). Instead of resorting to the fourth moment as in former parallelepiped-learning attacks, we work with the second order statistics covariance and use its spectral decomposition to recover the secret information. Our approach substantially reduces the requirement for measurements and computation resources: 220 000 traces is sufficient to recover the secret key of Falcon-512 within half an hour with a probability of $\approx 25\%$. As a comparison, even with 10^6 traces, the former attack still needs about 1000 hours CPU time of lattice reduction for a full key recovery. In addition, our approach is robust to inaccurate leakage classification, which is another advantage over parallelepiped-learning attacks.

Our second result is a practical power analysis targeting the integer Gaussian sampler of Falcon. The analysis relies on the leakage of random sign flip within the integer Gaussian sampling. This leakage was exposed in 2018 by Kim and Hong, but it is not considered in Falcon's implementation and unexploited for side-channel analysis until now. We identify the leakage within the reference implementation of Falcon on an ARM Cortex-M4 STM32F407IGT6 microprocessor. We also show that this single bit of leakage is in effect enough for practical key recovery: with 170 000 traces one can fully recover the key of Falcon-512 within half an hour. Furthermore, combining the sign leakage and the aforementioned leakage, one can recover the key with only 45 000 signature measurements in a short time.

As a by-product, we also extend our power analysis to Mitaka which is a recent variant of Falcon. The same leakages exist within the integer Gaussian samplers of Mitaka, and they can also be used to mount key recovery attacks. Nevertheless, the key recovery in Mitaka requires much

more traces than it does in Falcon, due to their different lattice Gaussian samplers.

1 Introduction

Recently, NIST announced the first post-quantum cryptography algorithms to be standardized. For digital signatures, two of the three selected algorithms are lattice-based: Dilithium [25] and Falcon [33], the third one is a hash-based signature scheme SPHINCS$^+$ [19]. In comparison, Dilithium and Falcon have better overall performance.

Dilithium and Falcon are constructed in two distinct frameworks. Dilithium uses "Fiat-Shamir with aborts" paradigm, developed by Lyubashevsky [23,24] and Falcon uses the hash-and-sign paradigm. Two schemes achieve acceptable overall performance for many use cases and also have their own advantages: Dilithium has a simpler implementation and more flexible parameter selections, while Falcon has a greatly smaller public key and signature sizes. For this, each of them would have potential applications in various situations and NIST eventually selected both schemes for standardization.

For a real-world deployed scheme, implementation security is of great importance. For insecure implementations, sensitive information may leak through side channels, e.g. execution time, power consumption, and electromagnetic emanations. These leakages may be exploited to mount devastating attacks that are the major threat to cryptographic embedded devices. The implementation security of Dilithium is relatively well-studied. The reference implementation of Dilithium is constant time, which eliminates side-channel vulnerabilities in former Fiat–Shamir lattice signatures [16,30,10,2,34]. Moreover, efficient masking of Dilithium at any order is proposed in [27], which protects Dilithium against stronger side-channel attacks.

In contrast, the implementation security of Falcon is intricate. Falcon follows the GPV framework [14] to prevent statistical attacks [28,7,37,9]. In the GPV signature scheme, signing requires Gaussian sampling which is a notorious target of side-channel attacks [16,10,21,13]. Furthermore, Falcon's sampling heavily relies on floating-point operations, which complicates the secure implementation. For the above reasons, while the implementation of Falcon is now secure against timing attacks [18,31], countermeasures against stronger side-channel attacks like power analysis remain a challenging open problem. The lack of side-channel protections provides an avenue for side-channel attacks. The first side-channel attack on Falcon is an electromagnetic attack presented by Karabulut and Aysu [20] that targets the floating-point multiplications within Falcon's Fast Fourier Transform. The Karabulut-Aysu attack is substantially improved later [17]: 5 000 power traces is sufficient for a full key recovery of Falcon-512 on ChipWhisperer. Also in [17], Guerreau et al. proposed another practical power analysis on Falcon based on a different side-channel leakage. It exploits the power leakage within the base Gaussian sampler to filter signatures in a secret-dependent region, and completes the key recovery by applying

parallelepiped-learning attacks [28,7]. As the very first side-channel attack targeting Falcon's Gaussian sampling, this attack is rather expensive in terms of computation resources and measurements: practical key recovery needs millions of traces.

Our contributions. In this work, we develop several power analysis attacks on Falcon. Our contributions are mainly twofold.

We substantially improve the key recovery in the power analysis of Falcon's base sampler of [17]. The exploited leakage, called *half Gaussian leakage*, filters signatures in the slice $\{\mathbf{v} : |\langle \mathbf{v}, \mathbf{b} \rangle| \leq \|\mathbf{b}\|^2\}$ where $\mathbf{b}$ is the secret key. The key recovery is in essence to learn $\mathbf{b}$ from this secret-dependent slice, which was done by parallelepiped-learning attacks [28,7] in [17]. Our main idea stems from the observation that the projection of filtered signatures tends to be unusually short in the direction of the slice. We therefore proceed to learn the direction and the width of the slice, i.e. $\frac{\mathbf{b}}{\|\mathbf{b}\|}$ and $\|\mathbf{b}\|$, through the spectral decomposition of the covariance of filtered signatures. Compared with the fourth moment considered in previous parallelepiped-learning attacks, covariance, as a lower order statistic, allows smaller measure errors and thus leads to a more accurate approximation of $\mathbf{b}$. As a result, our new key recovery algorithm significantly lowers the requirement of measurements and computation resources: 220 000 traces is sufficient for our algorithm to recover the secret key within half an hour with a probability of $\approx 25\%$; by contrast, even with 10^6 traces, the key recovery of [17] still requires around 1000 hours CPU time of lattice reduction. Moreover, the effectiveness of our key recovery relies on the condition number and the measurement of the covariance of filtered signatures, thus our algorithm can even work with inaccurate leakage classification, say with accuracy 55%. In comparison, parallelepiped-learning attacks do not work well if the domain of filtered signatures has no clear boundary, which makes previous analysis reliant on accurate leakage classification. Therefore our result validates half Gaussian leakage to be a threat more serious than previously imagined.

We also propose a new power analysis of Falcon's integer sampler that is at the layer[6] above the base sampler investigated in [17]. To cope with variable parameters, Falcon's integer sampler first transforms a sample z^+ from fixed half integer Gaussian into a bimodal half Gaussian sample $z = b + (2b-1)z^+$ with a random $b \in \{0, 1\}$ and then accepts z with corresponding probability. The random bit b can be retrieved via simple power analysis as shown in [21]. This leakage, called *sign leakage*, filters signatures in the halfspace $\{\mathbf{v} : \langle \mathbf{v}, \mathbf{b} \rangle \geq 0\}$. The aforementioned statistical attack can be directly applied to the case of halfspace in the same spirit, but the approximate direction, denoted $\mathbf{u}$, of $\mathbf{b}$ is less accurate. To refine the key recovery, we use the rough approximation $\mathbf{u}$ to filter signatures in the slice $\{\mathbf{v} : |\langle \mathbf{v}, \mathbf{b} \rangle| \leq b\}$ with a well-chosen b. Applying the former key recovery again, we can recover the key given 170 000 of traces. Moreover, the sign leakage can be combined with the previous half Gaussian

[6] There are 3 layers of Gaussian samplers in Falcon: lattice Gaussian sampler - integer Gaussian sampler - base sampler.

leakage and then filter a thinner slice. Focusing on the thinner slice, the statistical attack can be even more effective: only 45 000 traces is sufficient for a direct key recovery with a probability of $\approx 25\%$. Additionally, through simple power analysis, we practically identify the sign leakage in the reference implementation of Falcon. Furthermore, we also propose an efficient countermeasure to mitigate this leakage, which (based on our settings of leakage acquisition) decreases the sign classification accuracy to $\approx 52\%$ from almost 100%.

As an additional contribution, we extend the power analysis on Falcon to its recent variant Mitaka. Since Mitaka uses Falcon's integer sampler, both half Gaussian leakage and sign leakage exist within its reference implementation in the same manner. Different from Falcon, Mitaka uses the hybrid sampler [32] for lattice Gaussian sampling, in which the output is the sum of two samples from two ellipsoid Gaussians. This makes the domain of filtered signatures the ambient space rather than a slice or a half-space. Nevertheless, the distribution of filtered signatures is still secret-dependent, thus our aforementioned approach is able to recover the key with more traces.

Roadmap. We start in Section 2 with preliminary material. Section 3 introduces the Gaussian samplers of Falcon that are the targets of our side-channel attacks. We present in Section 4 an improved key recovery using the same side-channel leakage studied in [17]. Section 5 exhibits a new power analysis attack targeting the integer Gaussian sampling of Falcon and a countermeasure against this attack is provided. We extend the above power analysis to Mitaka in Section 6 and conclude in Section 7.

2 Preliminaries

We use bold lowercase (resp. uppercase) letters for vectors (resp. matrices). By convention, vectors are in column form. For a distribution D, we write $z \leftarrow D$ when the random variable z sampled from D and denote by $D(x)$ the probability of $z = x$. We denote by $z \sim D$ a random variable distributed as D. Let $\mathbb{E}[z]$ be the expectation of random variable z and $\mathbf{var}[z]$ be the variance. For a random vector $\mathbf{z} = (z_0, \cdots, z_{n-1}) \in \mathbb{R}^n$, its covariance is

$$\mathbf{Cov}[\mathbf{z}] = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n-1,0} & c_{n-1,1} & \cdots & c_{n-1,n-1} \end{pmatrix} \text{ where } c_{i,j} = \mathbb{E}[z_i z_j] - \mathbb{E}[z_i]\,\mathbb{E}[z_j].$$

For a real-valued function f and a countable set S, we write $f(S) = \sum_{x \in S} f(x)$ assuming this sum is absolutely convergent. Let $\mathcal{N}(\mu, \sigma^2)$ be the normal distribution of the mean μ and the standard deviation σ.

2.1 Linear Algebra and Lattices

Let b_i (resp. $\mathbf{b}_i$) denote the i-th coordinate (resp. column) of $\mathbf{b}$ (resp. $\mathbf{B}$). Given $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, their inner product is $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=0}^{n-1} u_i v_i$. When $\langle \mathbf{u}, \mathbf{v} \rangle = 0$, we call $\mathbf{u}$

and $\mathbf{v}$ are orthogonal. Let $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$ be the ℓ_2-norm of $\mathbf{v}$, $\|\mathbf{v}\|_1 = \sum_i |v_i|$ be the ℓ_1-norm and $\|\mathbf{v}\|_\infty = \max_i\{|v_i|\}$ be the ℓ_∞-norm. Let $\mathbf{I}$ denote the identity matrix. For $\mathbf{H} \in \mathbb{R}^{n \times m}$, we let span($\mathbf{H}$) be the linear span of the rows of $\mathbf{H}$.

A symmetric matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is positive definite, denoted $\mathbf{\Sigma} > 0$, if $\mathbf{x}^t \mathbf{\Sigma} \mathbf{x} > 0$ for all nonzero $\mathbf{x} \in \mathbb{R}^n$. The spectral decomposition of a positive definite matrix $\mathbf{\Sigma}$ is $\mathbf{\Sigma} = \mathbf{Q}\mathbf{D}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{D}\mathbf{Q}^t$ where $\mathbf{D} = \text{diag}(\lambda_1, \cdots, \lambda_n)$ with $\lambda_1 \leq \cdots \leq \lambda_n$ being the eigenvalues of $\mathbf{\Sigma}$ and $\mathbf{Q}$ is an orthogonal matrix whose i-th column is the eigenvector corresponding to λ_i.

Let $\mathbf{B} = (\mathbf{b}_0, \cdots, \mathbf{b}_{n-1}) \in \mathbb{R}^{m \times n}$ of rank n. The Gram-Schmidt Orthogonalization (GSO) of $\mathbf{B}$ is the unique matrix $\widetilde{\mathbf{B}} = \left(\widetilde{\mathbf{b}}_0, \cdots, \widetilde{\mathbf{b}}_{n-1}\right) \in \mathbb{R}^{m \times n}$ such that $\mathbf{B} = \widetilde{\mathbf{B}}\mathbf{U}$ where $\widetilde{\mathbf{b}_i}$'s are pairwise orthogonal and $\mathbf{U}$ is upper-triangular with 1 on its diagonal. Let $\|\mathbf{B}\|_{GS} = \max_i \|\widetilde{\mathbf{b}}_i\|$.

A lattice $\mathcal{L}$ is the set of all integer linear combinations of linearly independent vectors $\mathbf{b}_0, \cdots, \mathbf{b}_{n-1} \in \mathbb{R}^m$, i.e. $\mathcal{L} = \left\{ \sum_{i=0}^{n-1} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}$. We call $\mathbf{B} = (\mathbf{b}_0, \cdots, \mathbf{b}_{n-1})$ a basis and n the dimension of $\mathcal{L}$. Let $\mathcal{L}(\mathbf{B})$ denote the lattice generated by a basis $\mathbf{B}$.

2.2 Gaussian Distributions

Let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}\right)$ be the Gaussian function with center $\mathbf{c} \in \mathbb{R}^n$ and standard deviation σ. The discrete Gaussian over a lattice $\mathcal{L}$ with center $\mathbf{c}$ and standard deviation σ is defined by the probability function $D_{\mathcal{L},\sigma,\mathbf{c}}(\mathbf{v}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{v})}{\rho_{\sigma,\mathbf{c}}(\mathcal{L})}$ for any $\mathbf{v} \in \mathcal{L}$.

We call $D_{\mathbb{Z},\sigma,c}$ integer Gaussian that is of particular interest. It suffices to study the case where $c \in [0, 1)$, since $D_{\mathbb{Z},\sigma,c} = i + D_{\mathbb{Z},\sigma,c-i}$ for any $i \in \mathbb{Z}$. By restricting $D_{\mathbb{Z},\sigma,c}$ over $\mathbb{N}$, we get a half integer Gaussian $D^+_{\mathbb{Z},\sigma,c}$ satisfying $D^+_{\mathbb{Z},\sigma,c}(v) = \frac{\rho_{\sigma,c}(v)}{\rho_{\sigma,c}(\mathbb{N})}$ for any $v \in \mathbb{N}$.

2.3 NTRU

Typically, an NTRU-based scheme is defined over some polynomial ring $\mathcal{R}$ along with a modulus q. In this work, $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ with n a power-of-2. Let $\mathcal{K} = \mathbb{Q}[x]/(x^n + 1)$ and $\mathcal{K}_\mathbb{R} = \mathbb{R}[x]/(x^n + 1)$. The NTRU secret key consists of two short polynomials $f, g \in \mathcal{R}$ where f is invertible modulo q, and the public key is $h = f^{-1}g \bmod q$. The NTRU module determined by h is $\mathcal{L}_{NTRU} = \{(u, v) \in \mathcal{R}^2 \mid u + vh = 0 \bmod q\}$. From (f, g), by solving the NTRU equation $fG - gF = q$, one can compute $\mathbf{B}_{f,g} = \begin{pmatrix} g & G \\ -f & -F \end{pmatrix}$ a basis of $\mathcal{L}_{NTRU}$ that is called an NTRU trapdoor basis. When the context is clear, we simply denote $\mathbf{B}_{f,g}$ as $\mathbf{B}$. Elements in $\mathcal{R}$ are identified with their matrix of multiplication in a certain basis, thus the NTRU module is seen as a lattice of dimension $2n$ that is an NTRU lattice.

2.4 Falcon Signature Scheme

We now briefly describe the Falcon signature scheme. Some details that are unnecessary for understanding this work are omitted, and we refer to [33] for a complete description of Falcon.

Falcon is an instantiation of the GPV hash-and-sign framework [14] over NTRU lattices. The secret key of Falcon is an NTRU trapdoor basis $\mathbf{B}_{f,g}$ and the public key is $h = f^{-1}g \bmod q$. The secret polynomials f and g are drawn from $D_{\mathcal{R},\sigma,0}$ with $\sigma = 1.17\sqrt{\frac{q}{2n}}$ for nearly optimal parameters as per [6]. In addition, $\mathbf{B}_{f,g}$ is required to have a bounded Gram-Schmidt norm: $\|\mathbf{B}_{f,g}\|_{GS} \leq 1.17\sqrt{q}$. This work focuses on the parameters of Falcon-512 for NIST Level-I where $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$, $n = 512$ and $q = 12289$.

Following the GPV hash-and-sign framework, the signing procedure of Falcon is in essence sampling a lattice point $\mathbf{v}$ from $D_{\mathcal{L}(\mathbf{B}),\sigma,\mathbf{c}}$ with a relatively small σ where $\mathbf{c}$ is the hashed message. The signature is $\mathbf{s} = \mathbf{v} - \mathbf{c}$ that is short: $\|\mathbf{s}\| \approx \sigma\sqrt{2n}$. To verify the signature, one just needs to compute the hashed message $\mathbf{c}$ and then check if $\mathbf{s} + \mathbf{c} \in \mathcal{L}$ and if $\|\mathbf{s}\|$ is less than the acceptance bound B. A simplified description of the signing and verification algorithms is given as follows:

Sign$(m, \mathsf{sk} = \mathbf{B})$
Compute $c = \mathsf{hash}(m) \in \mathcal{R}$;
Using sk, sample a short (s_1, s_2)
such that $s_1 + s_2 h = c \bmod q$;
If $\|(s_1, s_2)\| > B$, restart
Return $s = s_2$.

Verify$(m, s, \mathsf{pk} = h)$
Compute $c = \mathsf{hash}(m) \in \mathcal{R}$;
Compute $s_1 = c - sh \bmod q$;
If $\|(s_1, s)\| > B$, reject.
Accept.

Falcon sets $\sigma = 1.17\sqrt{q} \cdot \eta_\epsilon(\mathcal{R}^2)$ where $\eta_\epsilon(\mathcal{R}^2)$ is the smoothing parameter with respect to $\mathcal{R}^2$ and a small $\varepsilon > 0$. The acceptance bound is $B \approx 1.1\sqrt{2n}\sigma$.

3 Gaussian Samplers of Falcon

This section is dedicated to the presentation of the Gaussian samplers used in the signing procedure of Falcon. Indeed these samplers are the target of our side-channel attacks.

The signing procedure of Falcon relies on three layers of Gaussian sampling. At the top layer, the used sampler is FFOSampler and the output distribution is a lattice Gaussian $D_{\mathcal{L}(\mathbf{B}),\sigma,\mathbf{c}}$. At the intermediate layer, the sampler SamplerZ samples from some integer Gaussian $D_{\mathbb{Z},\sigma',c}$ where σ' and c are variable. At the bottom layer, the sampler BaseSampler samples from a fixed half integer Gaussian $D^+_{\mathbb{Z},\sigma_{\max},0}$.

3.1 FFOSampler

The FFOSampler algorithm is a ring variant of the KGPV sampler [14,22] based on fast Fourier nearest plane algorithm [8]. In FFOSampler, lattice Gaussian sampling is reduced to a series of integer Gaussian samplings, which is the same as

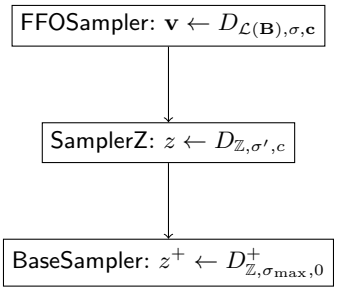

Fig. 1. Three layers of Gaussian samplers in Falcon signing algorithm.

Algorithm 1: The KGPV sampler

Input: a basis $\mathbf{B} = (\mathbf{b}_0, \cdots, \mathbf{b}_{n-1})$, a center $\mathbf{c}$ and $\sigma \geq \|\mathbf{B}\|_{GS} \cdot \eta_\epsilon(\mathbb{Z})$
Output: a lattice point $\mathbf{v}$ following a distribution close to $D_{\mathcal{L}(\mathbf{B}),\sigma,\mathbf{c}}$.

$\mathbf{v} \leftarrow \mathbf{0}, \mathbf{c}' \leftarrow \mathbf{c}$
for $i = n-1, \cdots, 0$ **do**
  $c_i'' = \langle \mathbf{c}', \widetilde{\mathbf{b}}_i \rangle / \|\widetilde{\mathbf{b}}_i\|^2, \sigma_i = \sigma / \|\widetilde{\mathbf{b}}_i\|$
  $z_i \leftarrow \mathsf{SamplerZ}(\sigma_i, c_i'' - \lfloor c_i'' \rceil) + \lfloor c_i'' \rceil$
  $\mathbf{c}' \leftarrow \mathbf{c}' - z_i \mathbf{b}_i, \mathbf{v} \leftarrow \mathbf{v} + z_i \mathbf{b}_i$
end for
return $\mathbf{v}$

the KGPV algorithm. Therefore we just describe the KGPV sampler in Algorithm 1.

3.2 SamplerZ

The integer Gaussian samplings in FFOSampler have variable standard deviations and centers, which complicates the implementation. To this end, SamplerZ uses rejection sampling to obtain target samples from a fixed half integer Gaussian. It first generates $z^+ \sim D^+_{\mathbb{Z},\sigma_{\max},0}$ by calling BaseSampler, then computes $z \leftarrow b + (2b-1)z^+$ with a random bit b, and finally outputs z with certain probability. A detailed algorithmic description is given in Algorithm 2 where $\sigma_{min} = 1.2778$ and $\sigma_{max} = 1.8205$ for Falcon-512. Particularly, SamplerZ is provably resistant against timing attacks [18].

3.3 BaseSampler

The BaseSampler algorithm for $D^+_{\mathbb{Z},\sigma_{\max},0}$ is implemented by table-based approach as described in Algorithm 3. Specifically, BaseSampler uses the (scaled) reverse cumulative distribution table (RCDT) of 18 items, which ensures the distribution sufficiently close to $D^+_{\mathbb{Z},\sigma_{\max},0}$. Also, the implementation of BaseSampler is constant time.

Algorithm 2: SamplerZ

Input: a center $c \in [0, 1)$ and standard deviation $\sigma' \in [\sigma_{min}, \sigma_{max}]$
Output: an integer $z \sim D_{\mathbb{Z},\sigma',c}$

1 $z^+ \leftarrow \mathsf{BaseSampler}()$
2 $b \xleftarrow{\$} \{0, 1\}$
3 $z \leftarrow b + (2b - 1)z^+$
4 $x \leftarrow -\frac{(z-c)^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma^2_{max}}$
5 **return** z with probability $\frac{\sigma_{min}}{\sigma'} \cdot \exp(x)$, otherwise restart;

Algorithm 3: BaseSampler

Output: an integer $z^+ \sim D^+_{\mathbb{Z},\sigma_{\max},0}$

1 $u \xleftarrow{\$} \{0, 1\}^{72}$
2 $z^+ \leftarrow 0$
3 **for** $i = 0 \cdots 17$ **do**
4 $\quad z^+ \leftarrow z^+ + [\![u < RCDT[i]]\!]$
5 **end for**
6 **return** z^+

4 Improved Key Recovery from Half Gaussian Leakage

While the distribution of Falcon signatures is statistically independent of the secret key, the intermediate variables during Falcon's Gaussian sampling are sensitive, which poses a threat to the side-channel security. Recently, Guerreau et al. proposed a side-channel attack on Falcon exploiting power leakage within BaseSampler [17]. This attack is quite demanding in terms of computation resources and measurements: a direct key recovery for Falcon-512 needs $\approx$ 10 million of signature measurements, and with 1 million traces, the key recovery has to resort to lattice reduction requiring around 1000 hours CPU time.

In this section, we propose an improved key recovery exploiting the same side-channel leakage exposed in [17]. With around 220 000 traces, our attack suffices to recover the key within half an hour with a probability of $\approx 25\%$. If lattice reduction is allowed, the number of required traces can be further reduced.

4.1 The Attack of [17]

Let us first recall the attack of [17] for better completeness and comparisons.

Half Gaussian leakage. Falcon's BaseSampler uses a table-based approach that was shown to be vulnerable to simple power analysis in [21]. More precisely, through the power consumption of the comparison $[\![u < RCDT[i]]\!]$ (line 4, Algorithm 3), one can effectively determine the value of z^+. The attack of [17]

exploits this leakage to classify if $z^+ = 0$ or not. When $z^+ = 0$, the corresponding output of SamplerZ belongs to $\{0, 1\}$. This allows to filter the signatures $\mathbf{s} = \sum_{i=0}^{2n-1} y_i \cdot \widetilde{\mathbf{b}}_i$ with $y_0 \in (-1, 1]$ where $\widetilde{\mathbf{b}}_0 = \mathbf{b}_0 = (g, -f)$ is the secret key. The region of filtered signatures is a slice in the direction of $\mathbf{b}_0$ (see Figure 2). In this paper, the leakage used in [17] is called *half Gaussian leakage*.

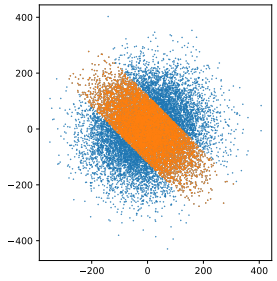

Fig. 2. Simplified 2-dimensional representation of Falcon signatures. Signatures with $y_0 \in (-1, 1]$ are in orange.

The key recovery. The slice of filtered signatures can be seen as a deformed parallelepiped of $\mathbf{B}$. The authors of [17] thus propose to recover the secret key using a variant of the parallelepiped-learning attack, developed in [28,7]. Since only one direction of the parallelepiped of $\mathbf{B}$ is preserved in the slice, the key recovery of [17] needs much more signatures to reconstruct $\mathbf{B}$ compared with the previous attacks [28,7].

4.2 Our Key Recovery

Let us first formally define the *Learning Slice Problem*.

Definition 1 ($\mathsf{LSP}_{b,\sigma,N}$). *Given* $\mathbf{b} \in \mathbb{R}^n$, *let* $\mathcal{S}_{\mathbf{b}}(b) = \{\mathbf{v} : |\langle \mathbf{v}, \mathbf{b} \rangle| \leq b\}$. *Let* D_s *be the conditional distribution of* $\mathbf{z} \sim (\mathcal{N}(0, \sigma^2))^n$ *given* $\mathbf{z} \in \mathcal{S}_{\mathbf{b}}(b)$. *Given* N *independent samples drawn from* D_s, *find an approximation of* $\pm\mathbf{b}$.

With half Gaussian leakage, we are able to identify signatures in $\mathcal{S}_{\mathbf{b}_0}(\|\mathbf{b}_0\|^2)$. Hence the key recovery now becomes to solve $\mathsf{LSP}_{b,\sigma,N}$. Our idea stems from the geometric intuition that the projection of signatures in the slice on $\mathbf{b}_0$ tends to be unusually short. Instead of resorting to the fourth moment (known as kurtosis) as in parallelepiped-learning attacks, we discover that the covariance of the samples in the slice, i.e. filtered signatures, suffices to reveal the secret $\mathbf{b}_0$. Our LSP algorithm consists of two steps:

1. we learn the direction of $\mathbf{b}_0$;
2. we estimate $\|\mathbf{b}_0\|$;

Learning the slice direction. Let $\mathbf{B} = (\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{n-1})$ of full-rank where $\mathbf{b}_0$ is the solution to the LSP instance. Let $\mathbf{d}_i = \widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|$, then $\mathbf{D} = (\mathbf{d}_0, \cdots, \mathbf{d}_{n-1})$ is orthogonal. For $\mathbf{s} \sim (\mathcal{N}(0, \sigma^2))^n$, let $\mathbf{s} = \sum_i y_i \mathbf{d}_i$, then the coefficients y_i independently follow $\mathcal{N}(0, \sigma^2)$ and $\mathbf{Cov}[\mathbf{s}] = \sigma^2 \mathbf{I}$. When $\mathbf{s} \in \mathcal{S}_{\mathbf{b}_0}(b)$, we have $|y_0| \leq \frac{b}{\|\mathbf{b}_0\|}$ and thus the variance of y_0 is $\sigma'^2 < \sigma^2$. Then the covariance of $\mathbf{s}$ given $\mathbf{s} \in \mathcal{S}_{\mathbf{b}_0}(b)$ becomes

$$\mathbf{Cov}[\mathbf{s} | \mathbf{s} \in \mathcal{S}_{\mathbf{b}_0}(b)] = \mathbf{D} \cdot \begin{pmatrix} \sigma'^2 & \\ & \sigma^2 \mathbf{I} \end{pmatrix} \cdot \mathbf{D}^t.$$

In the above covariance matrix, the smallest eigenvalue σ' is unique and clearly less than others. In addition, the eigenvector corresponding to the smallest eigenvalue is in the same direction as $\mathbf{b}_0$. Therefore, we can learn the direction of $\mathbf{b}_0$ through spectral decomposition.

Learning the norm of the secret. The covariance $\mathbf{Cov}[\mathbf{s} | \mathbf{s} \in \mathcal{S}_{\mathbf{b}_0}(b)]$ also leaks the information of $\|\mathbf{b}_0\|$. Specifically, the coefficient y_0 of samples in the slice follows the truncated normal distribution $\mathcal{N}(0, \sigma^2)$ over $\left[-\frac{b}{\|\mathbf{b}_0\|}, \frac{b}{\|\mathbf{b}_0\|}\right]$. Its variance is

$$\sigma'^2 = \frac{\int_{-b'}^{b'} x^2 \exp(-\frac{x^2}{2\sigma^2}) dx}{\int_{-b'}^{b'} \exp(-\frac{x^2}{2\sigma^2}) dx} \quad \text{where} \quad b' = \frac{b}{\|\mathbf{b}_0\|}.$$

that can be also computed through spectral decomposition. Then $\|\mathbf{b}_0\|$ can be numerically estimated given σ'.

With the approximate direction and the norm of $\mathbf{b}_0$, we can immediately construct a solution to the $\mathsf{LSP}_{b,\sigma,N}$ instance. A theoretical justification for the effectiveness of our LSP algorithm is provided in Appendix A.

Key recovery from approximate vectors. Up to now, we have shown that one is able to get an approximate secret key $\mathbf{b}'_0$ by solving the underlying LSP instance given by half Gaussian leakage. By rounding the coefficients of $\mathbf{b}'_0$, an integer vector $(g', -f') \in \mathcal{R}^2$ is recovered. As a certain number, denoted N_0, of signature measurements are performed, $(g', -f')$ is exactly the key with good probability, that is set around 25% throughout the paper, in practice. Even with fewer traces, the key can be fully recovered by combining exhaustive search or lattice reduction and the cost depends on the size of $\mathbf{e} = (g - g', f' - f) \in \mathcal{R}^2$. We further introduce N_1 and $N_1(x)$ as follows:

- N_1 : when the number of traces $\geq N_1$, $\|\mathbf{e}\|_\infty \leq 1$ with good probability;
- $N_1(x)$: when the number of traces $\geq N_1(x)$, $\|\mathbf{e}\|_\infty \leq 1$ and $\|\mathbf{e}\|_1 \leq x$ with good probability.

It is worth noting that when $\|\mathbf{e}\|_\infty \leq 1$ and $\|\mathbf{e}\|_1 \leq x$, either $g - g'$ or $f - f'$ has hamming weight $\leq \lfloor x/2 \rfloor$. In practice, it suffices to correct either g' or f': exploiting the NTRU public key h, it is easy to derive the other half and to check if the guess is correct or not.

Remark 1. We particularly treat the case where $\mathbf{e}$ is ternary, as this allows a practical key recovery by a simple exhaustive search. However, larger errors can also be corrected by expensive lattice reduction. (see Section 4.3 for details).

4.3 Experimental Results of Key Recovery

The experiments focus on the key recovery, since our attack uses the same side-channel leakage presented in [17]. In fact, [17] has shown that the leakage can be correctly identified in practice with a fairly high probability: 94% for ChipWhisperer and 100% for ELMO. We did not repeat the measurements and just assumed a 100% accurate classification as done in [17].

We tested our key recovery attack over 40 Falcon-512 instances and 400 000 traces per instance. The practicality of our new key recovery is well supported by experimental results. More precisely, 360 000 traces suffices for our attack to directly recover the key. As a comparison, the attack in [17] requires about 10 000 000 traces. The value of $N_1(7)$ is around 220 000, and in this region, a certain proportion of keys can be recovered by combining a simple exhaustive search within half an hour. For clarity, we highlight that the trace number counts all signature measurements which is about twice the number of filtered signatures in the slice. Detailed experimental results are shown in Figure 3. We also tested our attack on Falcon-1024 and Falcon-256, and experimental results are given in Appendix B.

Furthermore, there is a tradeoff between measurement and computation. The approximation obtained from fewer traces can be used by lattice reduction to effectively reduce the cost of key recovery. Figure 4 shows the bit security estimated by leaky LWE estimator [4] given a certain number of signature measurements. Given 20 000 traces, the security of Falcon-512 would decrease from 133 bits to 85 bits.

In practice, the half Gaussian leakage is noisy, inducing errors in the classification. The error can be further amplified in presence of side-channel protections. In this respect, we conduct the attack by emulating the case that the classification of $z^+ = 0$ or not only has imperfect accuracy. The result is shown in Figure 5, where the required trace number increases with the classification accuracy. Notably, when the accuracy is 65%, an adversary is still able to practically recover the key using our attack with 10 million traces. In comparison, the attack in [17] cannot apply to inaccurate leakage classifications, because it requires that the domain of filtered signatures has a clear boundary.

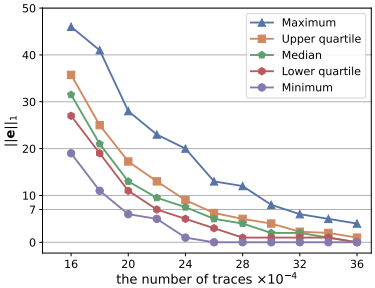

Fig. 3. The approximate error size $\|\mathbf{e}\|_1$ measured over 40 Falcon-512 instances. The vector $\mathbf{e}$ is ternary for all 40 tested instances with 10^5 traces.

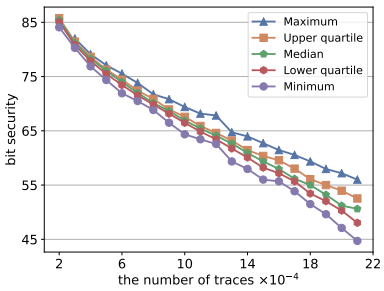

Fig. 4. The bit security estimated as per the approximate error. We use the Core-SVP model in classical setting, i.e. $2^{0.292\beta}$ where β is the required BKZ blocksize

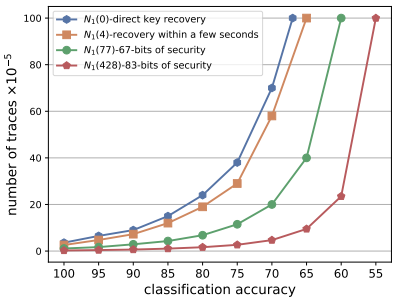

Fig. 5. The required trace numbers for different classification accuracies using half Gaussian leakage.

5 Power Analysis Using Sign Leakage

As outlined in Section 3.2, the integer Gaussian sampler SamplerZ requires transforming a half Gaussian sample into a bimodal one via a uniformly random sign flip and then accepting it with proper probability. While both the half Gaussian sample and the random sign can be revealed through single trace analysis as shown in [21], the sign leakage remains unexploited until now. Compared to the half Gaussian leakage, the sign flip seems to offer less information, as it can only help to filter signatures in a half-space instead of a slice.

In this section, we first identify the sign leakage in the reference implementation of Falcon. Then we show that sign leakage can indeed be used to mount effective key recovery attacks: about 170 000 traces is enough to fully recover the key. Perhaps counter-intuitively, the key recovery solely using sign leakage needs even fewer signature measurements than the one solely using half Gaussian leak-

age. Moreover, combining sign leakage with half Gaussian leakage, we can further reduce the requirements of measurements and computations for key recovery: a full key recovery needs only 45 000 signatures given two sources of leakage. At last, we propose a practical countermeasure to mitigate the sign leakage.

5.1 Side-Channel Analysis

As the goal of the side-channel analysis is to classify the sign of z (which is indicated by b), it is necessary to analyze its leakages. The most straightforward leakage of the sign should be directly from the generation of variable b, including the loading and storing process. We term this leakage type-1. Besides, the value of b also affects the intermediate variables in the Gaussian sampling. By its instruction, Falcon first performs half Gaussian sample to obtain the value z^+ and maps it to z using the sign-flip function based on a bit b, i.e., $[\![z \leftarrow b + (2b-1)z^+]\!]$ (line 3, Algorithm 2). Then, z is involved in the computation of x: $[\![x \leftarrow -\frac{(z-c)^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma_{max}^2}]\!]$ (line 4, Algorithm 2). We term the sign leakage from the calculation of z and x type-2.

To better analyze the leakages of the above two types, we insert delay macros (by using an empty loop) between the generation of b, the calculation of $[\![z \leftarrow b+(2b-1)z^+]\!]$ and $[\![x \leftarrow -\frac{(z-c)^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma_{max}^2}]\!]$. Thus, the power consumption before the first delay only contains the type-1 leakage. Meanwhile, the algorithm after the first delay comprises the loading of b and computation of z and x, thus containing both type-1 and type-2 leakages.

We run the reference implementation of Falcon (with the delay macro inserted) on an ARM Cortex-M4 STM32F407IGT6 microprocessor. The power traces are collected by using a PicoScope 3206D oscilloscope at a sampling rate of 1 GSa/s, equipped with a Mini-Circuits 1.9 MHz low pass filter. We collect 50 000 traces with different random seeds, and compute the Signal-to-Noise Ratio (SNR) with respect to the sign of z.

As shown in Figure 6, we can identify the three regions, as well as the corresponding leakages by peak clusters. Moreover, the SNRs of regions B and C (containing type-1 and type-2 leakage) are much larger than those of region A (only containing type-1 leakage), showing that the type-2 leakage is much more significant than the type-1. It conveys that the calculations of z and x can amplify the leakage of the sign (i.e., the value of b). We attribute the leakage amplification to the following reasons.

- The first reason should be the power consumption of $((\mathsf{b} \ll 1) - 1)$. Concretely, the corresponding register is assigned to the value of b, then adds itself and minus 1. The value in register turns into -1 (0xFFFFFFFF for complement) when $b = 0$ and 1 (0x00000001 for complement) when $b = 1$. The Hamming distance of two results is 31, which is sufficiently large to distinguish the sign of the output z. It should be noted that this type of leakage was detected in [21] and comprehensively analyzed in the very recent work by Wisiol et al. [36].

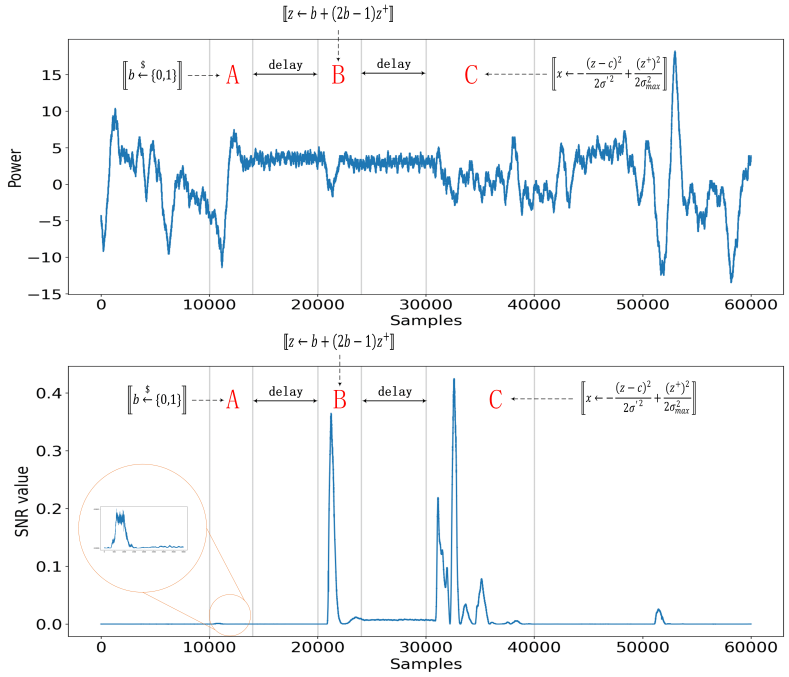

Fig. 6. Power traces and the corresponding SNR value.

– Another reason should be in $[\![x \leftarrow -\frac{(z-c)^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma^2_{max}}]\!]$ (line 4, Algorithm 2). In the calculation of $z - c$, z is an integer while c is a floating point number. In most cases, z is first converted to the floating point number, and then the subtraction is performed. The former is essentially a conversion between complement and floating representations of z. If z is negative, the Hamming distance of complement and floating representation is relatively large. This eventually brings the sign leakage in the step of $[\![x \leftarrow -\frac{(z-c)^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma^2_{max}}]\!]$.

To verify the vulnerabilities in practice, we conduct the Gaussian template attack [3], where the number of profiling traces varies from 70 to 100, 000. After the profiling, we repeat the single-trace attack 5, 000 times (with different attacking traces) to calculate the success rate. We perform the evaluation with four different configurations: 3 attacks targeting regions A, B, and C separately, and 1 attack targeting their combination. For each configuration, we apply the principal component analysis (PCA) to the samples before profiling and attacking, and then only target the points of the first 65 principal components.

Figure 7 presents the classification accuracy (as functions of the number of profiling traces). The results show that the attacks using samples in regions B and C (involving type-1 and type-2 leakages) are significantly better than those using region A (only involving type-1 leakages). The leakages in region C have

led to an attack with an almost 1 classification accuracy, and using the leakages in region B can also achieve an accuracy ≈ 0.9. On the contrary, the classification accuracy corresponding to region A is up to 0.52. At last, using the combination of three regions leads to the best attack, which is slightly better than the attack using region C.

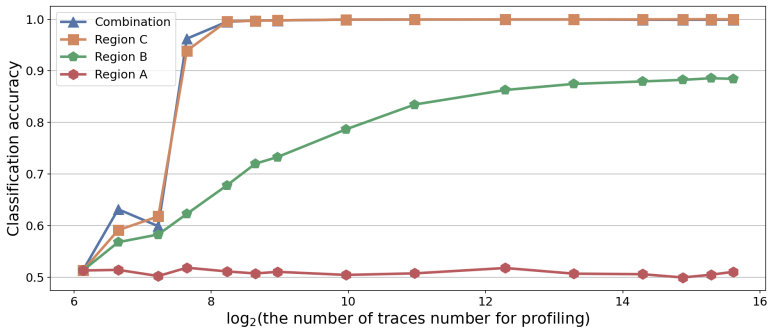

Fig. 7. The classification accuracy targeting the three regions and combination thereof.

5.2 Key Recovery Using Sign Leakage

Using the sign leakage, one can determine whether a signature $\mathbf{s}$ is in the halfspace $\mathcal{H}^+ = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{b}_0 \rangle \geq 0\}$ or $\mathcal{H}^- = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{b}_0 \rangle < 0\}$ (see Figure 8). It is worth noting that one can transform a signature in $\mathcal{H}^+$ into one in $\mathcal{H}^-$ by multiplying -1. Therefore no waste of signature measurements occurs in this classification, which is different from the case presented in Section 4.

To study the key recovery, we define the *Learning Halfspace Problem*.

Definition 2 ($\mathsf{LHP}_{\sigma,N}$). *Given* $\mathbf{b} \in \mathbb{R}^n$, *let* $\mathcal{H}^+_{\mathbf{b}} = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{b} \rangle \geq 0\}$. *Let* D_h *be the conditional distribution of* $\mathbf{z} \sim (\mathcal{N}(0, \sigma^2))^n$ *given* $\mathbf{z} \in \mathcal{H}^+_{\mathbf{b}}$. *Given* N *independent samples drawn from* D_h, *find an approximate direction of* $\pm\mathbf{b}$.

Exploiting the sign leakage, signatures can be transformed into Gaussian samples in $\mathcal{H}^+_{\mathbf{b}_0}$. By solving $\mathsf{LHP}_{\sigma,N}$, we can get an approximate direction of $\mathbf{b}_0$.

The distribution of the given samples in the $\mathsf{LHP}_{\sigma,N}$ instance is determined by the secret $\mathbf{b}_0$. It is feasible to get a solution to $\mathsf{LHP}_{\sigma,N}$ through the spectral decomposition of $\mathbf{Cov}[\mathbf{s}|\mathbf{s} \in \mathcal{H}^+_{\mathbf{b}_0}]$ as done in Section 4. However, the accuracy of the solution is poor because the gap between the smallest eigenvalue and others is reduced. To overcome this issue, we propose to use a rough LHP solution to filter a slice and then apply the previous LSP algorithm to get an accurate solution. This can be roughly viewed as the following reduction:

$$\mathsf{LHP}_{\sigma,N} \rightarrow \mathsf{LSP}_{b,\sigma,N'}.$$

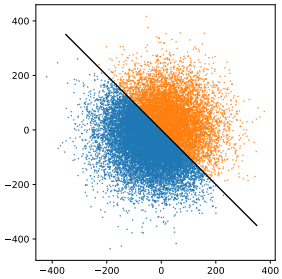

Fig. 8. Simplified 2-dimensional representation of Falcon signatures. Signatures in $\mathcal{H}^+$ (resp. $\mathcal{H}^-$) are in orange (resp. blue).

Specifically, our LHP algorithm proceeds as follows:

1. we learn a relatively rough direction, denoted $\mathbf{v}$, of $\mathbf{b}_0$ from samples in $\mathcal{H}^+_{\mathbf{b}_0}$;
2. we filter out those samples in $\mathcal{S}_{\mathbf{v}}(b)$ using $\mathbf{v}$;
3. we learn the direction of $\mathbf{b}_0$ from the filtered samples in $\mathcal{S}_{\mathbf{v}}(b)$;

Learning a rough direction. By the same argument with Section 4, we have

$$\mathbf{Cov}[\mathbf{s}|\mathbf{s} \in \mathcal{H}^+_{\mathbf{b}_0}] = \mathbf{D} \cdot \begin{pmatrix} \sigma'^2 & \\ & \sigma^2\mathbf{I} \end{pmatrix} \cdot \mathbf{D}^t$$

where $\mathbf{D} = (\mathbf{d}_0, \cdots, \mathbf{d}_{n-1})$ with $\mathbf{d}_i = \widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|$. The term σ'^2 equals the variance of half Gaussian, and a routine computation yields $\sigma'^2 = \sigma^2(1 - \frac{2}{\pi}) < \sigma^2$. Therefore the direction of $\mathbf{b}_0$ still corresponds to the eigenvector with respect to eigenvalue σ'^2.

Remark 2. The expectation of samples in $\mathcal{H}^+_{\mathbf{b}_0}$ is also the direction of $\mathbf{b}_0$. Nevertheless, the use of the expectation does not improve the learning accuracy as per our experimental results.

Filtering out a slice. To refine the learning accuracy, we attempt to amplify the distinction between σ' and σ. To do so, we use the above rough direction, denoted $\mathbf{v}$, to classify all samples into two sets $\mathcal{S} = \{\mathbf{s} \mid |\langle \mathbf{s}, \mathbf{v} \rangle| \leq b\}$ and $\mathcal{C} = \{\mathbf{s} \mid |\langle \mathbf{s}, \mathbf{v} \rangle| > b\}$ (see Figure 9). The parameter b decides both the width of the approximate slice and the proportion of filtered samples. For a tradeoff, our key recovery sets $b = 1.17\sqrt{q}$ that is around the expectation of the secret key norm as per Falcon parameters. This actually corresponds to the case in Section 4. It should be noted here that the covariance of filtered samples is

$$\mathbf{Cov}[\mathbf{s}|\mathbf{s} \in \mathcal{H}^+_{\mathbf{b}_0} \cap \mathcal{S}] = \mathbf{D}' \cdot \begin{pmatrix} \sigma'^2 & & & \\ & \sigma_1^2 & & \\ & & \ddots & \\ & & & \sigma_{n-1}^2 \end{pmatrix} \cdot \mathbf{D}'^t$$

where $\mathbf{D}'$ is slightly different from $\mathbf{D}$ due to the inaccuracy of $\mathbf{v}$ and $\sigma_1, \cdots, \sigma_{n-1}$ are no longer equal. Still, σ' is clearly less than others. As a consequence, its corresponding eigenvector is supposed to be in a very close direction of $\mathbf{b}_0$. Applying the LSP algorithm in Section 4 on filtered samples, one can obtain an approximate direction of $\mathbf{b}_0$ that is more accurate than $\mathbf{v}$.

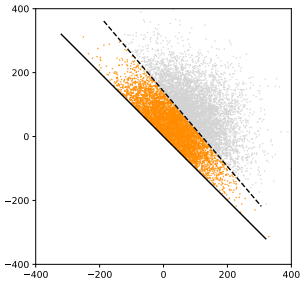

Fig. 9. Simplified 2-dimensional representation of Falcon signatures. Signatures in $\mathcal{H}^+_{\mathbf{b}_0} \cap \mathcal{S}_{\mathbf{v}}(b)$ are in orange.

Remark 3. Strictly speaking, the region $\mathcal{H}^+_{\mathbf{b}_0} \cap \mathcal{S}_{\mathbf{v}}(b)$ is not an exact slice as the directions of $\mathbf{b}_0$ and $\mathbf{v}$ differ. But this region still maintains some information of $\mathbf{b}_0$ from $\mathcal{H}^+_{\mathbf{b}_0}$ that is captured by the LSP algorithm to refine the direction.

Remark 4. We can also use this idea to reduce the width of the slice in Section 4, but the effect is not good for our key recovery. The reason is that signatures in $\mathcal{S}$ are distributed densely, the reduction of slice width would eliminate a big number of signatures. In this section, the signature density in $\mathcal{C}$ is lower than that in $\mathcal{S}$. We therefore can reduce the slice width at the cost of fewer signatures.

Key recovery from an approximate direction. While $\|\mathbf{b}_0\|$ cannot be learnt purely from the sign information, we can still approximate $\|\mathbf{b}_0\|$ with some alternatives in $\{1.17\sqrt{q}, \cdots, 1.17\sqrt{q} - 10\}$. This works well in practice: one can always get one approximation well close to $\mathbf{b}_0$ using some alternatives to $\|\mathbf{b}_0\|$. In later experimental results, we shall present the best approximation for each tested instance.

5.2.1 Experimental Results We use the same 40 Falcon-512 instances as in Section 4. Figure 10 shows the detailed experimental results. In the context of the key recovery in this subsection, $N_1(7)$ is around 170 000. This implies that one can recover the key from the approximation within half an hour with a probability of $\approx 25\%$ given a moderate number of traces. Compared with the key recovery presented in Section 4, the attack exploiting sign leakage seems more

powerful, as it requires fewer traces to achieve the same size of the approximate error. A crucial reason for this is that the sign information of each signature contributes to the key recovery, at least to recover a rough direction, but in the attack in Section 4, about one half measured signatures are directly discarded in the first place. We also exhibit the tradeoff between measurements and the cost of key recovery combining lattice reduction in Figure 11: 20 000 signature measurements would reduce the security of Falcon-512 by 50 bits.

Furthermore, we test our attack in the case of inaccurate sign classification. As shown in Figure 12, our attack is robust to inaccurate classification: for 65% classification accuracy, 10 million traces are sufficient for key recovery in a short time. One can also observe that the number of required traces grows sharply as the accuracy gets below 65%.

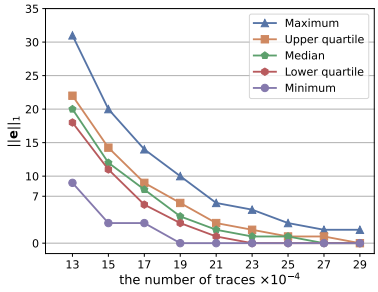

Fig. 10. The approximate error size $\|\mathbf{e}\|_1$ measured over 40 Falcon-512 instances solely using the sign leakage.

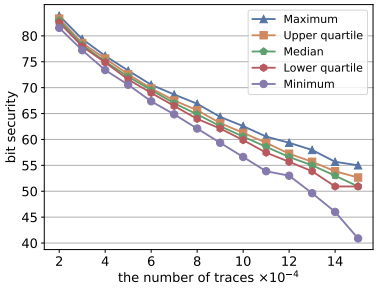

Fig. 11. The bit security estimated as per the approximate error. We use the Core-SVP model in classical setting, i.e. $2^{0.292\beta}$ where β is the required BKZ blocksize

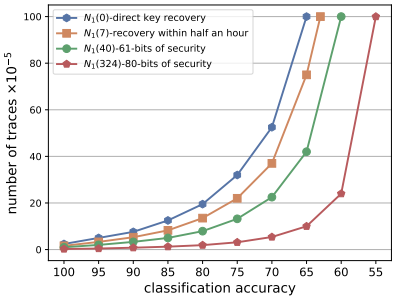

Fig. 12. The required trace numbers for different classification accuracies using sign leakage.

5.3 Key Recovery Using Both Sign and Half Gaussian Leakages

It is natural to work with both sign leakage and half Gaussian leakage (presented in Section 4). Specifically, this allows filtering a slice that is only one half wide as the one filtered solely by half Gaussian leakage. Through spectral decomposition of the covariance, one can learn the secret key.

The combination of two leakages significantly improves the key recovery. When 20 000 signature measurements are available, the approximate error becomes ternary for all 40 tested instances. Detailed experimental results are presented in Figures 13 and 14. In particular, with only 45 000 traces, one can fully recover the key with a probability of $\approx 25\%$ within half an hour. With around 12 000 traces, the attacker may reduce the security of Falcon-512 by 60 bits. Nevertheless, further tradeoff seems infeasible. As the number of traces is insufficient, the approximate error can enlarge quickly due to the measurement error, which makes the approximation ineffective.

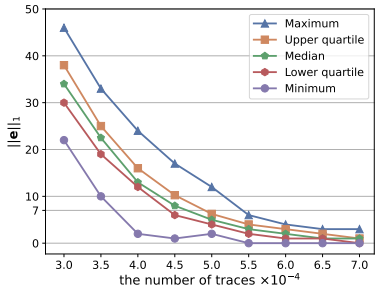

Fig. 13. The approximate error size $\|\mathbf{e}\|_1$ measured over 40 Falcon-512 instances using both sign and half Gaussian leakages.

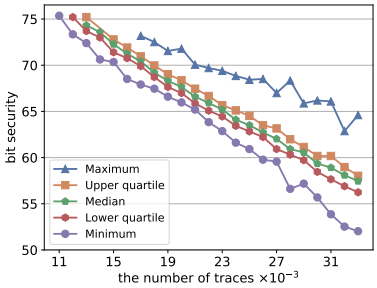

Fig. 14. The bit security estimated as per the approximate error. We use the Core-SVP model in classical setting, i.e. $2^{0.292\beta}$ where β is the required BKZ blocksize

5.4 A Countermeasure Against the Sign Leakage

We present a countermeasure to mitigate the leakage of the sign in the Gaussian sampling, which is made up of two components as follows.

The first component is for the direct leakage of the sign b. The leakage (e.g., power consumption) of a variable in software is largely related to its Hamming weight [26]. Thus, to eliminate the difference in Hamming weight between different values of b, the countermeasure encodes the sign by $\{1, 2\}$ instead of $\{0, 1\}$. Concretely, we first generate a 4-bit variable t by uniformly sampling a value in

Algorithm 4: Protected SamplerZ

Input: a center c'' and standard deviation $\sigma' \in [\sigma_{min}, \sigma_{max}]$
Output: an integer $z \sim D_{\mathbb{Z},\sigma',c''}$

1 $c \leftarrow c'' - \lfloor c'' \rfloor$
2 $z^+ \leftarrow$ BaseSampler()
3 $(\tilde{t}[0], \ldots, \tilde{t}[15]) \leftarrow (2,1,1,2,2,1,1,2,2,1,2,1,1,2,1,2)$
4 $t \xleftarrow{\$} \{0, \ldots, 15\}$
5 $b \leftarrow \tilde{t}[t]$
6 $(\tilde{c}[0], \tilde{c}[1], \tilde{c}[2]) \leftarrow (0, c, 1-c)$
7 $(\tilde{z}[0], \tilde{z}[1], \tilde{z}[2]) \leftarrow (0, \lfloor c'' \rfloor - z^+, \lfloor c'' \rfloor + 1 + z^+)$
8 $x \leftarrow -\frac{(z^+ + \tilde{c}[b])^2}{2\sigma'^2} + \frac{(z^+)^2}{2\sigma_{max}^2}$
9 **return** $\tilde{z}[b]$ with probability $\frac{\sigma_{min}}{\sigma'} \cdot \exp(x)$, otherwise restart;

$\{0, \ldots, 15\}$, and map it to the variable b in $\{1, 2\}$ by using a look-up table with 16 entries in $\{1, 2\}$.

The second component is for the leakage amplified from the computation of z and x. By its instruction in Algorithms 1 and 2, the output of SamplerZ will be added by $\lfloor c'' \rfloor$. Thus, we can consider the output to be $z + \lfloor c'' \rfloor$ instead of z. We observe that, unlike the computation, the leakages of variables $z + \lfloor c'' \rfloor$ and x are not quite related to the sign. It conveys that our goal should be to mitigate the leakage during the computation. The main idea is that, for each sign value (positive or negative), we directly compute the values of $z + \lfloor c'' \rfloor$ and x, and then choose the correct ones by using $b \in \{1, 2\}$. The sign leakage within the calculation of x is from the calculation of $(z - c)^2$, more precisely, $z - c$. We note that $(z - c)^2 = (z^+ + \tilde{c}[b])^2$ where $\tilde{c}[b] = c$ for $b = 1$ and $\tilde{c}[b] = 1 - c$ for $b = 2$. Instead of computing two x's for $b \in \{1, 2\}$, it suffices to compute two $\tilde{c}[b]$'s and to perform the calculation of x using $(z^+, \tilde{c}[b])$ only once.

The new SamplerZ equipped with the above components is provided in Algorithm 4. Lines 3-5 present the generation of $b \in \{1, 2\}$, and Lines 6-9 present the calculation of x and $z + \lfloor c'' \rfloor$.

To verify the effectiveness of the countermeasure, we implement the protected Gaussian sampling in C and collect the power traces using the same setup as in Section 5.1. The SNR for the sign value is depicted in Figure 15, which is much lower than that of the unprotected algorithm (see Figure 6). We conduct the template attack with 5 000 traces to calculate the classification accuracy. As shown in Figure 16, with the increase of the number of profiling traces, the classification accuracy is growing up to ≈ 0.52.

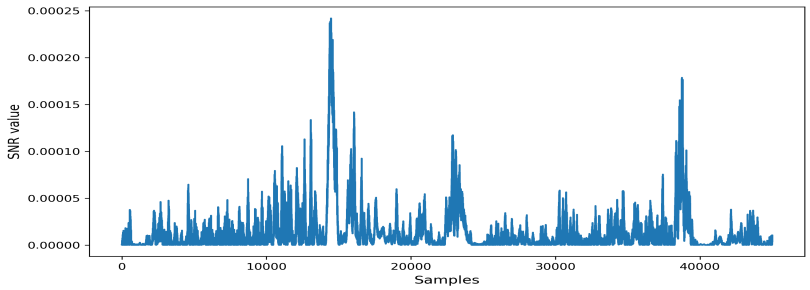

Fig. 15. SNR for the sign value of the Algorithm 4.

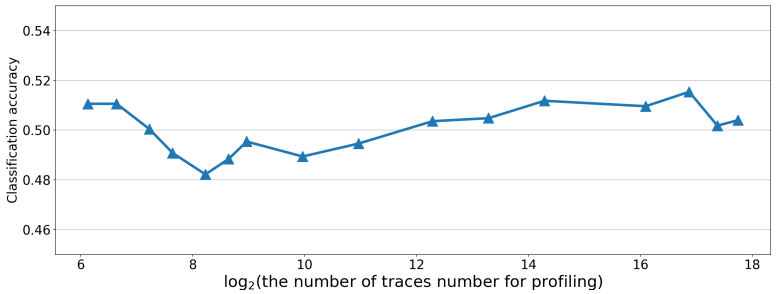

Fig. 16. Classification accuracy of the template attack on Algorithm 4.

Remarks. Our countermeasure can mitigate, but cannot prevent the leakage of the sign. As we can see from the experimental results, the accuracy of the template attack for sign classification is still significant (up to ≈ 0.52), and it is even possible to be higher if an adversary

1. adopts a more sophisticated setup for acquisition, or
2. exploits other targets than the direct leakage of b and the computation of x and z.

In this respect, we position our countermeasure as a (quite) efficient method to make the offline key recovery attack more difficult. As shown in Figure 12, the number of required traces grows dramatically when the accuracy of the sign classification decreases. One can still conduct a successful attack if she can sign a lot of times (it usually can be avoided in practice by setting a counter for the maximum time of calls). A candidate of sufficiently secure countermeasures

might be masking, with inevitably high overhead. Thus, we deem an efficient and provably secure countermeasure for Gaussian sampling as challenging and promising further work.

6 Attacks on Mitaka

Mitaka [11] is a recent variant of Falcon. Its base sampler and integer sampler are almost the same as those in Falcon, hence both half Gaussian leakage and sign leakage can be identified within Mitaka. Nevertheless, Mitaka performs lattice Gaussian sampling in a different way from Falcon, which significantly changes the distributions of the signatures filtered as per the leakages. It is therefore unclear if and how previous attacks apply to Mitaka.

To this end, we test previous attacks on Mitaka. Experiments verify that half Gaussian leakage and sign leakage can indeed lead to a key recovery in Mitaka, but the key recovery requires much more traces compared to the case of Falcon.

6.1 Mitaka Signatures Filtered by Leakages

Mitaka uses the hybrid sampler [32] (Algorithm 5) as its lattice Gaussian sampler.[7] The hybrid sampler follows the framework of the KGPV sampler (Algorithm 1) that is a randomized version of Babai's nearest plane algorithm, but the randomization is done at the ring level instead of the integer level. The ring-level randomization (Algorithm 6) is accomplished by Peikert's sampler [29] which is a randomized version of Babai's rounding-off algorithm. Specifically, to sample from $D_{\mathcal{R},\sigma,D}$, the randomization subroutine proceeds in two steps:

1. (perturbation sampling): it samples a perturbation $U \leftarrow \sigma_p \cdot \mathcal{N}_{\mathcal{K}_\mathbb{R},1}$ where $\sigma_p\overline{\sigma_p} = \sigma\overline{\sigma} - r^2$ and $\mathcal{N}_{\mathcal{K}_\mathbb{R},1}$ denotes the normal distribution over $\mathcal{K}_\mathbb{R}$.
2. (rounding-off): it samples the output $Z \leftarrow D_{\mathcal{R},r,D-U}$.

From Algorithms 5 and 6, it follows that the signature $\mathbf{s}$ can be written as $\mathbf{s} = \mathbf{v} - \mathbf{c} = \sum_{i=0}^{1} \widetilde{\mathbf{b}}_i(Z_i - D_i) = \sum_{i=0}^{1} \widetilde{\mathbf{b}}_i(Y_i + U_i)$ where $Y_i = Z_i - D'_i \in \mathcal{K}_\mathbb{R}$ and $U_i \in \mathcal{K}_\mathbb{R}$ is the perturbation. Then the signature is identified with $\mathbf{s} = \sum_{i=0}^{2n-1}(y_i + u_i)\mathbf{h}_i$ where $(y_0, \cdots, y_{2n-1})$ (resp. $(u_0, \cdots, u_{2n-1})$) is the coefficient vectors of (Y_0, Y_1) (resp. (U_0, U_1)) and $\mathbf{h}_i$'s correspond to $\widetilde{\mathbf{B}}$.

We target the integer Gaussian sampler called in the rounding-off step. Similar to the case of Falcon, half Gaussian leakage allows filtering signatures with $y_0 \in (-1, 1]$, while sign leakage allows distinguishing $y_0 > 0$ or not. Exploiting these leakages, one can actually distort the spherical Gaussian in the direction of $\mathbf{h}_0$ (See Figure 17). This makes our previous attacks feasible. Note that the domain of filtered signatures is now the ambient space due to the perturbation u_0, thus parallelepiped-learning attacks do not seem to work.

[7] We do not discuss the integer arithmetic friendly version of Mitaka that uses the integral perturbation sampler [5,12] proceeding differently.

Algorithm 5: Hybrid sampler

Input: a basis $\mathbf{B} = (\mathbf{b}_0, \mathbf{b}_1) \in \mathcal{R}^{2\times 2}$ and its GSO (over $\mathcal{K}$) $\widetilde{\mathbf{B}} = (\widetilde{\mathbf{b}_0}, \widetilde{\mathbf{b}_1})$, a center $\mathbf{c} \in \mathcal{K}^2$ and $\sigma > 0$

Output: a lattice point $\mathbf{v}$ following a distribution close to $D_{\mathcal{L}(\mathbf{B}),\sigma,\mathbf{c}}$.

$\mathbf{v}_1 \leftarrow \mathbf{0}, \mathbf{c}_1 \leftarrow \mathbf{c}$

$D_1 \leftarrow \frac{\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{c}_1} \rangle}{\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle}$, $\sigma_1 \leftarrow \sqrt{\frac{\sigma^2}{\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle}}$

$Z_1 \leftarrow \mathsf{RingPeikert}(D_1, \sigma_1)$

$\mathbf{v}_0 \leftarrow \mathbf{b}_1 Z_1, \mathbf{c}_0 \leftarrow \mathbf{c}_1 - \mathbf{b}_1 Z_1$

$D_0 \leftarrow \frac{\langle \widetilde{\mathbf{b}_0}, \widetilde{\mathbf{c}_0} \rangle}{\langle \widetilde{\mathbf{b}_0}, \widetilde{\mathbf{b}_0} \rangle}$, $\sigma_0 \leftarrow \sqrt{\frac{\sigma^2}{\langle \widetilde{\mathbf{b}_0}, \widetilde{\mathbf{b}_0} \rangle}}$

$Z_0 \leftarrow \mathsf{RingPeikert}(D_0, \sigma_0)$

$\mathbf{v} \leftarrow \mathbf{v}_0 + \mathbf{b}_0 Z_0$

return $\mathbf{v}$

Algorithm 6: RingPeikert

Input: a center $D \in \mathcal{K}$ and $\sigma \in \mathcal{K}_{\mathbb{R}}$.

Output: $z \in \mathcal{R}$ following a distribution close to $D_{\mathcal{R},\sigma,D}$

Compute $\sigma_p \in \mathcal{K}_{\mathbb{R}}$ such that $\sigma_p \overline{\sigma_p} = \sigma\overline{\sigma} - r^2$ for some $r > 0$

$U \leftarrow \sigma_p \cdot \mathcal{N}_{\mathcal{K}_{\mathbb{R}},1}$, $D' \leftarrow D + U$

for $i = 0 \cdots n-1$ **do**

  $z_i \leftarrow \mathsf{SamplerZ}(r, d'_i - \lfloor d'_i \rfloor) + \lfloor d'_i \rfloor$ /** $D' = \sum_i d'_i x^i \in \mathcal{K}_{\mathbb{R}}$ **/

end for

return $Z = \sum_i z_i x^i \in \mathcal{R}$

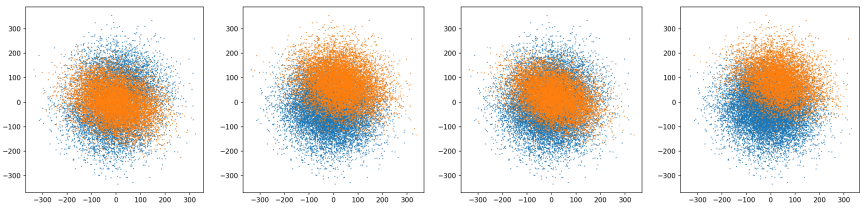

Fig. 17. Simplified 2-dimensional representation of Mitaka signatures. From left to right, the first graph is for half Gaussian leakage: signatures with $y_0 \in (-1, 1]$ in orange; the second graph is for sign leakage: signatures with $y_0 \in [0, +\infty)$ in orange; the third graph is for the combination of two leakages: signatures with $y_0 \in [0, 1]$ in orange; the fourth graph is for the combination of two leakages: signatures with $y_0 \in (1, +\infty)$ in orange.

6.2 Experimental Results

Both half Gaussian leakage and sign leakage can be well detected as shown in [17] and Section 5. We thus only present the experimental results for the key recovery procedure.

6.2.1 Key Recovery Using Half Gaussian Leakage. We tested the attack in Section 4 and experimental results are shown in Figure 18. Experiments validate the effectiveness of the attack: one can get a good approximation or a full recovery of the key with a certain number of Mitaka signatures. However, compared to the attack on Falcon, the key recovery on Mitaka requires much more signatures: the number of signatures required for a quick key recovery gets close to 9 million. This is because the condition number of the covariance of filtered signatures gets smaller due to the existence of the perturbation.

6.2.2 Key Recovery Using Sign Leakage. We can only partially apply the attack in Section 5.2 to Mitaka. Specifically, it is feasible to learn a relatively rough direction, denoted $\mathbf{h}'$, of $\mathbf{h}_0$ through the covariance as in the first step. However, refining $\mathbf{h}'$ via filtering out a slice does not work, as the domain of filtered signatures does not have a clear boundary. Hence we have to use $\mathbf{h}'$ directly for key recovery. We also observed that for Mitaka, using the expectation can give a better approximate direction than using the covariance, which is different from the case in Remark 2. For this, our test used the expectation to get $\mathbf{h}'$ and then used $\mathbf{h}'$ to recover the key. In this way, a practically efficient key recovery needs about 2.25 million signatures. Figure 19 shows the detailed experimental results.

6.2.3 Key Recovery Using Two Leakages. As shown in the last two graphs of Figure 17, combining two leakages allows filtering signatures in two regions. For each region, we can obtain an approximate direction using the approach in the last subsection based on either the expectation or the covariance of filtered signatures. Extensive experiments suggested that using the expectation of the signatures with $y_0 \in (1, +\infty)$ gives the most accurate approximate direction. Different from the case in Section 5.3, using two leakages together does not reduce the number of required signatures so significantly: it still requires about 1.8 million signatures to recover the key in a short time. Detailed experimental results are shown in Figure 20.

7 Conclusion

In this work, we provide an improved power analysis for Falcon. Our first result is a new key recovery using the half Gaussian leakage within the base sampler. It turns out to be much more effective than the existing method [17] in terms of both measurements and computations. Our second result is to show that the

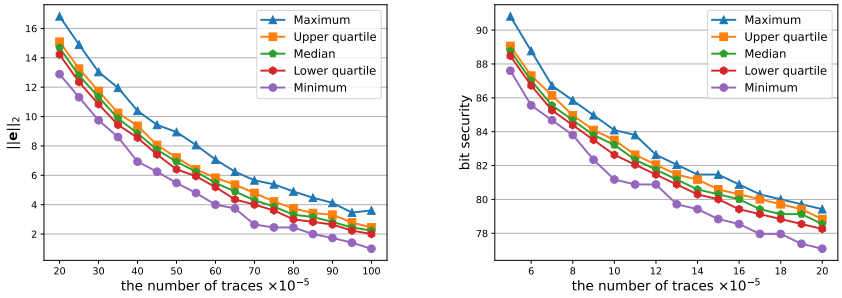

Fig. 18. Experimental results for the attack solely using half Gaussian leakage. The left figure shows the approximate error size $\|\mathbf{e}\|_2$, and the right one shows the bit security estimated as per $\|\mathbf{e}\|_2$. Experiments ran over 40 Mitaka-512 instances.

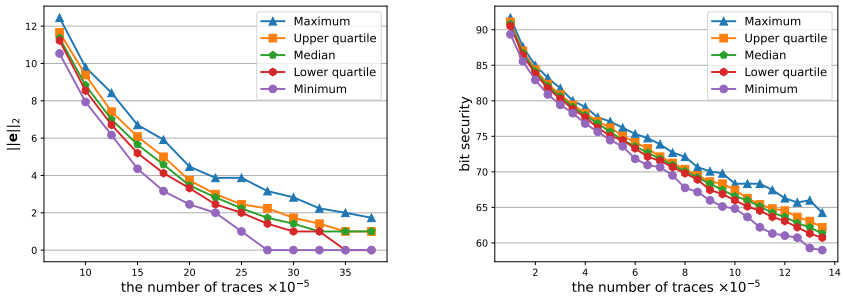

Fig. 19. Experimental results for the attack solely using sign leakage. The left figure shows the approximate error size $\|\mathbf{e}\|_2$, and the right one shows the bit security estimated as per $\|\mathbf{e}\|_2$. Experiments ran over 40 Mitaka-512 instances.

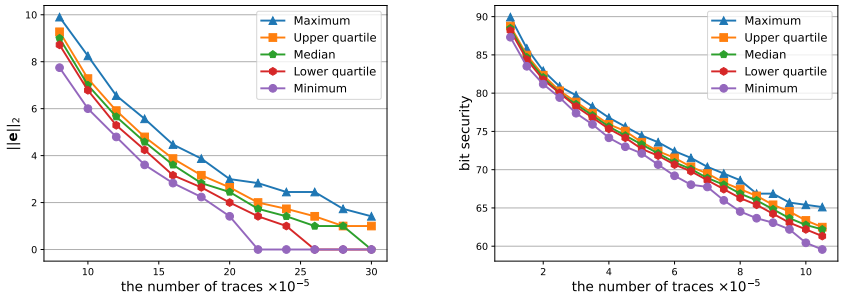

Fig. 20. Experimental results for the attack using both half Gaussian leakage and sign leakage. The left figure shows the approximate error size $\|\mathbf{e}\|_2$, and the right one shows the bit security estimated as per $\|\mathbf{e}\|_2$. Experiments ran over 40 Mitaka-512 instances.

sign leakage within the integer Gaussian sampler also can be well exploited to recover the key. This is the very first side-channel analysis on Falcon taking the sign leakage into account. We also extend our power analysis to the Mitaka signature scheme.

Our attacks are practical and powerful: only tens of thousand traces are enough to greatly weaken the security of Falcon; they can even work when leakage classification is inaccurate. This suggests that two exploited leakages are more dangerous than previously imagined. In addition, though we target the reference implementation of Falcon, the attacks apply to many other implementations including clean PQClean and pqm4 implementations.

With the standardization and deployment of Falcon underway, there is a clear need for side-channel protections. While we have proposed some countermeasure to mitigate the attacks, it cannot completely prevent the leakages. Masking might be a reassuring countermeasure. Despite some efforts [15,1,11], efficient masked implementation of integer Gaussian sampling, particularly for variable and sensitive parameters, remains a challenging problem.

Acknowledgements. The first two authors contributed equally to this work and are considered as co-first authors. We thank Pierre-Alain Fouque, Mélissa Rossi, Zongyue Wang, Yu Yu and Yuanyuan Zhou for valuable discussions. Yang Yu is supported by the National Natural Science Foundation of China (No. 62102216), the Mathematical Tianyuan Fund of the National Natural Science Foundation of China (Grant No. 12226006), the National Key Research and Development Program of China (Grant No. 2018YFA0704701), the Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008) and Major Scientific and Technological Innovation Project of Shandong Province, China (Grant No. 2019JZZY010133). Weijia Wang is supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the National Natural Science Foundation of China (No. 62002204) and the Program of Qilu Young Scholars (Grant No 61580082063088) of Shandong University. This work is also supported by Shandong Key Research and Development Program (Grant No. 2020ZLYS09) and National key research and development program(Grant No. 2022YFB2702804).

A Theoretical Analysis for the LSP algorithm

In this section, we will show that given a sufficiently large polynomial number of samples, our LSP algorithm finds a solution with constant approximation errors with some constant probability. This is formally described in Lemma 1.

Lemma 1. *Given an* $\mathsf{LSP}_{b,\sigma,N}$ *instance with exact solution* $\mathbf{b}$*, let* $\mathbf{b}'$ *be the output by our* LSP *algorithm. Let* $\sigma'^2 = \mathbf{var}\left[x \sim \mathcal{N}(0,\sigma^2) \mid -\frac{b}{\|\mathbf{b}\|} \le x \le \frac{b}{\|\mathbf{b}\|}\right]$*. Then*

$$\|\mathbf{b}-\mathbf{b}'\| \le C_e \cdot \left(\sigma^2 + \frac{\sigma^2}{\sigma^2-\sigma'^2}\cdot\|\mathbf{b}\|\right)\left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right)$$

with a probability $\geq 1 - 4e^{-u}$ *for some constant* C_e.

To prove Lemma 1, we need the following theorems.

Theorem 1 (Weyl's inequality). *Let* $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{n\times n}$ *be symmetric matrices. Then*

$$\max_i |\lambda_i(\mathbf{S}) - \lambda_i(\mathbf{T})| \leq \|\mathbf{S} - \mathbf{T}\|_2.$$

Here $\|\mathbf{S}\|_2$ *denotes the spectral norm of* $\mathbf{S}$.

Theorem 2 (Davis-Kahan [35]). *Let* $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{n\times n}$ *be two symmetric matrices. Suppose that for some* i*, the* i*-th largest eigenvalue of* $\mathbf{S}$ *is well separated from the rest of the spectrum:*

$$\min_{j:j\neq i} |\lambda_i(\mathbf{S}) - \lambda_j(\mathbf{S})| = \delta > 0.$$

Then the unit eigenvectors $v_i(\mathbf{S})$ *and* $v_i(\mathbf{T})$ *satisfies:*

$$\exists \theta \in \{-1, 1\} : \|v_i(\mathbf{S}) - \theta v_i(\mathbf{T})\| \leq \frac{2^{\frac{3}{2}}\|\mathbf{S} - \mathbf{T}\|_2}{\delta}.$$

Theorem 3 (Adapted from Theorem 4.7.1 [35]). *Let* $\mathbf{X}$ *be a sub-gaussian random vector in* $\mathbb{R}^n$ *and* $\Sigma = \mathbf{Cov}[\mathbf{X}\mathbf{X}^t]$*. For independent samples* $\mathbf{X}_1, \cdots, \mathbf{X}_N$*, let* $\Sigma_N = \frac{1}{N}\sum_{i=0}^{N} \mathbf{X}_i\mathbf{X}_i^t$*. Then there exists a constant* K *(related to* $\mathbf{X}$*) and a universal constant* $C > 0$ *such that for any* $u \geq 0$*,*

$$\|\Sigma_N - \Sigma\|_2 \leq CK^2 \left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right) \|\Sigma\|_2$$

with probability at least $1 - 2e^{-u}$

Proof of Lemma 1. Let $\mathbf{v} = \frac{\mathbf{b}}{\|\mathbf{b}\|}$ and $\mathbf{v}' = \frac{\mathbf{b}'}{\|\mathbf{b}'\|}$. We have

$$\mathbf{e} = \mathbf{b} - \mathbf{b}' = \|\mathbf{b}\| \cdot (\mathbf{v} - \mathbf{v}') + (\|\mathbf{b}\| - \|\mathbf{b}'\|) \cdot \mathbf{v}'$$

and then $\|\mathbf{e}\| \leq \|\mathbf{b}\| \cdot \|\mathbf{v} - \mathbf{v}'\| + |\|\mathbf{b}\| - \|\mathbf{b}'\||$.

Let $\Sigma = \mathbf{Cov}[\mathbf{s}|\mathbf{s} \in \mathcal{S}_{\mathbf{b}}(b)]$ and Σ_N be the covariance matrix measured over N given samples. Then $\mathbf{v}$ and $\mathbf{v}'$ are respectively the eigenvectors corresponding to the smallest eigenvalue of Σ and Σ_N. Since the smallest eigenvalue of Σ is σ'^2 and other eigenvalues are σ^2, Theorem 2 shows $\|\mathbf{v} - \mathbf{v}'\| \leq \frac{2^{\frac{3}{2}}\|\Sigma - \Sigma_N\|_2}{\sigma^2 - \sigma'^2}$. By Theorem 3, we have $\|\mathbf{v} - \mathbf{v}'\| \leq 2^{\frac{3}{2}}CK^2\frac{\sigma^2}{\sigma^2 - \sigma'^2} \cdot \left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right)$ with probability at least $1 - 2e^{-u}$.

We next analyse the term $|\|\mathbf{b}\| - \|\mathbf{b}'\||$. Let $\sigma_N'^2$ be the smallest eigenvalue of Σ_N. By Theorems 1 and 3, we have $|\sigma'^2 - \sigma_N'^2| \leq \|\Sigma_N - \Sigma\|_2 \leq \sigma^2 CK^2 \left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right)$ with probability at least $1 - 2e^{-u}$. Our LSP algorithm uses the equation

$$\sigma'^2 = \frac{\int_{-\frac{b}{\|\mathbf{b}\|}}^{\frac{b}{\|\mathbf{b}\|}} x^2 \exp(-\frac{x^2}{2\sigma^2})dx}{\int_{-\frac{b}{\|\mathbf{b}\|}}^{\frac{b}{\|\mathbf{b}\|}} \exp(-\frac{x^2}{2\sigma^2})dx}$$

to estimate $\|\mathbf{b}\|$, hence $\|\mathbf{b}\| = f(\sigma'^2)$ for some continuous function f determined by b, σ and σ'. Accordingly, $\|\mathbf{b}'\| = f(\sigma_N'^2)$. Therefore, there exists a constant C' such that $|\|\mathbf{b}\| - \|\mathbf{b}'\|| \leq C'|\sigma'^2 - \sigma_N'^2| \leq \sigma^2 CC'K^2 \left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right)$. So far, we prove that $\|\mathbf{e}\| \leq C_e \left(\sigma^2 + \frac{\sigma^2}{\sigma^2 - \sigma'^2} \cdot \|\mathbf{b}\|\right) \left(\sqrt{\frac{n+u}{N}} + \frac{n+u}{N}\right)$ with probability at least $1 - 4e^{-u}$ where $C_e = \max\{CC'K^2, 2^{\frac{3}{2}}CK^2\}$. □

B Attacks on Other Falcon Parameters

Our attacks easily apply to other Falcon parameter sets. The base sampler and the integer Gaussian sampler are exactly the same for different n, thus both leakages can be measured in the same way. Table 1 shows the experimental data for $n = 256, 512, 1024$, where $N_1(x)$ is the number of required traces to get an approximate error of hamming weight $\leq x$ with probability $\approx \frac{1}{4}$.

Table 1. Experimental data of $N_1(x)$ measured over 40 Falcon instances for each n. The item "A/B/C" represents the values for $n = 256/512/1024$.

	Half Gaussian leakage	Sign leakage	Two leakages
$N_1(0) \times 10^{-3}$	270 / 360 / 400	210 / 230 / 280	60 / 70 / 75
$N_1(5) \times 10^{-3}$	200 / 230 / 270	142.5 / 175 / 203	39 / 47 / 56
$N_1(7) \times 10^{-3}$	182.5 / 217 / 255	132.5 /164 / 188	37 / 45 / 54
$N_1(9) \times 10^{-3}$	175 / 201 / 240	127 / 155 / 184	35 / 43 / 50

References

1. Barthe, G., Belaïd, S., Espitau, T., Fouque, P.A., Rossi, M., Tibouchi, M.: Galactics: Gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisited. In: ACM CCS 2019. pp. 2147–2164 (2019). https://doi.org/10.1145/3319535.3363223
2. Bootle, J., Delaplace, C., Espitau, T., Fouque, P.A., Tibouchi, M.: LWE without modular reduction and improved side-channel attacks against BLISS. In: ASIACRYPT 2018. pp. 494–524 (2018). https://doi.org/10.1007/978-3-030-03326-2_17
3. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. vol. 2523, pp. 13–28. Springer (2002). https://doi.org/10.1007/3-540-36400-5_3
4. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with side information: attacks and concrete security estimation. In: CRYPTO 2020. pp. 329–358 (2020). https://doi.org/10.1007/978-3-030-56880-1_12

5. Ducas, L., Galbraith, S., Prest, T., Yu, Y.: Integral matrix gram root and lattice gaussian sampling without floats. In: EUROCRYPT 2020. pp. 608–637 (2020). https://doi.org/10.1007/978-3-030-45724-2_21
6. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: ASIACRYPT 2014. pp. 22–41 (2014). https://doi.org/10.1007/978-3-662-45608-8_2
7. Ducas, L., Nguyen, P.Q.: Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In: ASIACRYPT 2012. pp. 433–450 (2012). https://doi.org/10.1007/978-3-642-34961-4_27
8. Ducas, L., Prest, T.: Fast Fourier Orthogonalization. In: ISSAC 2016. pp. 191–198 (2016). https://doi.org/10.1145/2930889.2930923
9. Ducas, L., Yu, Y.: Learning strikes again: the case of the drs signature scheme. Journal of Cryptology **34**, 1–24 (2021). https://doi.org/10.1007/s00145-020-09366-9
10. 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: ACM CCS 2017. pp. 1857–1874 (2017). https://doi.org/10.1145/3133956.3134028
11. Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., Yu, Y.: MITAKA: A Simpler, Parallelizable, Maskable Variant of FALCON. In: Eurocrypt 2022 (2022). https://doi.org/10.1007/978-3-031-07082-2_9
12. Fouque, P.A., Gérard, F., Rossi, M., Yu, Y.: Zalcon: an alternative FPA-free NTRU sampler for Falcon. In: Proc. 3rd NIST PQC Workshop. pp. 1–23 (2021)
13. Fouque, P.A., Kirchner, P., Tibouchi, M., Wallet, A., Yu, Y.: Key Recovery from Gram–Schmidt Norm Leakage in Hash-and-Sign Signatures over NTRU Lattices. In: EUROCRYPT 2020. pp. 34–63 (2020). https://doi.org/10.1007/978-3-030-45727-3_2
14. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206 (2008). https://doi.org/10.1145/1374376.1374407
15. Gérard, F., Rossi, M.: An efficient and provable masked implementation of qtesla. In: CARDIS 2019. pp. 74–91 (2019). https://doi.org/10.1007/978-3-030-42068-0_5
16. Groot Bruinderink, L., Hülsing, A., Lange, T., Yarom, Y.: Flush, gauss, and reload–a cache attack on the BLISS lattice-based signature scheme. In: CHES 2016. pp. 323–345 (2016). https://doi.org/10.1007/978-3-662-53140-2_16
17. 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). https://doi.org/10.46586/tches.v2022.i3.141-164
18. Howe, J., Prest, T., Ricosset, T., Rossi, M.: Isochronous Gaussian Sampling: From Inception to Implementation. In: PQCrypto 2020. pp. 53–71 (2020). https://doi.org/10.1007/978-3-030-44223-1_4
19. Hulsing, A., Bernstein, D.J., Dobraunig, C., Eichlsedera, M., Fluhrer, S., Gazdag, S.L., Kampanakis, P., Kolbl, S., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Aumasson, J.P., Westerbaan, B., Beullens, W.: SPHINCS+: Submission to the NIST's post-quantum cryptography standardization process (2020), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

20. Karabulut, E., Aysu, A.: Falcon Down: Breaking Falcon Post-Quantum Signature Scheme through Side-Channel Attacks. In: DAC 2021. pp. 691–696 (2021). https://doi.org/10.1109/DAC18074.2021.9586131
21. Kim, S., Hong, S.: Single trace analysis on constant time CDT sampler and its countermeasure. Applied Sciences **8**(10), 1809 (2018). https://doi.org/10.3390/app8101809
22. Klein, P.N.: Finding the closest lattice vector when it's unusually close. In: SODA 2000. pp. 937–941 (2000)
23. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: ASIACRYPT 2009. pp. 598–616 (2009). https://doi.org/10.1007/978-3-642-10366-7_35
24. Lyubashevsky, V.: Lattice signatures without trapdoors. In: EUROCRYPT 2012. pp. 738–755 (2012). https://doi.org/10.1007/978-3-642-29011-4_43
25. Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D., Bai, S.: Dilithium: Submission to the NIST's post-quantum cryptography standardization process (2020), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
26. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks - revealing the secrets of smart cards. Springer (2007). https://doi.org/10.1007/978-0-387-38162-6
27. Migliore, V., Gérard, B., Tibouchi, M., Fouque, P.A.: Masking Dilithium. In: ACNS 2019. pp. 344–362 (2019). https://doi.org/10.1007/978-3-030-21568-2_17
28. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: EUROCRYPT 2006. pp. 271–288 (2006). https://doi.org/10.1007/11761679_17
29. Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: CRYPTO 2010. pp. 80–97 (2010). https://doi.org/10.1007/978-3-642-14623-7_5
30. Pessl, P., Bruinderink, L.G., Yarom, Y.: To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures. In: ACM CCS 2017. pp. 1843–1855 (2017). https://doi.org/10.1145/3133956.3134023
31. Pornin, T.: New Efficient, Constant-Time Implementations of Falcon. Cryptology ePrint Archive, Report 2019/893 (2019), https://ia.cr/2019/893
32. Prest, T.: Gaussian Sampling in Lattice-Based Cryptography. Ph.D. thesis, École Normale Supérieure, Paris, France (2015)
33. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Submission to the NIST's post-quantum cryptography standardization process (2020), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
34. Tibouchi, M., Wallet, A.: One bit is all it takes: a devastating timing attack on BLISS's non-constant time sign flips. Journal of Mathematical Cryptology **15**(1), 131–142 (2021). https://doi.org/10.1515/jmc-2020-0079
35. Vershynin, R.: High-dimensional probability: An introduction with applications in data science, vol. 47. Cambridge university press (2018). https://doi.org/10.1080/14697688.2020.1813475
36. Wisiol, N., Gersch, P., Seifert, J.: Cycle-accurate power side-channel analysis using the chipwhisperer: a case study on gaussian sampling. IACR Cryptol. ePrint Arch. p. 903 (2022), https://eprint.iacr.org/2022/903
37. Yu, Y., Ducas, L.: Learning strikes again: the case of the DRS signature scheme. In: ASIACRYPT 2018. pp. 525–543 (2018). https://doi.org/10.1007/978-3-030-03329-3_18