Crowhammer: Full Key Recovery Attack on FALCON with a Single Rowhammer Bit Flip

Calvin Abou Haidar[1], Quentin Payet[2⋆], and Mehdi Tibouchi[1]

[1] NTT Social Informatics Laboratories, Japan
{calvin.haidar,mehdi.tibouchi}@ntt.com
[2] CentraleSupelec, France
quentin.payet@student-cs.fr

Abstract. The Rowhammer attack is a fault-injection technique leveraging the density of RAM modules to trigger persistent hardware bit flips that can be used for probing or modifying protected data. In this paper, we show that FALCON, the hash-and-sign signature scheme over NTRU lattices selected by NIST for standardization, is vulnerable to an attack using Rowhammer.

FALCON's Gaussian sampler is the core component of its security, as it allows to provably decorrelate the short basis used for signing and the generated signatures. Other schemes, lacking this guarantee (such as NTRUSign, GGH or more recently PEREGRINE) were proven insecure. However, performing efficient and secure lattice Gaussian sampling has proved to be a difficult task, fraught with numerous potential vulnerabilities to be exploited. To avoid timing attacks, a common technique is to use distribution tables that are traversed to output a sample. The official FALCON implementation uses this technique, employing a hardcoded reverse cumulative distribution table (RCDT). Using Rowhammer, we target FALCON's RCDT to trigger a very small number of targeted bit flips, and prove that the resulting distribution is sufficiently skewed to perform a key recovery attack.

Namely, we show that a *single* targeted bit flip suffices to fully recover the signing key, given a few hundred million signatures, with more bit flips enabling key recovery with fewer signatures. Interestingly, the Nguyen–Regev parallelepiped learning attack that broke NTRUSign, GGH and PEREGRINE does not readily adapt to this setting unless the number of bit flips is very large. However, we show that combining it with principal component analysis (PCA) yields a practical attack.

This vulnerability can also be triggered with other types of persistent fault attacks on memory like optical faults. We suggest cheap countermeasures that largely mitigate it, including rejecting signatures that are unusually short.

1 Introduction

Rowhammer. The Rowhammer attack is a hardware vulnerability that exploits the unintended electrical interference between adjacent rows in DRAM (Dynamic Random-Access Memory). Discovered by Kim et al. [33], Rowhammer occurs when repeatedly accessing (or "hammering") a specific row in memory induces bit flips in neighboring rows, potentially allowing for privilege escalation or data corruption. Since its discovery, various attack variants have been demonstrated, including remote Rowhammer exploits [25,26,58] and cross-VM attacks in cloud environments [65]. Despite mitigation efforts such as ECC (Error-Correcting Code) memory and refresh rate increases, new attack techniques continue to bypass defenses [22], making Rowhammer a persistent concern in modern computing system security.

Rowhammer has long been successfully used in fault attacks against symmetric as well as asymmetric cryptographic schemes, including the AES [67], RSA signatures [3, 57, 63] and ECDSA/EdDSA [42, 48]. More recently, a handful of papers have considered Rowhammer attacks against postquantum cryptosystems submitted to the NIST standardization process: multivariate signature scheme LUOV [43], lattice-based signature Dilithium [1,31], and key encapsulation mechanisms FrodoKEM [19], Kyber and BIKE [1].

The attacks on classical schemes tend to be fairly direct, necessitating only a few Rowhammer fault injections and resulting in immediate key exposure afterwards: e.g., Razavi et al. [57] flip some bits of an RSA modulus

⋆ Work carried out as part of the author's internship at the NTT Social Informatics Laboratories.

making it easy to factor with good probability, Weissman et al. [63] induce Bellcore-style faults on RSA signatures enabling the classic GCD key recovery, etc. In contrast, the postquantum attacks tend to be more contrived: for example, the attacks against LUOV and Dilithium in [31,43] involve thousands of Rowhammer bit flips in many successive signature generations.

Of particular interest are the attacks on FrodoKEM in [19] and Kyber in [1] which only involve a few Rowhammer bit flips (8 bit flips for FrodoKEM and 2 bit flips for Kyber) in a one-time phase within key generation, in order to produce faulty key pairs that appear to work normally but have much higher decryption error probabilities than validly generated keys. This makes it possible to break those keys later on using decryption failure attacks. While the small number of bit flips and the one-time nature of the fault injection make these attacks reasonably practical, they have to be completed within the time frame of key generation and the bit flips have to be injected at specific positions in memory, making them somewhat challenging still.

In any case, as the standardization and deployment of postquantum schemes progresses, those recent attacks demonstrate the importance of assessing the security of those schemes with respect to Rowhammer-based attacks, which remain highly relevant to software implementations on modern CPU architectures, especially in cloud environments.

FALCON *and the pitfalls of hash-and-sign lattice-based signatures.* Since Rowhammer-based attacks have been considered against Kyber (a.k.a. ML-KEM) [59] and Dilithium (a.k.a. ML-DSA) [40], two of the lattice-based schemes already selected by NIST for standardization, it is natural to ask about the third one, FALCON (the future FN-DSA) [53].

Whereas Dilithium follows Lyubashevky's "Fiat–Shamir with aborts" [38, 39] paradigm for constructing lattice-based signatures, FALCON is a modern instantiation of the other main paradigm: hash-and-sign signatures based on lattice trapdoors.

In lattice-based hash-and-sign signatures, the signing is a *good basis* of some full-rank lattice, which allows to find relatively close lattice vectors to arbitrary points in the ambient space of the lattice. The verification key is a *bad basis* of the same lattice (typically the Hermite Normal Form in the case of sublattices of $\mathbb{Z}^n$), which makes it possible to check lattice membership, but does not allow to find close lattice vectors to random targets outside the lattice. Then, to sign a message, one hashes that message to a target in the ambient space, finds a close vector to the target, and outputs the difference as the signature. The verification algorithm then checks that the signature is a sufficiently short vector, and that its difference with the message hash is indeed in the lattice.

This idea was first considered in the late 1990s, resulting in the GGH [24] and NTRUSign [29] signature schemes. In those schemes, the decoding step of selecting a close lattice vector to the target message digest is carried out in a deterministic way, using Babai's nearest plane algorithm [2]. Unfortunately, it turns out that this makes the scheme insecure, as the distribution of the resulting signatures directly depends on the (Gram–Schmidt Orthogonalization of) the secret "good basis". Nguyen and Regev [44] showed how statistical techniques (Frieze et al.'s algorithm for learning a linear transformation [21]) would then effectively recover the signing key from a few tens of thousands of signatures. Attempts at mitigating the problem in heuristic ways (particularly in patched versions of NTRUSign) were also broken using similar techniques [11], as were more recent proposed hash-and-sign constructions that did not specifically make the signature distribution independent of the trapdoor, including DRS [13], PEREGRINE [36] and EHTv3 [51].

The first provably secure solution came from Gentry, Peikert and Vaikuntanathan [23], who showed how Klein's randomized version of Babai's nearest plane algorithm [34] could be used to ensure that signatures follow a discrete Gaussian distribution depending only on the lattice itself, and not the trapdoor basis used for signing. This approach is now known as the GPV framework, and has been efficiently instantiated over NTRU lattices by Ducas, Lyubashevky and Prest (DLP) [10].

FALCON is a faster improvement of the DLP scheme that achieves lattice Gaussian sampling over the NTRU lattice by randomizing not the (quadratic time) Babai nearest plane algorithm itself, but a quasilinear time variant of it for structured lattices with cyclotomic symmetries: the fast Fourier nearest plane algorithm of Ducas and Prest [12]. Several other designs combining different choices of lattice structure and lattice Gaussian samplers (including Pekert's sampler [46] or Prest's hybrid sampler [52]) have also been subsequently considered, such as MODFALCON [6], MITAKA [16], ANTRAG [17], HUFU [66], SQUIRRELS [18], and more [5], but FALCON stands out as the most prominent by far.

FALCON features small key and signature sizes, very fast verification, and excellent signing efficiency on platforms with fast floating point arithmetic, making it a particularly attractive design. It is however notoriously difficult to implement correctly and securely. For example, early versions of FALCON relied on a variable-time one-dimensional Gaussian sampler, leading to structural leakage of key information [20]. This was subsequently fixed [30,49], but it turned out that the first version of the new, constant-time implementation had an incorrect implementation of the one-dimensional sampler, with a non-Gaussian distribution [50]. That buggy implementation also caused signatures to leak information about the trapdoor.

While FALCON is now free of such bugs, its Gaussian sampler has been shown to be vulnerable to side-channel leakage [27, 32, 68] in ways that are difficult to protect from, since countermeasures like masking seem difficult to adopt, in view, among other issues, of FALCON's reliance on floating-point arithmetic. This use of floating-point arithmetic, incidentally, has also recently been shown to be a source of other potential security vulnerabilities [37], although they are mostly a concern for derandomized variants of FALCON, or in the presence of repeated randomness.

Interestingly, while many fault attacks have been proposed against other lattice-based schmes like Kyber, Dilithium and even DLP [4, 9, 15, 28, 47, 54–56, 64], they have not appeared to be a major concern for FALCON. We are only aware of *one* proposed fault attack against it [41], which attempts to adapt the loop-abort technique of Espitau et al. [14] to FALCON's signature generation. However, due to the recursive, tree-like structure of the FALCON sampler, a successful attack requires multiple synchronized instruction skipping faults in order to exit the signature generation sufficiently early to achieve a feasible key recovery. It is doubtful whether actually mounting such an attack in practice and collecting sufficiently many faulty signatures is possible at all. In that sense, it seems like the complexity of FALCON's algorithm has deterred rather than enabled fault analysis so far.

Contributions and technical overview of this paper. In this paper, we present the first Rowhammer-based fault attack against FALCON (and possibly the first fault attack at all on FALCON in a realistic fault model). The idea is to perturb the one-dimensional Gaussian sampler upon which FALCON's lattice Gaussian sampler is built, by flipping a few bits to zero in the reverse cumulative distribution table (RCDT) describing the half-Gaussian base sampler. This amounts to artificially introducing a bug similar to the one that inadvertantly occurred in the first 2019 constant-time implementation of FALCON by Pornin [50].

This one-time fault injection causes all subsequently generated signatures (which, nonetheless, remain valid FALCON signatures) to present a slight dependency on the secret signing key. We show that collecting sufficiently many of those signatures makes it possible to fully recover the secret signing key.

In order to mount this key recovery, we first give a simple description, under mild heuristic conditions, of the output distribution of a perturbed version of the Klein–GPV sampler when the one-dimensional Gaussian distributions are replaced with some other, not necessarily Gaussian distributions. The same description also applies to the FALCON's fast Fourier sampler, with the twist that one has to consider the Gram–Schmidt orthogonalization of the secret basis in *bit-reversed order*.

Based on that description, we can deduce that recovering the signing key from signatures generated after the Rowhammer fault is an instance of the *hidden transformation problem* introduced in [36], which can in principle be solved using gradient descent techniques essentially identical to those of Nguyen and Regev [44, 45]. This does work well if *lots* of bits of the RCDT are zeroed out (for example if they are all set to zero), but it is unlikely that Rowhammer can achieve such an extreme amount of directional bit flips. However, for more realistic bit flip patterns of say 8 bits or fewer (as considered in [1, 19]), the bias in signatures is less pronounced, and appears to require simply too many samples for the Nguyen–Regev attack to be feasible, in view of its substantial time and space complexity (the storage of the signatures and the recomputation of the gradient at each step are expensive!).

One can however considerably improve the attack by combining it with *principal component analysis* (PCA). Indeed, based on the description of the output distribution of the faulty lattice Gaussian sampler, we find that the vectors of the Gram–Schmidt orthogonalization (GSO) of the trapdoor (in bit-reversed order) are eigenvectors of its correlation matrix $\tilde{\boldsymbol{\Sigma}}$. For example, the longest vector in that GSO (which is either $(g, -f)$ or the projection of $(G, -F)$ orthogonally to $(g, -f)$ over the cyclotomic field) will be an eigenvector for either the largest or the smallest eigenvalue of $\tilde{\boldsymbol{\Sigma}}$. Therefore, one can try to recover them by generating many signatures, computing the largest or smallest eigenvalue of the associated sample correlation matrix, and hope that the corresponding

eigenvector reveals the key. This approach does not work directly, though: the eigenvectors fail to converge, because the structure of the bit-reversed order GSO causes the eigenspaces to be of dimension at least 2.

There are various ways of circumventing that difficulty. One approach is to notice that the structure which yields those two-dimensional eigenspaces is the fact that the GSO is actually defined over $\mathbb{Q}(i)$, so that the correlation matrix $\tilde{\boldsymbol{\Sigma}}$ can in fact be seen as a Hermitian matrix of half the dimension, which will typically have distinct eigenvalues, and Hermitian PCA will therefore converge to a top or bottom eigenvector revealing the key up to multiplication by a complex constant, which can be recovered using a simple circle search.

Another approach, which requires fewer signatures, is to use the PCA for dimension reduction before applying the Nguyen–Regev attack: one first identifies a subspace of relatively small dimension containing a close approximation of the key (namely, the sum of the top few eigenspaces of the sample correlation matrix), and then carries out an Nguyen–Regev-style optimization within that small-dimensional subspace, which is much less costly. This approach can be combined with the aforementioned Hermitian PCA for maximum efficiency.

Finally, although this point is not elaborated further in the paper, we also note that it is possible to cast the problem in lattice terms, at least when the largest GSO vector is $(g, -f)$ itself. In that case, learning a subspace that mostly contains it can be interpreted as learning a different Euclidean norm on the ambient space of the public lattice with respect to which $(g, -f)$ has almost the same length, but the volume of the lattice is considerably larger (equivalently, this can be seen as a collection of *a posteriori hints* in the language of [7]), possibly making lattice reduction attacks feasible. Based on rough root Hermite factor estimates, this approach appears to be substantially more computationally expensive than the purely statistical ones, at least for our parameters of interest, so we focus on those instead.

All in all, these techniques allow us to leverage the Rowhammer fault injection attack into a full key recovery, even with a *single* Rowhammer bit flip to zero on a well-chosen bit of the RCDT, namely, the most significant bit of the first coefficient. After this single-bit fault, collecting 200 million signatures suffices to achieve over 70% success probability of full key recovery, by combining the Hermitian PCA to recover a subspace of complex dimension 8 (real dimension 16) mostly containing the secret, and then applying Nguyen–Regev on this small subspace.

Strictly speaking, our attack validation is not fully end-to-end: we separately verify the fact that the Rowhammer one-bit faults can be injected successfully on one machine (which is somewhat older and has DDR3 memory), and use a different machine (a powerful computation server) to generate the signatures using a modified FALCON implementation with the hardcoded bit flip. This is mainly to avoid having to replicate well-trodden territory like memory massaging, and also prevent data corruption on the expensive server, while still having fast turn around times for experimental results. We believe that our experiments nevertheless suffice to establish that our attack is quite realistic.

With a larger number of bit flips, we can substantially reduce the number of required signatures; for example, with 8 targeted bit flips as in the attack against FrodoKEM [19], the number of required signatures decreases to about 20 millions.

We conclude by pointing out that, aside from generic Rowhammer and fault mitigations, simple, common sense countermeasures can largely avoid the type of attacks we consider at essentially no cost. In particular, seeing as the length of validly generated signatures in FALCON concentrates heavily around its expected value, we suggest rejecting not only signatures that are unusually long, but also those that are unusually short (both during signature generation and verification). This is basically for free (it would have only negligible impact on repetition probability in signing), and while it doesn't guarantee that the signatures do not leak secret key information, it is hard to imagine how a fault attack like ours could pass this test while still allowing key recovery from a non-astronomical number of signatures.

Organization of the paper. We start in Section 2 by introducing notations and background. In Section 3, we briefly present how the FALCON signature works and focus on its Gaussian sampler and the distributions of FALCON signatures. We introduce a generalized framework of Nearest Plane bases samplers in Section 4, in order to capture the behaviour of the FALCON sampler with a faulty integer Gaussian sampler. In Section 5, we give a step-by-step analysis of our attack and present our results in Section 6

2 Preliminaries

2.1 Notation

Vectors are represented in bold lowercase letters, and the i-th coordinate of a vector $\mathbf{b}$ is written as b_i, i.e., $\mathbf{b} = (b_1, \ldots, b_n)$. The inner product of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ is $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i b_i$.

The ℓ_2-norm of a vector $\mathbf{a} \in \mathbb{R}^n$ is $\|\mathbf{a}\| = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}$, the ℓ_1-norm is $\|\mathbf{a}\|_1 = \sum_i |a_i|$, and the ℓ_∞-norm is $\|\mathbf{a}\|_\infty = \max_i |a_i|$. For any matrix $\mathbf{A}$, we write $\|\mathbf{A}\|_F = \mathrm{Tr}(\mathbf{A}\mathbf{A}^T) = \sum_{i,j} \mathbf{A}_{i,j}^2$ and $\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2 = 1} \mathbf{A}\mathbf{x}$.

Matrices are represented by bold uppercase letters. The i-th column of matrix $\mathbf{B}$ is denoted $\mathbf{b}_i$, i.e., $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$. The transpose and inverse of matrix $\mathbf{B}$ are represented as $\mathbf{B}^T$ and $\mathbf{B}^{-1}$ respectively. The identity matrix in dimension n is denoted as $\mathbf{I}_n$, or $\mathbf{I}$ when the dimension is clear from the context.

2.2 Lattices

A lattice L is a discrete subgroup of $\mathbb{R}^m$. It is the set of all integer combinations of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, i.e.,

$$L = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}.$$

The matrix $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called the basis, and n is the rank of L. When $n = m$, the lattice is said to be full-rank. We denote the lattice generated by a basis $\mathbf{B} \in \mathrm{GL}_n(\mathbb{R})$ as $L(\mathbf{B})$.

GSO and LDL^T decomposition. Every full rank matrix $\mathbf{B} \in \mathcal{R}^{n \times m}$ admits a unique decomposition (called GSO decomposition)

$$\mathbf{B} = \mathbf{L}\tilde{\mathbf{B}}$$

where $\mathbf{L}$ is unit lower triangular and $\tilde{\mathbf{B}}$ is pairwise orthogonal.

Their Gram matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^T$ admits a unique decomposition (called LDL^T decomposition)

$$\mathbf{G} = \mathbf{L}\mathbf{D}\mathbf{L}^T$$

where $\mathbf{L}$ is unit lower triangular and $\mathbf{D}$ is a positive diagonal matrix.

The two decomposition are linked since as $\tilde{\mathbf{B}}$ is pairwise orthogonal, the product $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$ is diagonal. Thus $\mathbf{L}\tilde{\mathbf{B}}$ is the GSO decomposition of $\mathbf{B}$ if and only if $\mathbf{L}(\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T)\mathbf{L}^T$ is the LDL^T decomposition of $\mathbf{B}\mathbf{B}^T$.

Notice that since $\tilde{\mathbf{B}}$ is pairwise orthogonal, one can decompose it as

$$\tilde{\mathbf{B}} = \tilde{\mathbf{D}}\mathbf{U}$$

where $\tilde{\mathbf{D}}$ is a diagonal matrix and $\mathbf{U}$ is orthogonal (i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$). The GSO decomposition of $\mathbf{B}$ can then be written as $\mathbf{B} = \mathbf{L}\tilde{\mathbf{D}}\mathbf{U}$.

2.3 Statistics and Probability

For a distribution D, we write $y \leftarrow D$ when the random variable y is sampled from D. We also write $y \sim D$ to indicate that y follows the distribution D. Let $U(S)$ represent the uniform distribution over the set S, and $\#S$ represent the number of elements in S. The expectation of a random variable y is denoted $\mathbb{E}[y]$. A distribution D over $\mathbb{R}$ is called centered when $\mathbb{E}_{y \leftarrow D}[y] = 0$. For a distribution D over $\mathbb{R}^n$, the covariance matrix is $\mathrm{Cov}[D] = \mathbb{E}_{\mathbf{x} \leftarrow D}[\mathbf{x}\mathbf{x}^T]$.

We denote by $\mathcal{D}_{\boldsymbol{\Sigma}, \mathbf{c}}$ the discrete Gaussian distribution over $\mathbb{Z}^m$ of center $\mathbf{c}$ and covariance matrix $\boldsymbol{\Sigma}$. Whenever the center is 0, we use the notation $\mathcal{D}_{\boldsymbol{\Sigma}}$.

2.4 Cyclotomic Rings and NTRU

Let $\mathcal{R}_n = \mathbb{Z}[x]/(x^{n/2}+1)$ where $n \geq 4$ is a power of 2. We write $\mathcal{R}$ for $\mathcal{R}_n$ whenever n is clear from the context. Given $h \in \mathcal{R}$ and a rational prime q such that h is invertible modulo q, the lattice $L_{\text{NTRU}} = \{(s_1, s_2) \in \mathcal{R}^2 \mid s_1 + s_2 h = 0 \mod q\}$ is called an NTRU lattice. In a typical NTRU cryptosystem, the public key is $h = g/f \mod q$, where (f, g) is a pair of short polynomials in $\mathcal{R}$ used as the secret key.

For short $(F, G) \in \mathcal{R}^2$ such that $fG - gF = 0 \bmod q$, the matrix

$$\mathbf{B}_{f,g} = \begin{pmatrix} g & -f \\ G & -F \end{pmatrix} \in \mathcal{R}^{2\times 2}$$

is an NTRU trapdoor basis of L_{NTRU}.

3 FALCON and its sampler

In this section, we describe how the FALCON signature scheme works. We focus mostly on the Gaussian sampler using in the signing procedure, as this will be the main target of our attacks. For simplicity, we do not mention all the steps linked to the compression of the signature and avoid using the FFT-like representation that is used to speed up polynomial multiplication. This FFT-like representation, which we call bit-reversed order, is a morphism $\mathsf{R} : \mathbb{R}^n \to \mathbb{R}^n$ such that $\mathsf{R}(x_0, \ldots, x_{n-1}) \mapsto (x_{\mathsf{rev}(0)}, \ldots, x_{\mathsf{rev}(n-1)})$, with rev being a bit transformation that reverses the $\log(n)$ bits of the indices. We extend the morphism to matrices by setting $\mathsf{R}(\mathbf{M})$ to be the matrix where each line is the application of the morphism R to the corresponding line of matrix $\mathbf{M}$.

In FALCON, a secret key is comprised of four small polynomials $f, g, F, G \in \mathcal{R}$ satisfying the NTRU equation $fG - Fg = 0 \bmod q$. These polyonomials define a matrix

$$\mathbf{B}_{f,g} = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$$

that form the basis of a free $\mathcal{R}$-module of rank 2. The public key is defined as $h = gf^{-1} \bmod q$. In the following, we use the fact that $\mathbf{B}_{f,g}$ can be embedded in $\mathbb{Z}^{2n\times 2n}$ using the embedding $e : v = \sum a_i x^i \in \mathcal{R} \mapsto (a_i) \in \mathbb{Z}^n$ as

$$\mathbf{B} = \begin{bmatrix} e(g) & e(-f) \\ e(x^{\mathsf{rev}(1)} \cdot g) & e(x^{\mathsf{rev}(1)} \cdot -f) \\ \cdots & \\ e(x^{\mathsf{rev}(n-1)} \cdot g) & e(x^{\mathsf{rev}(n-1)} \cdot -f) \\ e(G) & e(-F) \\ e(x^{\mathsf{rev}(1)} \cdot G) & e(x^{\mathsf{rev}(1)} \cdot -F) \\ \cdots & \\ e(x^{\mathsf{rev}(n-1)} \cdot G) & e(x^{\mathsf{rev}(n-1)} \cdot -F) \end{bmatrix}$$

and work with the bit-reversed embedded matrix $\mathbf{B}$ in $\mathbb{Z}^{2n\times 2n}$ instead of working over $\mathcal{R}$, except when explicitly mentioned.

A specificity of FALCON is that together with the secret key, a binary tree T, called a "FALCON tree" is computed. This tree provides a compact representation of the $\mathbf{LDL}^T$ decomposition of the Gram matrix $\mathbf{BB}^T$. This representation is computed by leveraging the structure of the ring $\mathcal{R}$, and allows to perform a recursive variant of Babai's nearest plane (called fast Fourier nearest plane) algorithm that makes full use of the ring structure. FALCON's sampler is a randomized version of this fast Fourier nearest plane algorithm. Inner nodes of the tree contain a representation of $\mathbf{L}$, while the leaves contain the values $\sigma_i = \sigma/\|\tilde{\mathbf{b}}_i\|$, with σ a parameter of FALCON. Note that the matrix $\mathbf{D}$ of the $\mathbf{LDL}^T$ decomposition of $\mathbf{BB}^T$ consists of the squared norms of the GSO vectors $\tilde{\mathbf{b}}_i$, so the leaves σ_i are related to the diagonal elements of $\mathbf{D}$ by $\mathrm{diag}(\sigma_0^2, \ldots, \sigma_{2n-1}^2) = \sigma^2 \mathbf{D}^{-1}$.

Signing procedure. To sign a message m, one starts by hashing it to a point $(c, 0) \in \mathcal{R}^2$ (using salt r) and computing the preimage $\mathbf{t} = e((c, 0)) \cdot \mathbf{B}^{-1}$. Then, one samples a vector $\mathbf{z}$ by running the sampler described in Algorithm 1 on input the vector $\mathbf{t}$ and the FALCON tree T. One can then compute a preimage of $(c, 0)$ by setting

$\mathbf{s} = (\mathbf{t} - \mathbf{z}) \cdot \mathbf{B}$. If the vector $\mathbf{s}$ is shorter than a specified bound, the final signature $(r, \mathbf{s})$ is outputed. If not, the sampling step is run until a short solution is found.

Fast Fourier Sampling. The fast Fourier sampler is a randomized variant of Babai's nearest plane algorithm introduced in [12]. The key difference is that it exploits the structure of the tower of rings $\mathbb{Z} = \mathcal{R}_1 \subset \mathcal{R}_2 \cdots \subset \mathcal{R}_{n/2} \subset \mathcal{R}_n$. The sampler ffSampler is presented as Algorithm 1. It takes as input a target center $\mathbf{t} \in \mathbb{Z}^n$, a FALCON Tree T and outputs a sample $\mathbf{z}$ following the distribution $\mathcal{D}_{\boldsymbol{\Sigma},\mathbf{t}}$, with $\boldsymbol{\Sigma} = \mathbf{L}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-T}$. The sampler recurses over the ring dimension and uses SamplerZ (Algorithm 2) to handle sampling over integers. Over $\mathbb{Z}$, Algorithm 2 relies on rejection sampling using a function RejSamp (which we do not explicitly define since it is not needed here) and a half-Gaussian sampler (Algorithm 3).

3.1 Distribution of signatures

Signatures in FALCON are of the form $\mathbf{s} = (\mathbf{t} - \mathbf{z}) \cdot \mathbf{B} = \mathbf{x}\mathbf{B}$. The vector $\mathbf{x} = \mathbf{t} - \mathbf{z}$, as per the sampler's construction, follows the distribution $\mathcal{D}_{\boldsymbol{\Sigma}}$, with $\boldsymbol{\Sigma} = \mathbf{L}^{-T}\mathrm{diag}(\sigma_0^2, \ldots, \sigma_{2n-1}^2) \cdot \mathbf{L}^{-1} = \mathbf{L}^{-T} \cdot \sigma^2\mathbf{D}^{-1} \cdot \mathbf{L}^{-1}$. The signature then follows the distribution

$$\mathbf{s}\mathbf{B} \sim \mathcal{D}_{(c,0)+\Lambda(\mathbf{B}),\mathbf{B}^T\boldsymbol{\Sigma}\mathbf{B}}.$$

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

By substituting $\boldsymbol{\Sigma}$ by its expression, we get that the covariance matrix of the distribution is

$$\begin{aligned} \mathbf{B}^T\boldsymbol{\Sigma}\mathbf{B} &= \mathbf{U}^T\mathbf{D}^{1/2}\mathbf{L}^T\boldsymbol{\Sigma}\mathbf{L}\mathbf{D}^{1/2}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{D}^{1/2}\mathbf{L}^T\mathbf{L}^{-T}\sigma^2\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{L}\mathbf{D}^{1/2}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{D}^{1/2}\sigma^2\mathbf{D}^{-1}\mathbf{D}^{1/2}\mathbf{U} = \sigma^2\mathbf{I}. \end{aligned} \tag{1}$$

Since the distribution is also a discrete Gaussian, the signatures do not leak information on the secret basis.

Algorithm 1 $\mathsf{ffSampler}_{\mathcal{R}^d}(\mathbf{t}, \mathsf{T})$, sampler of FALCON

Require: Integer d a power of two, distribution $\mathcal{S}(\cdot,\cdot)$ over $\mathbb{Z}$, $\mathbf{t} \in (\mathcal{R}_d)^2$, a FALCON tree T.

if $d = 1$ **then**
  $\sigma \leftarrow \mathsf{T}$
  $t_0, t_1 \leftarrow \mathbf{t}$
  $z_0 \leftarrow \mathsf{SamplerZ}(\sigma, t_0)$
  $z_1 \leftarrow \mathsf{SamplerZ}(\sigma, t_1)$
  return (z_0, z_1)
end if
$(\ell, \mathsf{T}_0, \mathsf{T}_1) \leftarrow \mathsf{T}$
$(t_0, t_1) \leftarrow \mathbf{t}$
$\mathbf{t}_1 \leftarrow ((t_1(x) + t_1(-x))/2, (t_1(x) - t_1(-x))/2x)$
$\mathbf{z}_1 \leftarrow \mathsf{ffGNP}_{\mathcal{R}_{d/2},\mathcal{S}}(\mathbf{t}_1, \mathsf{T}_1)$
$z_1^0, z_1^1 \leftarrow \mathbf{z}_1$
$z_0 \leftarrow z_1^0(x) + xz_1^1(x)$
$t_0' \leftarrow t_0 + \ell(t_1 - z_1)$
$\mathbf{t}_0' \leftarrow ((t_0'(x) + t_0'(-x))/2, (t_0'(x) - t_0'(-x))/2x)$
$\mathbf{z}_0 \leftarrow \mathsf{ffGNP}_{\mathcal{R}_{d/2},\mathcal{S}}(\mathbf{t}_0', \mathsf{T}_0)$
$z_0^0, z_0^1 \leftarrow \mathbf{z}_0$
$z_0 \leftarrow z_0^0(x) + xz_0^1(x)$
return $\mathbf{z} = (z_0, z_1)$

Algorithm 2 $\mathsf{Sampler}_{\mathbb{Z}}(\sigma, t)$

Require: Integer $t \in \mathbb{Z}$, standard deviation σ
$r = t - \lfloor t \rceil$
while True **do**
  $z_0 \leftarrow \mathsf{BaseSampler}()$
  $b \leftarrow U(\{0, 1\})$
  $z \leftarrow b + (2 * b - 1)z_0$
  if $\mathsf{RejSamp}(z, z_0, \sigma, r)$ **then return** $z + \lfloor t \rceil$
  end if
end while

Algorithm 3 $\mathsf{BaseSampler}()$

$z \leftarrow 0$
$u \leftarrow U([2^{72}])$
for $i \in \{0, \ldots, 17\}$ **do**
  $z \leftarrow z + (u < \texttt{RCDT}[i])$
end for
return z

4 Generalized Nearest Plane Algorithms

In this section we define a framework for generalized Nearest-plane algorithm that will allow us to analyze the behaviour of the signature distribution during our attack.

4.1 Nearest Plane

The nearest plane algorithm was introduced by [2] as method to compute an approximated solution for CVP. More precisely, given a basis $\mathbf{B}$ of a full-rank lattice and any target vector in the ambient space, it allows computing a lattice point such that the difference between the point and the target lie in the parallelepiped spanned by the GSO vectors $\tilde{\mathbf{B}}$.

A randomized version was introduced by [34], allowing to decorrelate the distribution of the outputs from the basis $\mathbf{B}$ at the cost of performing slightly worse that the nearest plane algorithm in terms of CVP approximation. The KleinSampler is presented as Algorithm 4.

We recall a lemma from [23], demonstrating the relation between the output of KleinSampler and the target vector in the GSO basis.

Algorithm 4 $\mathsf{KleinSampler}(\mathbf{B}, \sigma, \mathbf{c})$

Require: Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{Z}^{n \times m}$, its GSO $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n\}$, a standard deviation σ, target $\mathbf{c} \in \mathbb{R}^m$
Ensure: $\mathbf{v}$ sampled in $D_{\Lambda(\mathbf{B}),\sigma,\mathbf{c}}$
$\mathbf{c}_n \leftarrow \mathbf{c}$
$\mathbf{v}_n \leftarrow 0$
for $i \leftarrow n, \ldots, 1$ **do**
  $c_i \leftarrow \frac{\langle \mathbf{c}_i, \tilde{\mathbf{b}}_i \rangle}{\|\tilde{\mathbf{b}}_i\|^2}$
  $\sigma_i \leftarrow \frac{\sigma}{\|\tilde{\mathbf{b}}_i\|}$
  $z_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_i,c_i}$
  $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$
  $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$
end for
return $\mathbf{v}_0$

Algorithm 5 $\mathsf{GNPSampler}_{\mathcal{S}}(\mathbf{B}, \sigma, \mathbf{c})$

Require: Distribution $\mathcal{S}(\cdot,\cdot)$ over $\mathbb{Z}$, basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{Z}^{n \times m}$, its GSO $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n\}$, a standard deviation σ, target $\mathbf{c} \in \mathbb{R}^m$

$\mathbf{c}_n \leftarrow \mathbf{c}$
$\mathbf{v}_n \leftarrow 0$
for $i \leftarrow n, \ldots, 1$ **do**
  $c_i \leftarrow \frac{\langle \mathbf{c}_i, \tilde{\mathbf{b}}_i \rangle}{\|\tilde{\mathbf{b}}_i\|^2}$
  $\sigma_i \leftarrow \frac{\sigma}{\|\tilde{\mathbf{b}}_i\|}$
  $z_i \leftarrow \mathcal{S}(\sigma_i, c_i)$
  $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$
  $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$
end for
return $\mathbf{v}_0$

Lemma 1. *For any input* $(\mathbf{B}, \sigma, \mathbf{c})$ *and any output* $\mathbf{v} = \sum_{i \in [n]} z_i \mathbf{b}_i \in \mathcal{L}(\mathbf{B})$ *of* KleinSampler*,*

$$\mathbf{v} - \mathbf{c} = \sum_{i \in [n]} (z_i - c_i) \cdot \tilde{\mathbf{b}}_i,$$

where the values c_i *are as in* KleinSampler*.*

We introduce as Algorithm 5 a slightly more general version of the KleinSampler, in which the Gaussian sampling step over the integers is replaced by an arbitrary distribution that we call the *base distribution*. This new GNPSampler allows to capture the behaviour of the KleinSampler with a faulty Gaussian sampler.

The following lemma shows that, under some mild assumptions on the base distribution, the distribution of the GNPSampler when called on a random target $\mathbf{c}$ heuristically keeps the same relation with the GSO basis as the KleinSampler. The heuristic is that the one-dimensional centers c_i that occur in the algorithm are uniformly random and independent modulo 1; this can be shown in the Gaussian case for random targets when the Gaussian parameters $\sigma_i / \|\tilde{\mathbf{b}}_i\|$ are large compared to the smoothing parameter of $\mathbb{Z}$ and heuristically extends to more general cases. Note that targets in signature schemes are not typically fully random (e.g., they are usually integer vectors), so the heuristic assumptions never strictly holds, but even then, the resulting rational numbers c_i behave close enough to random modulo 1 for the model to hold.

Lemma 2. *Suppose that the distribution* $\mathcal{S}$ *is translation invariant with respect to* $\mathbb{Z}$ *in its second parameter (i.e., for all* σ, c *and all* $m \in \mathbb{Z}$*,* $\mathcal{S}(\sigma, m + c) = m + \mathcal{S}(\sigma, c)$*). Assume furthermore that, in the notations of* GNPSampler*, when the algorithm in called on a random target* $\mathbf{c}$*, the intervening one-dimensional centers* c_i *behave as uniform and independent values modulo* 1*. Then:*

$$\mathbf{v} - \mathbf{c} \text{ and } \mathbf{x}\tilde{\mathbf{B}}$$

follow the same distribution, where $\mathbf{x}$ *is a random vector with independent coefficients, whose* i*-th coefficient is sampled according to the distribution* $\mathcal{S}(\sigma_i, u)$ *for* u *uniformly random in* $[0, 1)$ *(and* σ_i *is as in* GNPSampler*).*

In FALCON, the sampler differs from nearest plane to speed up computations by taking advantage of the ring $\mathcal{R}_n$ structure. The ffNP sampler was introduced as a randomized variant of the fast fourier sampler of [12]. The sampler recurses over the dimension of the ring, down to dimension 1 where $\mathcal{R}_1 = \mathbb{Z}$. Over the integers, the algorithm makes a call to a Gaussian sampler over the integers, to finally unwind the recursion calls.

Similarly to the KleinSampler case, we introduce as Algorithm 6 a generalized ffGNP that replaces the base distribution by an arbitrary distribution. The ffGNP sampler captures the behaviour of the sampler in any attack on FALCON that would target the base sampler consistently. Again, under some mild assumption on the base distribution, outputs of ffGNP can still be described relatively easily in the GSO basis.

Lemma 3. *Let* $\mathbf{B} = (\mathbf{b}_0, \mathbf{b}_1) \in \mathcal{R}^{2\times 2}$ *be a basis and* $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_0, \tilde{\mathbf{b}}_1)$ *its GSO in* $\mathcal{R}$*. Then vectors* $\mathbf{z} = \bar{\mathbf{z}}\mathbf{B}, \mathbf{t}$ *and* $\mathbf{t}' = (t'_0, t_1)$ *of Algorithm 6 are such that*

$$(\bar{\mathbf{z}} - \mathbf{t})\mathbf{B} = (\bar{\mathbf{z}} - \mathbf{t}')\tilde{\mathbf{B}}$$

This lemma is an application of [12, Lemma 3] in dimension 2 and with a power of two ring dimension. Next Lemma is an adapted version of [12, Theorem 2], which expresses a sample of the ffSampler sampler in the GSO of the bit-reversed basis.

Lemma 4. *Assume heuristically that the* t_i*'s in Algorithm 6 behave as independent and uniformly distributed values modulo* 1 *for each recursive call when calling the algorithm on a suitably random target* $\mathbf{t}$*, and suppose furthermore that the distribution* $\mathcal{S}$ *is translation invariant with respect to* $\mathbb{Z}$ *in its second parameter. Then, when sampling* $\mathbf{z} = \bar{\mathbf{z}}\mathbf{B} \leftarrow \mathsf{ffGNP}_{\mathcal{R}_n,\mathcal{S}}(\mathbf{t}, \mathsf{T})$ *for such a random target* $\mathbf{t}$*, the distribution satisfies:*

$$(\bar{\mathbf{z}} - \mathbf{t})\mathbf{B} \sim \mathbf{x}\tilde{\mathbf{B}}$$

where $\mathbf{x} \leftarrow (\mathcal{S}(d_i, u_i \bmod 1))_{i=0}^{2n-1}$*, where the* d_i *are the leaves of the tree* T*.*

Algorithm 6 $\mathsf{ffGNP}_{\mathcal{R}_n,\mathcal{S}}(\mathbf{t}, \mathsf{T})$

Require: Integer n a power of two, distribution $\mathcal{S}(\cdot,\cdot)$ over $\mathbb{Z}$, $\mathbf{t} \in (\mathcal{R}_n)^2$, a precomputed binary tree T of depth d, (implicitly) a matrix $\mathbf{B} \in \mathcal{R}^{2\times 2}$ such that T is the compact LDL^T decomposition tree of $\mathbf{B}\mathbf{B}^T$.

if $n = 2$ **then**
  $\sigma \leftarrow \mathsf{T}$
  $t_0, t_1 \leftarrow \mathbf{t}$
  $z_0 \leftarrow \mathsf{GNPSampler}(\mathcal{S}, \mathbf{B}, \sigma, t_0)$
  $z_1 \leftarrow \mathsf{GNPSampler}(\mathcal{S}, \mathbf{B}, \sigma, t_1)$
  return (z_0, z_1)
end if
$(\ell, \mathsf{T}_0, \mathsf{T}_1) \leftarrow \mathsf{T}$
$(t_0, t_1) \leftarrow \mathbf{t}$
$\mathbf{t}_1 \leftarrow ((t_1(x) + t_1(-x))/2, (t_1(x) - t_1(-x))/2x)$
$\mathbf{z}_1 \leftarrow \mathsf{ffGNP}_{\mathcal{R}_{n/2},\mathcal{S}}(\mathbf{t}_1, \mathsf{T}_1)$
$z_1^0, z_1^1 \leftarrow \mathbf{z}_1$
$z_1 \leftarrow z_1^0(x) + xz_1^1(x)$
$t'_0 \leftarrow t_0 + \ell(t_1 - z_1)$
$\mathbf{t}'_0 \leftarrow ((t'_0(x) + t'_0(-x))/2, (t'_0(x) - t'_0(-x))/2x)$
$\mathbf{z}_0 \leftarrow \mathsf{ffGNP}_{\mathcal{R}_{n/2},\mathcal{S}}(\mathbf{t}'_0, \mathsf{T}_0)$
$z_0^0, z_0^1 \leftarrow \mathbf{z}_0$
$z_0 \leftarrow z_0^0(x) + xz_0^1(x)$
return $\mathbf{z} = (z_0, z_1)$

4.2 Learning a parallelepiped

We work in the extended framework of the Nguyen-Regev attack [44] introduced in [36], recalling the definition of the *Hidden Transformation Problem* (HTP).

Definition 1 (HTP_D)**.** *Let* D *be a public distribution over* $\mathbb{R}^n$*. Given a hidden matrix* $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathrm{GL}_n(\mathbb{R})$ *and a certain number of independent samples* $\mathbf{y} = \mathbf{x}\mathbf{B}$ *with* $\mathbf{x} \leftarrow D$*, find an approximation of* $\pm\mathbf{b}_i$*'s.*

In our case, the distribution D is the output of a faulty ffSampler which is explicited in Lemma 4. We write $D(\mathbf{M})$ for the distribution of $\mathbf{x}\mathbf{M}$ with $\mathbf{x} \sim D$.

Lemma 5. *Let* $\mathbf{B} \in \mathrm{GL}_n(\mathbb{R})$ *and* $\mathbf{K} = \mathrm{Cov}[D(\mathbf{B})]$. *Let* $\mathbf{L} \in \mathrm{GL}_n(\mathbb{R})$ *such that* $\mathbf{L}^T\mathbf{L} = \mathbf{K}^{-1}$. *Then the distribution of* $\mathbf{y}\mathbf{L}^T$ *with* $\mathbf{y} \sim D(\mathbf{B})$ *is* $D(\mathbf{C})$ *with* $\mathbf{C} = \mathbf{B}\mathbf{L}^T$ *such that* $\mathrm{Cov}[D(\mathbf{C})] = \mathbf{I}_n$. *In particular,* $\mathbf{C}$ *is orthogonal when* $\mathrm{Cov}[D] = \mathbf{I}_n$.

Since we do not have access to the covariance matrix $\mathbf{K} = \mathrm{Cov}[D(\mathbf{B})]$, we rely on samples of the distribution to compute an approximation. Given a list of samples $S = (\mathbf{s}_i)$, the Bessel-corrected sample covariance matrix (assuming the distribution is centered)

$$\frac{1}{\#S - 1} \sum_i \mathbf{s}_i^T \mathbf{s}_i$$

can be used to get an estimate of $\mathrm{Cov}[D(\mathbf{B})]$.

We follow the lines of [36] for the recovery of the secret basis $\mathbf{B}$. Let us consider that the distribution D is a joint distribution of centered distributions D_i. Moreover, if $\sigma_i^2 = \mathrm{Var}(D_i)$ and $\mathbf{D} = \mathrm{diag}(\sigma_i^2)$, with the notations of Lemma 5, if we let $\mathbf{C}' = \mathbf{D}\mathbf{C}$ and D' be the distribution of $\mathbf{x}\mathbf{D}^{-\mathbf{1}}$ for $x \sim D$, then $D'(\mathbf{C}') = D(\mathbf{C})$ and $\mathbf{C}'$ is orthogonal. In the following, we consider the case where $\mathbf{C}$ is orthogonal and $\mathrm{Cov}[D] = \mathbf{I}_n$.

The 4-th moment over $\mathbf{w} \in \mathbb{R}^n$ of distribution $D(\mathbf{C})$ is defined as

$$M_{D(\mathbf{C})}(\mathbf{w}) = \mathbb{E}_{\mathbf{s} \sim D(\mathbf{C})}[\langle \mathbf{w}, \mathbf{s} \rangle^4].$$

If we write $\mathbf{s} = \sum z_i \mathbf{c}_i$, for $z_i \sim D_i$, and $\alpha_i = \mathbb{E}[z_i^4]$, the 4-th moment can be rewritten as

$$\begin{aligned}
M_{D(\mathbf{C})}(\mathbf{w}) &= \mathbb{E}[\langle \mathbf{s}, \mathbf{w} \rangle^4] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} z_i \langle \mathbf{c}_i, \mathbf{w} \rangle\right)^4\right] \\
&= \sum_{i=1}^{n} \mathbb{E}[z_i^4] \langle \mathbf{c}_i, \mathbf{w} \rangle^4 + 3 \sum_{i \neq j} \langle \mathbf{c}_i, \mathbf{w} \rangle^2 \langle \mathbf{c}_j, \mathbf{w} \rangle^2 \\
&= 3\|\mathbf{w}\|^4 - \sum_{i=1}^{n} (3 - \alpha_i) \langle \mathbf{c}_i, \mathbf{w} \rangle^4.
\end{aligned}$$

The third equation is obtained by expanding the 4-th power and using the fact that the z_i are centered ($\mathbb{E}[z_i] = 0$) and $\mathrm{Cov}[D] = \mathbf{I}_n$ ($\mathbb{E}[z_i^2] = 1$ and $\mathbb{E}[z_i z_j] = 0$ for $j \neq i$). The last equation is verified as $\mathbf{C}$ is orthogonal, the norm $\|w\|$ can be rewritten as $\sum_i \langle \mathbf{c}_i, \mathbf{w} \rangle^2$.

The following lemma from [36] states that vectors $\pm\mathbf{c}_i$ are the only local minimas of $M_{D(\mathbf{C})}$ over $\mathbb{S}^{n-1}$.

Lemma 6. *Suppose that* $\mathbb{E}_{z_i \sim D_i}[z_i^4] = \alpha_i < 3$ *for all* $1 \leq i \leq n$, *the local minimum of* $M_{D(\mathbf{C}),4}(\mathbf{w})$ *over all* $\mathbf{w} \in \mathbb{S}^{n-1}$ *are obtained at* $\pm\mathbf{c}_1, \ldots, \pm\mathbf{c}_n$. *There are no other local minima.*

Informally, the conditions of Lemma 6 state that for any centered joint distribution (normalized) such that the fourth moment is less that of a spherical gaussian $\mathcal{N}(0, \mathbf{I}_n)$, information about the secret basis is leaked and the fourth moment function can be used to recover $\mathbf{C}$.

This provides us with a clear way to recover the secret basis $\mathbf{B}$, assuming that the faulty distribution is more concentrated around zero than a Gaussian (i.e., it has negative excess kurtosis), a condition which is clearly satisfied in our setting. We will see, however, that a simple application of the techniques from [36] is not sufficient for our purposes.

5 Key Recovery Attack

In this section, we explain how to attack FALCON's sampler to generate faulty signatures and analyze their distribution, in order to apply the framework developed in the previous section. In that process, we show a new attack, that leverages eigenvectors of the faulty sampler's covariance matrix is possible and in fact much more efficient.

We go through two main steps, the first being the fault injection performed with the Rowhammer technique, and the second performs the key recovery using the faulty signatures.

5.1 First step: Rowhammer

Fault attack target. The purpose of FALCON's sampler is to decorrelate the signatures from the basis used to sign by making the overall signature distribution close to a discrete Gaussian with standard deviation σ, where σ depends on the selected parameter set. In order to achieve this, the sampler relies on the ability to sample discrete Gaussians over $\mathbb{Z}$ for a fixed range $[\sigma_{\min}, \sigma_{\max}]$ of standard deviations.

The way this integer Gaussian sampler SamplerZ (Algorithm 2) is implemented by first relying on a sampler BaseSampler (Algorithm 3) to sample from a "half-Gaussian" distribution $D_{\mathbb{Z}^+,\sigma_{\max}}$ which is then made symmetric and is corrected by rejection sampling to correct the center.

The real point of interest is thus the "half-Gaussian" sampler BaseSampler. It is implemented to run in constant-time, by performing a linear scan of the reverse cumulative distribution table `RCDT` of the distribution. This table consists of 18 entries that are 72-bit integer. For $i \in [\![18]\!]$, $\texttt{RCDT}[i] = 2^{72} - \sum_{j \leq i} \lfloor 2^{72} \cdot D_{\mathbb{Z}^+,\sigma_{\max}}(j) \rceil$. That table is stored in memory and is scanned every time the sampler is called. As described in algorithm 3, the BaseSampler starts by sampling a uniform 72-bit integer u and increments a counter for each i such that $u < \texttt{RCDT}[i]$. The value of the counter is the sample output by BaseSampler. A direct observation is that the lower the values of `RCDT` are, the lower the standard deviation of the distribution becomes.

As the `RCDT` table is stored in memory during the signing process, it is a potential target to the Rowhammer attack. We perform the attack on the `RCDT` by targeting the most significant bits of the first entries. In particular, we show that a *single* bit flip (zeroing out the most significant bit of the first entry of the `RCDT`) is enough to successfully carry out the remainder of the attack achieve full key recovery, with more bit flips enabling key recovery with fewer collected signatures. The objective of this part is to maximally reduce the standard deviation of the outputs of BaseSampler. As per the previous observation, this implies lowering the entries of `RCDT` as much as possible. The largest values are obviously the main targets, given that they trigger a counter increment more often.

Rowhammer. The Rowhammer attack leverages some electromagnetic effect such that repeatedly accessing a specific row in DRAM can cause bit flips in adjacent rows. The effect occurs because parasitic currents generated during the activation cause slight discharge in the capacitors storing bit values in neighboring rows. If this discharge happens in the interval of the refreshment of the DRAM, the flipped bit is stored in the RAM instead of the original. The attack can actually trigger bit flips in both directions, but since we only want to decrease the values in the `RCDT`, we care only about bit flips from 1 to 0.

An important property of Rowhammer is the repeatability of bit flips. Once a bit flip occurs in memory, it is likely to occur again in the future. This allows the adversary to scan memory in order to flag vulnerable areas for a future attack. Bit flips also appear to occur in a single direction at a particular location, so in our case we can figure out which bits in the memory are vulnerable to the 1 to 0 bit flip.

Once vulnerable bits have been flagged and an ideal chunk of memory has been identified for the attack, the sampler should be run such that the `RCDT` is stored in the vulnerable chunk. One can rely on a technique called *memory massaging* [35], which exploits the Linux page allocation system, to ensure that a new program runs in a specific page of memory.

In order to experimentally validate the feasibility of this step, we use the `hammertime` software suite [60] to check for vulnerable memory locations on a victim machine, equipped with the Ivy Bridge Intel Core i7-3770 CPU with 4×4GB DDR3 DRAM modules (part number AM2U16BC4P2-B01S). An 11-hour search with single-sided hammering on 1GB of memory revealed 81 vulnerable locations, including 4 or 6 with our exact desired bit flip[3]. Double-sided hammering also worked, but did not appear to be more effective. The search also showed that bit flips to 0 of the MSBs of the first *two* entries of the `RCDT` were also simultaneously achievable on our victim. A longer search is probably necessary to achieve more flips, like the 8 flips obtained in the attack against FrodoKEM [19].

[3] The `RCDT` table representation depends on whether AVX2 optimizations are enabled in FALCON. With the AVX2 representation, we want a suitably aligned `0x40` to `0x00` bit flip, and otherwise, `0x80` to `0x00`. In principle, only the latter matters for the victim machine, since it does not support AVX2 instructions, but it is useful to note that both are achievable.

We note that, contrary to [1, 19], there are no significant timing constraints in our setting for achieving the bit flips. If we assume that the process carrying out the FALCON signatures is continuously running, we can flip bits on its RCDT at any time; we can easily verify that our desired bit flip(s) have been obtained by simply ccazdeJ-jizbaj-4wycnohecking the length of generated signatures, which become much smaller (and repeat the attack until this is achieved). In contrast, the attacks in [1, 19], which target key generation algorithms, need to complete in the short time span before those algorithms return.

5.2 Key recovery by principal component analysis

Distribution of faulty signatures. Once the bitflips have been triggered, we can start collecting biased signatures. Given that ffSampler no longer returns vectors distributed as Gaussians and that the effective standard deviation of Gaussian sampling over $\mathbb{Z}$ has been lowered, we expect a potential leakage of the secret basis geometry. Due to the nature of the nearest plane algorithm, the leaked geometry should be linked to the GSO of the secret basis. To rigorously investigate this leakage, we carry out a statistical study of the correlation between the GSO basis and the biased signatures, using the ffGNP framework of Section 4. First we need to figure out the impact that the biased half-Gaussian sampler has on ffSampler. Let us call the biased sampler after k bitflips BS_k.

As discussed in Section 3.1, the original sampler outputs a sample following the distribution $\mathcal{D}_{\boldsymbol{\Sigma}}$, where $\boldsymbol{\Sigma} = \mathbf{L}^{-T}\mathbf{D}\mathbf{L}^{-1}$. Since the attack targets SamplerZ, only the base distribution is affected. Hence, the final distribution is that of $\mathsf{ffGNP}_{\mathcal{R}_d,\mathsf{BS}_k}$. As per Lemma 4, if we let d_i the values at the leaves of the FALCON tree, the signatures are now distributed as

$$(\mathsf{BS}_k(d_1, u_1), \dots, \mathsf{BS}_k(d_n, u_n))\tilde{\mathbf{B}}$$

where the u_i are uniformly distributed over $[0, 1]$.

Since only the half-gaussian part is affected by the bit flips, the distributions BS_k keep the symmetry of the original Gaussian sampler. More precisely, the SamplerZ algorithm ensures that the distribution of $\mathsf{BS}_k(\sigma, 1-c)$ is equal to that of $1 - \mathsf{BS}_k(\sigma, c)$ for any choice of (σ, c). Therefore, for u_i uniform in $[0, 1]$, we clearly have:

$$\mathbb{E}_{x\sim\mathsf{BS}_k(d_i,u_i)}[x] = 0.$$

Now let $\tilde{\mathbf{D}} = \mathrm{diag}(\tilde{d_1^2}, \dots, \tilde{d_n^2})$, where $\tilde{d_i^2}$ is the variance $\mathrm{Var}_{x\sim\mathsf{BS}_k(d_i,u_i)}[x]$. By our previous results, the model predicts that the faulty sampler outputs samples with covariance matrix $\bar{\boldsymbol{\Sigma}} = \mathbf{L}^{-T}\tilde{\mathbf{D}}^{-1}\mathbf{L}^{-1}$. This model is confirmed by our experiments, as the eigenvalues of the covariance matrix perfectly match those plotted in Figures 4 and 2.

With the modified distribution, the identity presented in equation 1 becomes

$$\begin{aligned}\tilde{\boldsymbol{\Sigma}} = \mathbf{B}^T\bar{\boldsymbol{\Sigma}}\mathbf{B} &= \mathbf{U}^T\mathbf{D}^{1/2}\mathbf{L}^T\bar{\boldsymbol{\Sigma}}\mathbf{L}\mathbf{D}^{1/2}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{D}^{1/2}\mathbf{L}^T\mathbf{L}^{-T}\tilde{\mathbf{D}}^{-1}\mathbf{L}^{-1}\mathbf{L}\mathbf{D}^{1/2}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{D}\tilde{\mathbf{D}}^{-1}\mathbf{U}\end{aligned}$$

which indicates a potential leakage of information, depending of the ratio $\mathbf{D}\tilde{\mathbf{D}}^{-1}$.

Eigenvectors and GSO basis. It follows from the previous equation that the diagonal elements $d_i^2/\tilde{d_i^2}$ of $\mathbf{D}\tilde{\mathbf{D}}^{-1}$ are eigenvalues of the covariance matrix $\tilde{\boldsymbol{\Sigma}}$, and the vectors of $\mathbf{U}$ are corresponding eigenvectors.

This observation leads to an alternative way to recover the GSO basis. If all values of $\mathbf{D}\tilde{\mathbf{D}}^{-1}$ differ (i.e., every eigenvalue is of multiplicity 1), a good approximation of $\tilde{\boldsymbol{\Sigma}}$ should allow to recover vectors of $\mathbf{U}$, revealing the secret basis $\mathbf{B}$.

Given a list of signatures $S = (\mathbf{s}_1, \mathbf{s}_2, \dots)$, we form the sample covariance matrix

$$\frac{1}{\#S}\sum_i \mathbf{s}_i^T\mathbf{s}_i$$

to get an estimate of the covariance matrix of the signatures $\tilde{\mathbf{\Sigma}} = \mathbf{U}^T\mathbf{D}\tilde{\mathbf{D}}^{-1}\mathbf{U}$ and hope that the corresponding eigenvectors reveal $\mathbf{U}$, which, up to scaling, is essentially the GSO: this is the principal component analysis (PCA) of the signature distribution.

As we will see, the assumption that the eigenvalues are of multiplicity 1 is actually never satisfied, but we will find ways of circumventing this difficulty nonetheless.

Next, we investigate the distribution of eigenvalues depending on the number of bit flips triggered in the RCDT by the Rowhammer attack. To fix ideas, we focus on three cases (all zeros, 8 bit flips and 1 bit flip), but one could easily extend the analysis to all other cases.

Expression of the eigenvalues. For any number of bit flips, we can give an expression (not in closed form) of the variance $\tilde{d}$ of $\mathsf{BS}_k(\sigma, u)$, $u \sim U([0,1])$, depending on d. A convenient way to do so is to introduce the function φ_k which gives, for any nonnegative integer x, the ratio between its probability to appear in the faulty base sampler to its probability in the correct distribution BaseSampler. If we denote by $\widetilde{\texttt{RCDT}}_k$ the faulty RCDT after k bit flips *sorted in decreasing order*, we have:

$$\varphi_k(i) = \frac{\widetilde{\texttt{RCDT}}_k[i-1] - \widetilde{\texttt{RCDT}}_k[i]}{\texttt{RCDT}[i-1] - \texttt{RCDT}[i]}$$

where to simplify notations we set $\widetilde{\texttt{RCDT}}_k[-1] = \texttt{RCDT}[-1] = 2^{72}$. In particular, if i_k is the largest index modified by bit flips *after sorting the table* $\widetilde{\texttt{RCDT}}_k$, we have $\varphi_k(i) = 1$ for $i > i_k$.

Then, for any $c \in [0,1)$, the probability that $\mathsf{BS}_k(\sigma, c)$ outputs $z \in \mathbb{Z}$ satisfies:

$$\Pr[z \leftarrow \mathsf{BS}_k(\sigma, c)] \propto \exp\big(-(z-c)^2/2\sigma^2\big) \cdot \psi_k(z_0)$$

where we define $\psi_k(z) = \varphi_k(z_0)$, where $z_0 = z - 1$ for $z \geq 1$ and $z_0 = -z$ otherwise. One can then deduce the probability that $\mathsf{BS}_k(\sigma, u)$ outputs $z \in \mathbb{Z}$ for $u \sim U([0,1])$:

$$\Pr[z \leftarrow \mathsf{BS}_k(\sigma, u)] = \int_0^1 \frac{\exp\big(-(z-t)^2/2\sigma^2\big) \cdot \psi_k(z)}{\sum_{z' \in \mathbb{Z}} \exp\big(-(z'-t)^2/2\sigma^2\big) \cdot \psi_k(z')}\, dt,$$

and the variance follows:

$$\tilde{d}^2 = \mathrm{Var}[\mathsf{BS}_k(\sigma, u)] = \sum_{z \in \mathbb{Z}} z^2 \int_0^1 \frac{\exp\big(-(z-t)^2/2\sigma^2\big) \cdot \psi_k(z)}{\sum_{z' \in \mathbb{Z}} \exp\big(-(z'-t)^2/2\sigma^2\big) \cdot \psi_k(z')}\, dt.$$

This expression is somewhat cumbersome, but easy to evaluate in any computer algebra system (we use PARI/GP and its convenient `intnum` and `suminf` functions).

The i-th eigenvalue $\tilde{d}_i^2/d_i^2$ of the correlation matrix then becomes:

$$\frac{1}{d_i}\,\mathrm{Var}[\mathsf{BS}_k(d_i, u)] \quad \text{where} \quad d_i^2 = \sigma^2/\|\tilde{\mathbf{b}}_i\|,$$

and the variance follows the expression above. Therefore, if we let:

$$f_k(s) = \frac{s^2}{\sigma^2} \sum_{z \in \mathbb{Z}} z^2 \int_0^1 \frac{\exp\big(-s^2(z-t)^2/2\sigma^2\big) \cdot \psi_k(z)}{\sum_{z' \in \mathbb{Z}} \exp\big(-s^2(z'-t)^2/2\sigma^2\big) \cdot \psi_k(z')}\, dt,$$

the eigenvalues of the correlation matrix are simply computed for the GSO norms as $\tilde{d}_i^2/d_i^2 = f_k\big(\sqrt{\|\tilde{\mathbf{b}}_i\|}\big)$.

The case of a fully zeroed out RCDT. The first case we study is the extreme case in which we turn all the entries of RCDT to 0. In practice, this amounts to turning BaseSampler into the 0 function.

One might expect that this would turn ffSampler into the nearest plane algorithm. However, some specificity of FALCON's sampler that make our case slightly more involved. Even with BaseSampler always returning 0, the

full integer Gaussian sampler with inputs (t, σ) does not only output $\lfloor t \rceil$, but might also output $\lfloor t \rceil + 1$. With the notation above, this is equivalent to setting $\varphi(0) = 1$ and $\varphi(z) = 0$ for $z > 0$, hence $\psi(0) = \psi(1) = 1$ and $\psi(z) = 0$ for $z \neq 0, 1$.

As expected, this makes FALCON significantly weaker. In fact in this setting, the original parallelepiped-learning attack of [44] suffices with a few million signatures. This scenario seems unlikely to be realistic since it puts all the burden of the attack on the Rowhammer side, expecting that the whole RCDT table gets flipped to zero.

Eight bit flips. This time we consider a less extreme case, where only 8 bitflips in the RCDT are expected. This scenario corresponds to the setup of the attack of [19] on FrodoKEM, where they target 8 specific bits of the secret key during key generation. We take on a min-max approach, that is, we consider the 8 bitflips minimizing the value $\max_i \texttt{RCDT}[i]$.

In this case, the f_8 function, represented in Figure 1, is increasing, meaning that larger GSO vectors are associated with larger eigenvalues. In particular, one of $\tilde{\mathbf{b}}_0$ or $\tilde{\mathbf{b}}_{512}$ must be an eigenvector for the top eigenvalue.

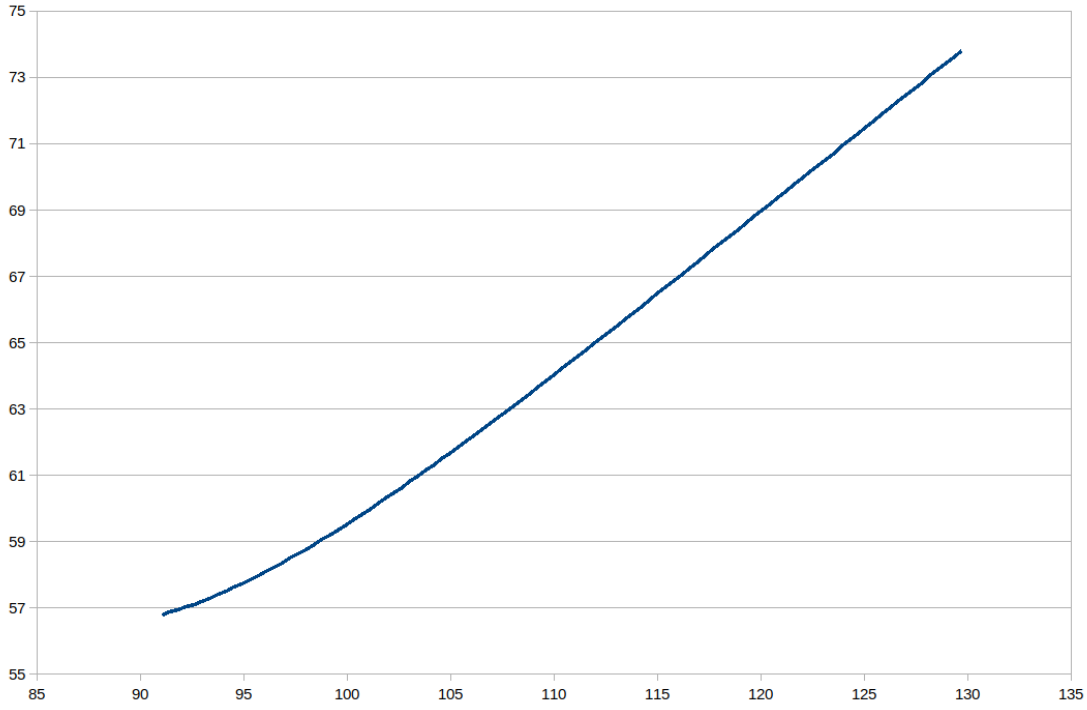

Fig. 1. Eigenvalues in terms of GSO norms, case of 8 bit flips.

We generate a fresh keypair and plot the eigenvalues associated to the covariance matrix of the signatures in Figure 2. The profile observed in Figure 2 is consistent over all keys that we tested. We notice that the eigenvalues for the first vector of the normalized GSO $\mathbf{u}_0$ and for the vector $\mathbf{u}_{511}$ are relatively high compared to most of the others.

Single bit flip. We here consider an even simpler scenario where only 1 bit flip is performed. As for 8 bitflips, we select the bit flip that minimize the value $\max_i \texttt{RCDT}[i]$, which corresponds here to flipping the most significant bit of $\texttt{RCDT}[0]$ to 0.

The function f_1 is now decreasing:

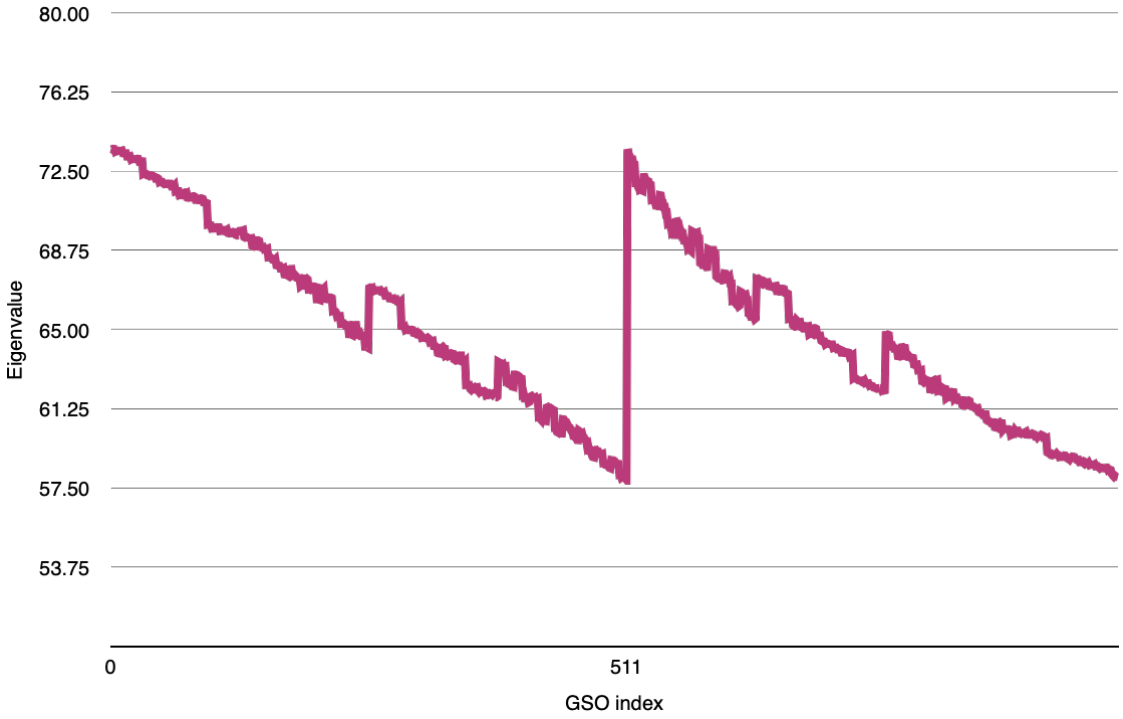

Fig. 2. Eigenvalues of the covariance matrix for 8 bitflips in the RCDT. The discontinuity occurs between indices 511 and 512. Eigenvalue of the 511-th vector is lower than the one of the 512-th.

We generate a fresh keypair and plot the eigenvalues associated to the covariance matrix of the signatures in Figure 4. The profile observed in Figure 4 is consistent over all keys that we tested. We notice that the eigenvalues for the vectors $\mathbf{u}_0$ and $\mathbf{u}_{512}$, contrary to the 8 bitflip case, are relatively low compared to most of the others. However, the useful observation for our attack is actually that this time, eigenvalues for vectors $\mathbf{u}_{511}$ and $\mathbf{u}_{1023}$ are relatively higher than others and the slope around them is higher. This implies that the gap between those eigenvalues and other is wider, which improves the probability to recover their corresponding eigenvectors, as discussed later on.

5.3 Full recovery

To carry on the attack, we first sample signatures coming from the biased sampler. We underline that we do not need to store the signatures since they are only used to compute an estimate of the covariance matrix $\tilde{\boldsymbol{\Sigma}}$ of the signatures. Hence, we can efficiently collect a high number of signatures and compute the approximation of $\tilde{\boldsymbol{\Sigma}}$.

While it is enough to find $\mathbf{u}_0$ to perform the attack, the symplectic and algebraic structure of NTRU lattices offer us more leeway to get to recover $\mathbf{b}_0$. The following theorem from [37, Theorem 1] gives us the tools we need.

Theorem 1. *The following properties hold.*

1. *Let $\omega : \mathbb{Z}^{2n} \to \mathbb{Z}^{2n}$ be the isometry given by:*

$$\omega(u_0, u_1, \ldots, u_{2k}, u_{2k+1}, \ldots, u_{2n-2}, u_{2n-1}) \\ = (-u_1, u_0, \ldots, -u_{2k+1}, u_{2k}, \ldots, -u_{2n-1}, u_{2n-2})$$

(i.e., ω negates the second element in each pair of consecutive coefficients and swaps the pair). In the bit reversed order representation of the module lattice, ω corresponds to multiplication by $x^{n/2} = \sqrt{-1}$ on both ring elements, so that, e.g., $\mathbf{b}_0 = (g, -f)$ is sent to the vector $\omega(\mathbf{b}_0) = (x^{n/2}g, -x^{n/2}f) = \mathbf{b}_1$.

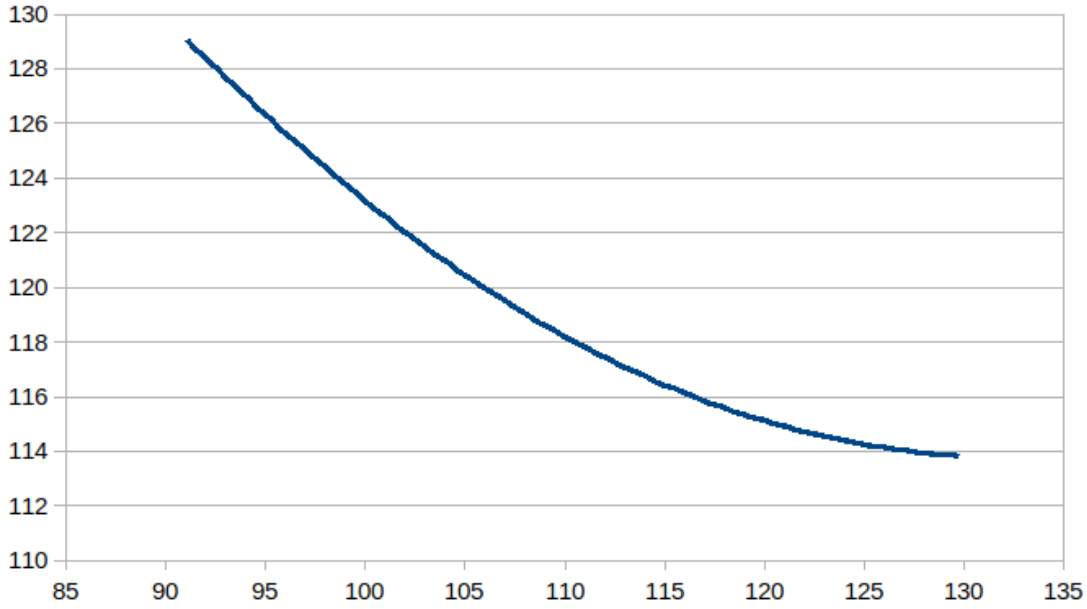

Fig. 3. Eigenvalues in terms of GSO norms, case of 1 bit flip.

Then, for $0 \leq i \leq n-1$*, we have:*

$$\omega(\mathbf{b}_{2i}) = \mathbf{b}_{2i+1} \quad \textit{and} \quad \omega(\mathbf{b}_{2i+1}) = -\mathbf{b}_{2i}.$$

Moreover, the same relation holds for the Gram-Schmidt vectors:

$$\omega(\mathbf{b}^*_{2i}) = \mathbf{b}^*_{2i+1} \quad \textit{and} \quad \omega(\mathbf{b}^*_{2i+1}) = -\mathbf{b}^*_{2i}.$$

In particular, $\|\mathbf{b}^*_i\| = \|\mathbf{b}^*_{2i+1}\|$.

2. *For all* i, $\|\mathbf{b}^*_i\| \cdot \|\mathbf{b}^*_{2n-1-i}\| = q$*. Moreover, we have:*

$$\frac{1}{\|\mathbf{b}^*_{2n-1-i}\|}\mathbf{b}^*_{2n-1-i} = \frac{1}{\|\mathbf{b}^*_i\|}\mathbf{b}^*_i\mathbf{J}$$

where $\mathbf{J}$ *is the standard symplectic involution.*

The isometry ω can be used to recover $\mathbf{u}_0$ from vector $\mathbf{u}_1$, and the symplectic identity given in Theorem 1 allows to recover $\mathbf{u}_0$ from $\mathbf{u}_{1023}$. Similarly, isometry ω allows to recover $\mathbf{u}_{1023}$ from $\mathbf{u}_{1022}$.

Corollary 1. *Let* $(\mathbf{b}_0, \ldots, \mathbf{b}_{1023})$ *be an NTRU basis in dimension 1024 and let* $(\tilde{\mathbf{b}}_0, \ldots, \tilde{\mathbf{b}}_{1023})$ *be its GSO. If we let* $\mathbf{u}_i = \tilde{\mathbf{b}}_0/\|\tilde{\mathbf{b}}_0\|$*, then assuming* $\|\mathbf{b}_0\|$ *is known, the vector* $\mathbf{b}_0$ *can be computed from any* $\mathbf{u}_i$ *for* $i \in \mathcal{T} = \{0, 1, 510, 511, 512, 513, 1022, 1023\}$.

Proof. From Theorem 1, it is clear that $\mathbf{u}_0$ can be computed from any $\mathbf{u}_i$ for $i \in \{0, 1, 1022, 1023\}$ and that $\mathbf{u}_{512}$ can be computed from any $\mathbf{u}_i$ for $i \in \{510, 511, 512, 513\}$. Since $\mathbf{b}_0 = \|\mathbf{b}_0\|\mathbf{u}_0$, it remains to show that $\mathbf{b}_0$ can be computed from $\mathbf{u}_{512}$.

As per [10, Lemma 3], we have the equality

$$\mathbf{u}_{512} = \frac{1}{K}(w, v) = \frac{1}{K}\left(q\frac{f^\star}{ff^\star + gg^\star}, q\frac{g^\star}{ff^\star + gg^\star}\right),$$

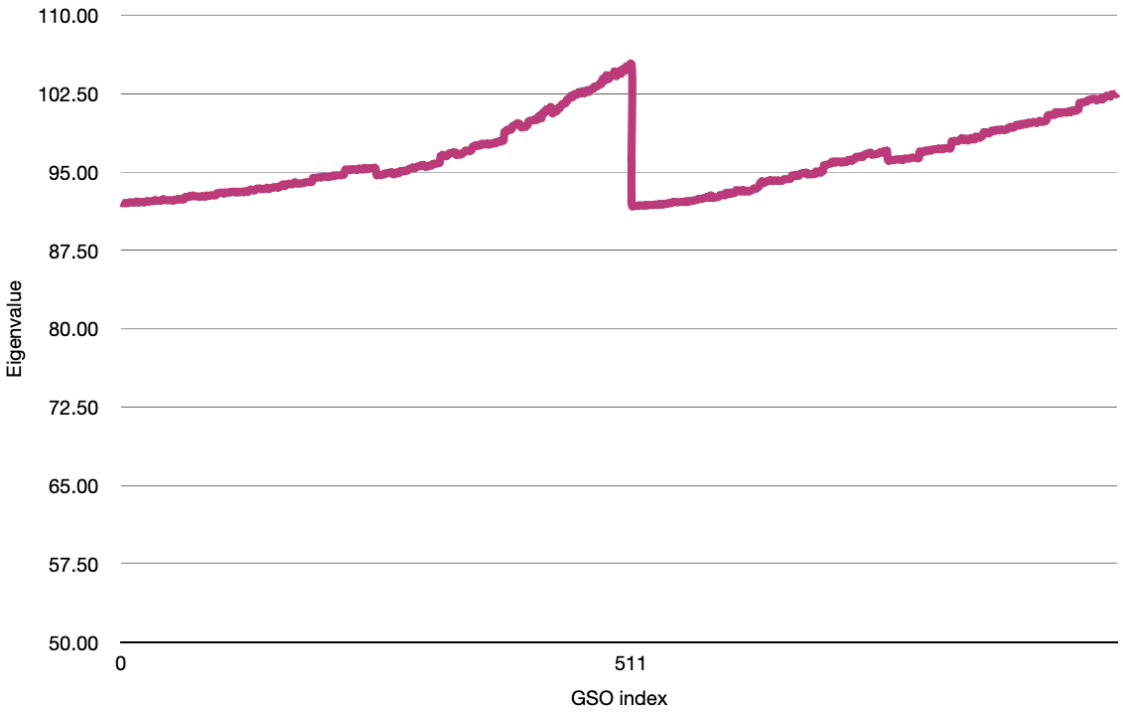

Fig. 4. Eigenvalues of the covariance matrix for 1 bitflip in the RCDT. The discontinuity occurs between indices 511 and 512. Eigenvalue of the 511-th vector is higher than the one of the 512-th.

with $K = \|\tilde{\mathbf{b}}_{512}\|$. Both polynomials f and g can be recovered through their FFT representation. Indeed, let ζ be a root of ϕ. Then by taking

$$\begin{aligned} c &= q(w(\zeta)w^\star(\zeta) + v(\zeta)v^\star(\zeta)) \\ &= \frac{q^2}{K}\frac{|f(\zeta)|^2 + |g(\zeta)|^2}{(|f(\zeta)|^2 + |g(\zeta)|^2)^2} \\ &= \frac{q^2}{K}\frac{1}{|f(\zeta)|^2 + |g(\zeta)|^2} \end{aligned}$$

we get both $qw^\star(\zeta)/c = f(\zeta)$ and $qv^\star(\zeta)/c = g(\zeta)$, which gives us the FFT representation of f and g, thus recovering $\mathbf{b}_0 = (g, -f)$.

Corollary 1 increases our success chances by allowing us to target 8 vectors instead of 1. In fact, in the 1 bitflip case, it proves crucial to make the attack significantly more efficient. Our success rate depends critically on our ability to identify the right eigenspaces to target, which in turns depends on both the eigenvalues spacing and the quality of the convergence.

Convergence of eigenvalues Let us make the assumption that the convergence error is Gaussian, such that $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} + \mathbf{E}$, with $\mathbf{E}_{i,j} \sim \mathcal{N}(0, s)$ where $s^2 = \mathcal{O}(1/N)$ and N is the number of samples.

By Weyl's inequality, the convergence of the eigenvalues is bounded by $\|\lambda_i(\tilde{\boldsymbol{\Sigma}}) - \lambda_i(\boldsymbol{\Sigma})\|_2 \leq \|\mathbf{E}\|_2$, for any $1 \leq i \leq N$. Since there exist a constant C independent of the dimension n and the number of samples N such that with high probability $\|\mathbf{E}\|_2 \leq C\sqrt{n}/\sqrt{N}$, the convergence inequality can be rewritten as

$$\|\lambda_i(\tilde{\boldsymbol{\Sigma}}) - \lambda_i(\boldsymbol{\Sigma})\|_2 \leq C\frac{\sqrt{n}}{\sqrt{N}}. \tag{2}$$

If for $1 \leq i \leq n$, we let $\lambda_i = \lambda_i(\boldsymbol{\Sigma})$ and $\tilde{\lambda}_i = \lambda_i(\tilde{\boldsymbol{\Sigma}})$, from Equation 2 we get that $\|\tilde{\lambda}_i - \tilde{\sigma}_j\| \geq \|\lambda_i - \lambda_j\| - 2C\sqrt{n}/\sqrt{N}$. Then, for $i \neq j$, at least $\mathcal{O}(1/\|\lambda_i - \lambda_j\|^2)$ samples are required to tell these eigenvalues appart. Consequently, the best strategy would appear to be to aim for $\mathbf{u}_{i^*}$ for $i^* \in \mathcal{T}$ which maximizes $\min_{j \neq i^*} |\lambda_{i^*} - \lambda_j|$.

However, as we'll see later, the behaviour of eigenvalues does not trivially dictates that of the eigenvectors.

Finishing the attack Once a good enough approximation of $\tilde{\boldsymbol{\Sigma}}$ is computed, we expect to recover an approximation of $\mathbf{U}$. However, this turns out to be more complex. Since $\mathbf{b}_{2i+1} = \omega(\mathbf{b}_{2i})$, vectors $\mathbf{b}_{2i}$ and $\mathbf{b}_{2i+1}$ have the same norm and since, in the FALCON tree, $d_i = \sigma/\sqrt{\|\mathbf{b}_i\|}$, we have that $d_{2i+1} = d_{2i}$. This implies that the eigenspaces of $\tilde{\boldsymbol{\Sigma}}$ are of dimension 2 and some method must be used to recover the target $\mathbf{u}_i$ in that space.

Since target vectors allow to recover $\mathbf{b}_0$, this eigensubspace can be mapped to a subspace of the same dimension, but containing $\mathbf{b}_0$, which is an integer vector whose norm is known. With a dimension 2 subspace, exhaustive search can be used to recover $\mathbf{b}_0$. However in practice, getting a good enough approximation such that the target vector is contained in a dimension 2 subspaces appears to require a very high number of signatures, even in the 8 bitflip scenario.

In order to make the attack more efficient, we adopt a hybrid approach, using the estimated eigenvectors to identify a k-dimensional space V, for $k \geq 2$, which contains a good approximation of the target vector, and recover the vector with the Nguyen-Regev attack, computing the minimas of the approximated 4-th moment $\tilde{M}_{(\mathbf{s}_i)} : w \mapsto \sum_{i=1}^{N} \langle \pi_V(\mathbf{s}_i), \mathbf{w} \rangle / N$ but this time over the orthogonal projection π_V of the signatures. To further understand the results achieved by this hybrid method, we perform an analysis of the behaviour of the subspace spanned by our k eigenvectors with respect to the number of samples.

Convergence of eigenvectors in a subspace Even though we can have a precise description of the convergence of eigenvalues, the main objects of interest are eigenvectors. The relation of their behaviour with the eigenvalues' is not trivial, and the Davis-Kahan theorem [8] can be used to show a link between them. In our case, the Davis-Kahan theorem allows to quantify how close the k-dimensional subspace V is to the space W of the k eigenvectors corresponding to the k largest eigenvalues of the covariance matrix. In particular, their "closeness" is roughly upper-bounded by $1/(\lambda_k - \lambda_{k+1})$. However, this bound performs pretty poorly in our case since the gap between successive eigenvalues might not increase with the dimension. This can be explained by the fact that the Davis-Kahan theorem describes the convergence of the whole subspace V to W, while we are only interested in the convergence of a small number of vectors in the space. To simplify our analysis, we will only be interested in the eigenvector $\mathbf{b}$ of the largest eigenvalue of $\tilde{\boldsymbol{\Sigma}}$, since it is always a vector that allows us to recover the secret key. This analysis can be adapted to work for any other vector. We start by providing a slightly generalized Davis-Kahan theorem.

A variant of Davis-Kahan theorem for key recovery In the following, for any matrix $\mathbf{A} \in \mathbb{R}^{n \times k}$, we define $\mathbf{P}_{\mathbf{A}} = \mathbf{A}\mathbf{A}^T$. Note that if $\mathbf{A}$ is orthonomal, the matrix $\mathbf{P}_{\mathbf{A}}$ is the orthogonal projection on the space spanned by its columns. For any matrix $\mathbf{X} \in \mathbb{R}^{k \times \ell}$, we use both operator notation $\mathbf{P}_{\mathbf{A}}(\mathbf{X})$ or the matrix multiplication notation $\mathbf{P}_{\mathbf{A}}\mathbf{X}$ for the same operation.

Lemma 7. *[62, Lemma 5] Set $\mathbf{A} \in \mathbb{R}^{n \times m}$. Then for any $\mathbf{U} \in \mathbb{R}^{k \times n}, \mathbf{W} \in \mathbb{R}^{m \times \ell}$ that both have orthonormal rows or orthonormal columns, we have*

$$\|\mathbf{U}^T\mathbf{A}\mathbf{W}\|_F \leq \|\mathbf{A}\|_F$$

.

In particular, this implies that for any subset of rows or columns $\mathbf{X}$ of $\mathbf{A}$,

$$\|\mathbf{X}\|_F \leq \|\mathbf{A}\|_F$$

Theorem 2. *Let $\boldsymbol{\Sigma}, \tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{n\times n}$ be symmetric matrices with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ and $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n \geq 0$ respectively. Let $1 \leq r \leq d \leq n$, $\lambda_{n+1} = -\infty$ and assume that $\lambda_r \neq \lambda_{d+1}$. Let $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_d)$ and $\tilde{\mathbf{V}} = (\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_r)$ be orthonomal basis formed of eigenvectors such that $\boldsymbol{\Sigma}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ and $\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{v}}_i = \tilde{\lambda}_i \tilde{\mathbf{v}}_i$ for $1 \leq i \leq d$. Then*

$$\|\mathbf{P}_{\mathbf{V}^\perp}(\tilde{\mathbf{V}})\|_F \leq \frac{2\min(r^{1/2}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2, \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F)}{\lambda_r - \lambda_{d+1}}$$

where $\mathbf{V}^\perp \in \mathbb{R}^{n\times n-d}$ is any orthogonal complement of $\mathbf{V}$ that is an orthonormal basis of eigenvectors of $\boldsymbol{\Sigma}$.

Equivalently,

$$\|\mathbf{P}_{\mathbf{V}}(\tilde{\mathbf{V}})\|_F^2 \geq r\left(1 - \frac{4\min(\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2^2, r^{-1}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F^2)}{(\lambda_r - \lambda_{d+1})^2}\right).$$

Proof. Let $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$ and $\tilde{\boldsymbol{\Lambda}} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_r)$. The proof follows the lines of [62]. The goal is to get the inequality by bounding the "eigenvector defect" $\|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F$ of basis $\tilde{\mathbf{V}}$ with respect to $\boldsymbol{\Sigma}$.

First we derive the wanted upper bound.

$$\begin{aligned}
\|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F &= \|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}} - (\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}})\tilde{\mathbf{V}}\|_F \\
&\leq \|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}}\|_F + \|(\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}})\tilde{\mathbf{V}}\|_F \\
&\leq \|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}\|_F + \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F
\end{aligned}$$

where for the third inequality we used Lemma 7. From there, we have two ways to bound this term. By using Weyl's inequality and that for any matrix $\mathbf{A} \in \mathbb{R}^{n\times r}$, we have the inequality $\|\mathbf{A}\|_F \leq \sqrt{r}\|\mathbf{A}\|_2$, we get

$$\begin{aligned}
\|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}\|_F + \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F &\leq \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2 + \sqrt{r}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2 \\
&\leq 2\sqrt{r}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2.
\end{aligned}$$

Instead, if we use the Wielandt–Hoffman theorem, we get the bound

$$\|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}\|_F + \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F \leq 2\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F.$$

From these two inequalities, we derive part of the right-hand side of the theorem since

$$\|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F \leq \min(2\sqrt{r}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2, 2\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F).$$

We now give a lower bound of $\|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F$ to conclude the proof. To simplify notations, let $\mathbf{V}_1 = \mathbf{V}, \mathbf{V}_2 = \mathbf{V}^\perp$ and $\boldsymbol{\Lambda}_2 = \mathrm{diag}(\lambda_{d+1}, \ldots, \lambda_n)$. Recall that $\boldsymbol{\Sigma}\mathbf{V}_2 = \mathbf{V}_2\boldsymbol{\Lambda}_2$.

Since $\mathbf{V}_2$ is an orthogonal complement of $\mathbf{V}_1$, we have $\mathbf{I}_n = \mathbf{P}_{\mathbf{V}_1} + \mathbf{P}_{\mathbf{V}_2}$. From this, we derive

$$\begin{aligned}
\|\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F &= \|(\mathbf{P}_{\mathbf{V}_1} + \mathbf{P}_{\mathbf{V}_2})\tilde{\mathbf{V}}\boldsymbol{\Lambda} - (\mathbf{P}_{\mathbf{V}_1} + \mathbf{P}_{\mathbf{V}_2})\boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F \\
&= \|\mathbf{P}_{\mathbf{V}_1}\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \mathbf{P}_{\mathbf{V}_1}\boldsymbol{\Sigma}\tilde{\mathbf{V}} + \mathbf{P}_{\mathbf{V}_2}\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \mathbf{P}_{\mathbf{V}_2}\boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F \\
&\geq \|\mathbf{P}_{\mathbf{V}_2}\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \mathbf{P}_{\mathbf{V}_2}\boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F,
\end{aligned}$$

where the last inequality is a consequence of the fact that $\|\mathbf{A}\|_F = \mathrm{Tr}(\mathbf{A}^T\mathbf{A})$ and $\mathbf{P}_{\mathbf{V}_2}^T\mathbf{P}_{\mathbf{V}_1} = \mathbf{0}$. Given that $\mathbf{P}_{\mathbf{V}_2} = \mathbf{V}_2\mathbf{V}_2^T$, we get the equality $\mathbf{P}_{\mathbf{V}_2}\boldsymbol{\Sigma} = \mathbf{V}_2\boldsymbol{\Lambda}_2\mathbf{V}_2^T = \mathbf{P}_{\mathbf{V}_2\boldsymbol{\Lambda}_2^{1/2}}$. This gives

$$\begin{aligned}
\|\mathbf{P}_{\mathbf{V}_2}\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \mathbf{P}_{\mathbf{V}_2}\boldsymbol{\Sigma}\tilde{\mathbf{V}}\|_F &= \|\mathbf{P}_{\mathbf{V}_2}\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \mathbf{P}_{\mathbf{V}_2\boldsymbol{\Lambda}_2^{1/2}}\tilde{\mathbf{V}}\|_F \\
&= \|\mathbf{V}_2(\mathbf{V}_2^T\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_2\mathbf{V}_2^T\tilde{\mathbf{V}})\|_F \\
&= \|\mathbf{V}_2^T\tilde{\mathbf{V}}\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_2\mathbf{V}_2^T\tilde{\mathbf{V}}\|_F \\
&\geq (\lambda_r - \lambda_{d+1})\|\mathbf{V}_2^T\tilde{\mathbf{V}}\|_F
\end{aligned}$$

where the last inequality is obtained by developping the Frobenius norm of the matrix since, for any matrix $\mathbf{H} = (h_{i,j}) \in \mathbb{R}^{r \times n-d}$, the following equality holds

$$\mathbf{H}\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_2\mathbf{H} = ((\lambda_i - \lambda_{d+j})h_{i,j}) \,.$$

Since $\|\mathbf{V}_2^T\tilde{\mathbf{V}}\|_F = \text{Tr}(\tilde{\mathbf{V}}^T\mathbf{V}_2\mathbf{V}_2^T\tilde{\mathbf{V}}) = \text{Tr}(\tilde{\mathbf{V}}^T\mathbf{V}_2\mathbf{V}_2^T\mathbf{V}_2\mathbf{V}_2^T\tilde{\mathbf{V}}) = \|\mathbf{P}_{\mathbf{V}_2}(\tilde{\mathbf{V}})\|_F$, we conclude that

$$(\lambda_r - \lambda_{d+1})\|\mathbf{P}_{\mathbf{V}_2}(\tilde{\mathbf{V}})\|_F \leq 2\min(\sqrt{r}\|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_2, \|\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}\|_F)$$

which is the inequality we wanted to prove.

The second inequality of the theorem is a direct consequence of $\|\mathbf{X}\|_F^2 = \|\mathbf{P}_{\mathbf{V}_1}(\mathbf{X})\|_F^2 + \|\mathbf{P}_{\mathbf{V}_2}(\mathbf{X})\|_F^2$ for any matrix $\mathbf{X}$.

Remark 1. This proof can be generalized for any $\tilde{\mathbf{V}}$ that is a subset of eigenvectors of size $r \leq d$.

In our case, we are interested in the distance between the eigenvector $\mathbf{b}$ corresponding to the largest eigenvalue of $\tilde{\boldsymbol{\Sigma}}$ and the subspace $\bar{\mathbf{V}}$ spanned by the k-first eigenvectors of the empirical covariance matrix $\bar{\boldsymbol{\Sigma}}$.

Corollary 2. *Under the assumption that* $\lambda_1 - \lambda_{k+1} > 2\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_F$,

$$\|\mathbf{P}_{\bar{\mathbf{V}}}(\mathbf{b})\|_F^2 \geq 1 - \frac{4\min(\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_2^2, \|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_F^2)}{(\lambda_1 - \lambda_{k+1} - 2\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_F)^2}.$$

Proof. We apply Theorem 2 to bound $\|\mathbf{P}_{\bar{\mathbf{V}}}(\mathbf{b})\|_F^2$ with respect to the eigenvalues $\bar{\lambda}_1, \bar{\lambda}_{k+1}$ of $\bar{\boldsymbol{\Sigma}}$. By Weyl's inequality, we get the inequality

$$\bar{\lambda}_1 - \bar{\lambda}_{k+1} \geq \lambda_1 - \lambda_{k+1} - 2\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_F.$$

Combining this inequality and the bound of Theorem 2 concludes the proof.

It remains to analyse the term $\min(\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_2^2, \|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|_F^2)$ in our upper bound.

Bounding the convergence Let $\bar{\mathbf{V}}$ be the space spanned by the k-first eigenvectors of the empirical covariance matrix $\bar{\boldsymbol{\Sigma}}$. Corollary 2 shows that the convergence of $\mathbf{b}$ into $\bar{\mathbf{V}}$ is controlled by two quantities: the quality of the approximation $\|\tilde{\boldsymbol{\Sigma}} - \bar{\boldsymbol{\Sigma}}\|$ and the gap between eigenvalues $\lambda_1 - \lambda_{k+1}$. While the latter depends on the keypair, the former can be analyzed using known results on the covariance estimation of sub-Gaussian distributions (which applies to our distribution, since it is bounded).

Lemma 8. *[61, Corollary 5.50] Consider a sub-Gaussian distribution in* $\mathbb{R}^n$ *with covariance matrix* $\boldsymbol{\Sigma}$*, and let* $\epsilon \in (0, 1)$, $t \geq 1$. *With probability at least* $1 - 2\exp(-t^2 n)$*, if* $N \geq C(t/\epsilon)^2 n$*, one has*

$$\|\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_N\|_F \leq \epsilon,$$

where $\boldsymbol{\Sigma}_N$ *is the sample covariance matrix with* N *sample and* C *is a constant.*

Plugging this with Corollary 2 gives us

$$\|\mathbf{P}_{\bar{\mathbf{V}}}(\mathbf{b})\|_F^2 \geq 1 - \frac{4\epsilon^2}{(\lambda_1 - \lambda_{k+1} - 2\epsilon)^2},$$

with probability $1 - 2\exp(-n)$, assuming that the sample size N is greater than $C(1/\epsilon)^2 n$.

6 Experiments

Experiments were made on a server with an Intel Xeon Platinum 8160 CPU (48 2.10GHz cores) and 4 NVIDIA Quadro GV100 GPUs.

To generate the faulty signatures and emulate the effects of the rowhammer attack, we manually perform the bitflips in the sampler of the official distribution of FALCON. The optimization process in the Nguyen-Regev attack is implemented using the GPU library PyTorch to further accelerate computations.

The full attack starts by generating faulty signatures and process them on the fly to compute an approximation of the Hermitian covariance matrix $\tilde{\boldsymbol{\Sigma}}$ and extract a PCA subspace corresponding to a certain number of top eigenvalues (we use complex dimension $k = 8$, real dimension $2k = 16$).

After that, new signatures are generated in parallel by the CPU and their projections on the PCA subspace are stored in memory. This (smaller) pool of projected signatures is used to compute the fourth moment function for the Nguyen-Regev attack.

Performing optimization on this smaller dimensional subspace helps both the probability to reach the target vector and the speed of convergence because of the considerably smaller space to search and the reduced memory footprint on the GPU of the projected signatures compared to the full signatures. Points computed as results of the optimization are stored and, using Corollary 1, estimates of $\mathbf{b}_0$ are computed. After this, if no suitable candidates are found, we increase the dimension k of the subspace, until it reaches so maximum value (in our experiments, dimensions bigger than $2k = 32$ did not significantly improve the results).

The number of signatures needed for the covariance estimation can be evaluated using Corollary 2, which ensures that the correlation r of an eigenvector for the top eigenvalue with our $2k$-dimensional subspace approaches 1 inversely proportionally to the number of signatures and the eigenvalue gap between the top eigenvalue and the $2k + 1$-st eigenvalue: there exists a constant C such that:

$$|1 - r| \leq \frac{C}{N g_{2k}}$$

where g_{2k} is the $(2k + 1)$-st real eigenvalue gap (equivalently, the $(k + 1)$-st Hermitian eigenvalue gap). Moreover, in the 1-bit flip setting, we have $C \approx 1.7 \cdot 10^7$ (this can be estimated using the fourth moment of the one-dimensional distribution).

Since, by routine computations, we find that we need a correlation of around 0.999 to obtain a vector that rounds to the key, we can deduce the required number of signatures based on eigenvalue gaps for random FALCON keys. The graph below show the number of signatures in million needed to mount a successful attack, based on the first few complex eigenvalue gaps among 1000 FALCON keys:

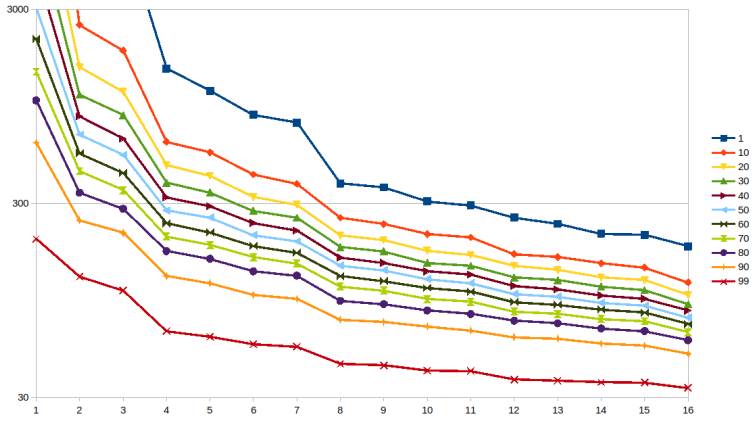

Fig. 5. Required signatures for 1 bit faults, in millions

We see that the $k = 8$ attack required fewer than 200 millions signatures for $> 70\%$ of keys. To confirm our estimates, we run the attack for several attack parameters. To compare with the Nguyen-Regev attack, we run both attacks with the same basis and recover 1000 points, each being potential solutions. Using Corollary 1, we try to recover $\mathbf{b}_0$ and compare with the secret key to assert our success.

In the one bit flip case, we perform the attack with up to 300 million signatures, and successfully recover the key with the PCA-aided attack in 10/10 experiments after only around 20 million signatures to estimate the fourth moment function, projected on a subspace of dimension $k \in [16, 32]$.

In the eight bit flips case, the full attack succeeds with only 20 million signatures, including 2 million for the Nguyen-Regev phase.

7 Countermeasures

In this section, we discuss ways to mitigate the attack. Of course, any hardware countermeasure allowing to prevent Rowhammer attacks will directly affect this attack, but we decide to focus on the countermeasures specific to FALCON, since our attacks mainly relies on the fault being injected rather than on the specific technique employed. We suggest two main countermeasures with minimal overhead on the overall computation.

RCDT integrity check. The first countermeasure is simply to check the integrity of the RCDT by, e.g., verifying that its SHA-3 digest matches the one of the correct RCDT. Since computing SHA-3 on data of the size of the RCDT takes negligible time compared to the rest of signature generation (and since Keccak is already part of FALCON's algorithm), this can ensure at minimal cost that the RCDT has not been tampered with.

Of course, an attacker able to inject an arbitrarily large number of controlled bit flips could in principle modify not only the RCDT but also the hash digest to obtain matching values. However, Rowhammer attacks are limited in practice in the number of bit flips they can achieve, and finding a bit flip pattern on the RCDT that also flips at most a few bits of the hash digest is a problem that is clearly hard in the random oracle model, and hence expected to be hard for SHA-3 as well.

Rejecting exceedingly small signatures. By virtue of their high-dimensional Gaussian distribution, correctly generated FALCON signatures have a squared Euclidean norm that sharply concentrates around the expected value $V = 2n\sigma^2$. In fact, the FALCON signing algorithm rejects signatures of squared Euclidean norm above $1.1^2 \cdot V$ with little impact, since they happen with probability $< 10^{-5}$. Likewise, signatures of square Euclidean norm below $0.9^2 \cdot V$ also appear with probability $< 10^{-5}$; adding a check to reject them, both in signature generation and signature verification, is essentially cost-free in terms of performance, and would go a long way towards mitigating the type of attack considered in this paper.

Indeed, signatures obtained from the faulty sampler we consider are significantly shorter than correctly generated FALCON signatures: even in our single bit flip attack, the expected squared Euclidean norm goes down below $0.52 \cdot S$, and the probability of the squared Euclidean norm satisfying the lower bound $0.9^2 \cdot S$ is around 10^{-10}, making it impractical to generate even a single signature passing the added check, let alone sufficiently many to mount the attack.

One could imagine a variant of the attack that would perturb the RCDT even less (e.g., flipping a lower order bit of another, less impactful coefficient than the first one) in such a way as to pass the added check with higher probability; however, this would also considerably increase the number of required signatures for successful key recovery. All in all, we conjecture that the added check basically eliminates the chance of a practical Rowhammer attack based on the ideas of this paper.

As an added benefit, it also acts as a sanity check against implementation bugs, like the one that briefly affected the official implementation of FALCON in 2019 [50]: indeed, signatures generated by that buggy implementation also fail this check with high probability.

References

1. Amer, S., Wang, Y., Kippen, H., Dang, T., Genkin, D., Kwong, A., Nelson, A., Yerukhimovich, A.: PQ-Hammer: End-to-end key recovery attacks on post-quantum cryptography using Rowhammer. In: IEEE S&P 2025. pp. 48–48.

IEEE Computer Society (2025). , https://doi.ieeecomputersociety.org/10.1109/SP61157.2025.00048
2. Babai, L.: On lovász' lattice reduction and the nearest lattice point problem (shortened version). In: Mehlhorn, K. (ed.) STACS'85. LNCS, vol. 182, pp. 13–20. Springer (1985). , https://doi.org/10.1007/BFb0023990
3. Bhattacharya, S., Mukhopadhyay, D.: Curious case of rowhammer: Flipping secret exponent bits using timing analysis. In: Gierlichs, B., Poschmann, A.Y. (eds.) CHES 2016. LNCS, vol. 9813, pp. 602–624. Springer, Berlin, Heidelberg (Aug 2016).
4. Bruinderink, L.G., Pessl, P.: Differential fault attacks on deterministic lattice signatures. IACR TCHES **2018**(3), 21–43 (2018). , https://tches.iacr.org/index.php/TCHES/article/view/7267
5. Chen, Y., Genise, N., Mukherjee, P.: Approximate trapdoors for lattices and smaller hash-and-sign signatures. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 3–32. Springer, Cham (Dec 2019).
6. Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., Xagawa, K.: ModFalcon: Compact signatures based on module-NTRU lattices. In: Sun, H.M., Shieh, S.P., Gu, G., Ateniese, G. (eds.) ASIACCS 20. pp. 853–866. ACM Press (Oct 2020).
7. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with side information: Attacks and concrete security estimation. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 329–358. Springer, Cham (Aug 2020).
8. Davis, C., Kahan, W.M.: The rotation of eigenvectors by a perturbation. iii. SIAM Journal on Numerical Analysis **7**(1), 1–46 (1970), http://www.jstor.org/stable/2949580
9. Delvaux, J.: Roulette: A diverse family of feasible fault attacks on masked Kyber. IACR TCHES **2022**(4), 637–660 (2022).
10. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 22–41. Springer, Berlin, Heidelberg (Dec 2014).
11. Ducas, L., Nguyen, P.Q.: Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 433–450. Springer, Berlin, Heidelberg (Dec 2012).
12. Ducas, L., Prest, T.: Fast Fourier orthogonalization. In: Abramov, S.A., Zima, E.V., Gao, X. (eds.) ISSAC 2016. pp. 191–198. ACM (2016). , https://doi.org/10.1145/2930889.2930923
13. Ducas, L., Yu, Y.: Learning strikes again: The case of the DRS signature scheme. Journal of Cryptology **34**(1), 1 (Jan 2021).
14. Espitau, T., Fouque, P.A., Gérard, B., Tibouchi, M.: Loop-abort faults on lattice-based fiat—shamir and hash-and-sign signatures. Cryptology ePrint Archive, Report 2016/449 (2016), https://eprint.iacr.org/2016/449
15. Espitau, T., Fouque, P.A., Gérard, B., Tibouchi, M.: Loop-abort faults on lattice-based Fiat-Shamir and hash-and-sign signatures. In: Avanzi, R., Heys, H.M. (eds.) SAC 2016. LNCS, vol. 10532, pp. 140–158. Springer, Cham (Aug 2016).
16. 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: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 222–253. Springer, Cham (May / Jun 2022).
17. Espitau, T., Nguyen, T.T.Q., Sun, C., Tibouchi, M., Wallet, A.: Antrag: Annular NTRU trapdoor generation - making mitaka as secure as falcon. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 3–36. Springer, Singapore (Dec 2023).
18. Espitau, T., Niot, G., Sun, C., Tibouchi, M.: SQUIRRELS — Square Unstructured Integer Euclidean Lattice Signature. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
19. Fahr, M., Kippen, H., Kwong, A., Dang, T., Lichtinger, J., Dachman-Soled, D., Genkin, D., Nelson, A., Perlner, R.A., Yerukhimovich, A., Apon, D.: When frodo flips: End-to-end key recovery on FrodoKEM via rowhammer. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 979–993. ACM Press (Nov 2022).
20. 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: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 34–63. Springer, Cham (May 2020).
21. Frieze, A.M., Jerrum, M., Kannan, R.: Learning linear transformations. In: 37th FOCS. pp. 359–368. IEEE Computer Society Press (Oct 1996).
22. Frigo, P., Vannacci, E., Hassan, H., van der Veen, V., Mutlu, O., Giuffrida, C., Bos, H., Razavi, K.: TRRespass: Exploiting the many sides of target row refresh. In: 2020 IEEE Symposium on Security and Privacy. pp. 747–762. IEEE Computer Society Press (May 2020).
23. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008).
24. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski, Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 112–131. Springer, Berlin, Heidelberg (Aug 1997).

25. Gruss, D., Lipp, M., Schwarz, M., Genkin, D., Juffinger, J., O'Connell, S., Schoechl, W., Yarom, Y.: Another flip in the wall of rowhammer defenses. In: 2018 IEEE Symposium on Security and Privacy. pp. 245–261. IEEE Computer Society Press (May 2018).
26. Gruss, D., Maurice, C., Mangard, S.: Rowhammer.js: A remote software-induced fault attack in JavaScript. In: Caballero, J., Zurutuza, U., Rodríguez, R.J. (eds.) DIMVA 2016. LNCS, vol. 9721, pp. 300–321. Springer (2016). , https://doi.org/10.1007/978-3-319-40667-1_15
27. Guerreau, M., Martinelli, A., Ricosset, T., Rossi, M.: The hidden parallelepiped is back again: Power analysis attacks on falcon. IACR TCHES **2022**(3), 141–164 (2022).
28. Hermelink, J., Pessl, P., Pöppelmann, T.: Fault-enabled chosen-ciphertext attacks on kyber. In: Adhikari, A., Küsters, R., Preneel, B. (eds.) INDOCRYPT 2021. LNCS, vol. 13143, pp. 311–334. Springer, Cham (Dec 2021).
29. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital signatures using the NTRU lattice. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 122–140. Springer, Berlin, Heidelberg (Apr 2003).
30. 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 (2020).
31. Islam, S., Mus, K., Singh, R., Schaumont, P., Sunar, B.: Signature correction attack on dilithium signature scheme. In: 2022 IEEE European Symposium on Security and Privacy. pp. 647–663. IEEE Computer Society Press (Jun 2022).
32. Karabulut, E., Aysu, A.: Falcon down: Breaking Falcon post-quantum signature scheme through side-channel attacks. In: DAC 2021 (2021)
33. Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J.H., Lee, D., Wilkerson, C., Lai, K., Mutlu, O.: Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In: ISCA 2014. pp. 361–372. IEEE Computer Society (2014)
34. Klein, P.N.: Finding the closest lattice vector when it's unusually close. In: Shmoys, D.B. (ed.) 11th SODA. pp. 937–941. ACM-SIAM (Jan 2000)
35. Kwong, A., Genkin, D., Gruss, D., Yarom, Y.: RAMBleed: Reading bits in memory without accessing them. In: 2020 IEEE Symposium on Security and Privacy. pp. 695–711. IEEE Computer Society Press (May 2020).
36. Lin, X., Suzuki, M., Zhang, S., Espitau, T., Yu, Y., Tibouchi, M., Abe, M.: Cryptanalysis of the Peregrine lattice-based signature scheme. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part I. LNCS, vol. 14601, pp. 387–412. Springer, Cham (Apr 2024).
37. Lin, X., Tibouchi, M., Yu, Y., Zhang, S.: Do not disturb a sleeping Falcon: Floating-point error sensitivity of the Falcon sampler and its consequences. Cryptology ePrint Archive, Paper 2024/1709 (2024), https://eprint.iacr.org/2024/1709
38. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Berlin, Heidelberg (Dec 2009).
39. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Berlin, Heidelberg (Apr 2012).
40. 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
41. 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
42. Mus, K., Doröz, Y., Tol, M.C., Rahman, K., Sunar, B.: Jolt: Recovering TLS signing keys via rowhammer faults. In: 2023 IEEE Symposium on Security and Privacy. pp. 1719–1736. IEEE Computer Society Press (May 2023).
43. Mus, K., Islam, S., Sunar, B.: QuantumHammer: A practical hybrid attack on the LUOV signature scheme. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 1071–1084. ACM Press (Nov 2020).
44. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 271–288. Springer, Berlin, Heidelberg (May / Jun 2006).
45. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. Journal of Cryptology **22**(2), 139–160 (Apr 2009).
46. Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80–97. Springer, Berlin, Heidelberg (Aug 2010).
47. Pessl, P., Prokop, L.: Fault attacks on CCA-secure lattice KEMs. IACR TCHES **2021**(2), 37–60 (2021). , https://tches.iacr.org/index.php/TCHES/article/view/8787
48. Poddebniak, D., Somorovsky, J., Schinzel, S., Lochter, M., Rösler, P.: Attacking deterministic signature schemes using fault attacks. In: 2018 IEEE European Symposium on Security and Privacy. pp. 338–352. IEEE Computer Society Press (Apr 2018).

49. Pornin, T.: New efficient, constant-time implementations of Falcon. Cryptology ePrint Archive, Report 2019/893 (2019), https://eprint.iacr.org/2019/893
50. Pornin, T.: OFFICIAL COMMENT: Falcon (bug & fixes). pqc-forum official comment (2019), https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/7Z8x5AMXy8s/m/Spyv8VYoBQAJ
51. Postlethwaite, E.W., van Woerden, W.P.J.: OFFICIAL COMMENT: EHTv3. pqc-forum official comment (2023), https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/mFl_5Rq6-RU
52. Prest, T.: Gaussian Sampling in Lattice-Based Cryptography. Ph.D. thesis, École Normale Supérieure, Paris, France (2015)
53. 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
54. Ravi, P., Jhanwar, M.P., Howe, J., Chattopadhyay, A., Bhasin, S.: Exploiting determinism in lattice-based signatures: Practical fault attacks on pqm4 implementations of NIST candidates. In: Galbraith, S.D., Russello, G., Susilo, W., Gollmann, D., Kirda, E., Liang, Z. (eds.) ASIACCS 19. pp. 427–440. ACM Press (Jul 2019).
55. Ravi, P., Roy, D.B., Bhasin, S., Chattopadhyay, A., Mukhopadhyay, D.: Number "not used" once - practical fault attack on pqm4 implementations of NIST candidates. In: Polian, I., Stöttinger, M. (eds.) COSADE 2019. LNCS, vol. 11421, pp. 232–250. Springer, Cham (Apr 2019).
56. Ravi, P., Yang, B., Bhasin, S., Zhang, F., Chattopadhyay, A.: Fiddling the twiddle constants - fault injection analysis of the number theoretic transform. IACR TCHES **2023**(2), 447–481 (2023).
57. Razavi, K., Gras, B., Bosman, E., Preneel, B., Giuffrida, C., Bos, H.: Flip feng shui: Hammering a needle in the software stack. In: Holz, T., Savage, S. (eds.) USENIX Security 2016. pp. 1–18. USENIX Association (Aug 2016)
58. de Ridder, F., Frigo, P., Vannacci, E., Bos, H., Giuffrida, C., Razavi, K.: SMASH: Synchronized many-sided rowhammer attacks from JavaScript. In: Bailey, M., Greenstadt, R. (eds.) USENIX Security 2021. pp. 1001–1018. USENIX Association (Aug 2021)
59. 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
60. Tatar, A., Giuffrida, C., Bos, H., Razavi, K.: Defeating software mitigations against rowhammer: A surgical precision hammer. In: Bailey, M.D., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID 2018. LNCS, vol. 11050, pp. 47–66. Springer (2018). , https://doi.org/10.1007/978-3-030-00470-5_3
61. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027 (2010)
62. Wang, T.: A useful variant of the davis–kahan theorem for statisticians. vol. 7, pp. 1–46
63. Weissman, Z., Tiemann, T., Moghimi, D., Custodio, E., Eisenbarth, T., Sunar, B.: JackHammer: Efficient Rowhammer on heterogeneous FPGA-CPU platforms. IACR TCHES **2020**(3), 169–195 (2020). , https://tches.iacr.org/index.php/TCHES/article/view/8587
64. Xagawa, K., Ito, A., Ueno, R., Takahashi, J., Homma, N.: Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part II. LNCS, vol. 13091, pp. 33–61. Springer, Cham (Dec 2021).
65. Xiao, Y., Zhang, X., Zhang, Y., Teodorescu, R.: One bit flips, one cloud flops: Cross-VM row hammer attacks and privilege escalation. In: Holz, T., Savage, S. (eds.) USENIX Security 2016. pp. 19–35. USENIX Association (Aug 2016)
66. Yu, Y., Jia, H., Li, L., Ran, D., Qiu, Z., Zhang, S., Lin, X., Wang, X.: HuFu — Hash-and-Sign Signatures From Powerful Gadgets. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
67. Zhang, F., Lou, X., Zhao, X., Bhasin, S., He, W., Ding, R., Qureshi, S., Ren, K.: Persistent fault analysis on block ciphers. IACR TCHES **2018**(3), 150–172 (2018). , https://tches.iacr.org/index.php/TCHES/article/view/7272
68. Zhang, S., Lin, X., Yu, Y., Wang, W.: Improved power analysis attacks on falcon. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part IV. LNCS, vol. 14007, pp. 565–595. Springer, Cham (Apr 2023).