# A Key-Recovery Attack against Mitaka in the $t$-Probing Model

Thomas Prest

thomas.prest@pqshield.com

PQShield

**Abstract.** Mitaka is a lattice-based signature proposed at Eurocrypt 2022. A key advertised feature of Mitaka is that it can be masked at high orders efficiently, making it attractive in scenarios where side-channel attacks are a concern. Mitaka comes with a claimed security proof in the $t$-probing model.

We uncover a flaw in the security proof of Mitaka, and subsequently show that it is not secure in the $t$-probing model. For any number of shares $d \geq 4$, probing $t < d$ variables per execution allows an attacker to recover the private key efficiently with approximately $2^{21}$ executions. Our analysis shows that even a constant number of probes suffices ($t = 3$), as long as the attacker has access to a number of executions that is linear in $d/t$.

**Keywords.** Mitaka, $t$-probing model, cryptanalysis.

## 1 Introduction

In the last decade, post-quantum cryptography has been an extremely dynamic research and engineering field. One of the main catalysts of this dynamism is the NIST post-quantum cryptography standardization project, which in July 2022 has announced its first standards for key establishment and stateless digital signatures [NIS22]. Two of the three selected standards for signatures are based on lattices: Dilithium [LDK+22] and Falcon [PFH+22]. Dilithium and Falcon are both based on structured lattices. They achieve good computational and bandwidth efficiency, and the underlying mathematical assumptions are well-understood.

When considering concrete security, it becomes important to consider side-channel attacks, in which adversaries may learn information about the behavior of the device executing the algorithm. Side-channel attacks based on power consumption [KJJ99], running time [Koc96], electromagnetic emissions [GMO01] and even acoustic emissions [AA04,GST14] have shown to be relevant.

The main countermeasure against side-channel attacks is masking [ISW03]. It consists of splitting sensitive information in $d$ shares (concretely: $x = x_0 + \cdots + x_{d-1}$), and of performing secure computation using MPC-based techniques.

In practice, the cost of a side-channel attack is expected to grow exponentially in the number of shares $d$ [DFS19].

In parallel, leakage models have been developed in order to reason and prove statements about side-channel countermeasures. The most standard model is the $t$-probing model [ISW03], in which an attacker is allowed to learn the value of $t$ variables during each execution of the protected algorithm. While not being the most realistic leakage model, the $t$-probing model is arguably the easiest to work in, especially when considering masking. In addition, proving security in this model is usually a good indicator of security, especially when augmenting the $t$-probing model with proof frameworks such as the SNI, PINI or IOS models.

Unfortunately, Dilithium and Falcon are not straightforward to mask. In the case of Dilithium, sampling from specific distributions and rejection sampling are two examples of operations that require conversions between Boolean and arithmetic representations (called A2B and B2A conversions), which is expensive when operating on masked values. Falcon seems even more challenging to mask, due to its intricate use of floating-point operations.

Mitaka [EFG+22] is a variant of Falcon that was proposed in order to address these caveats. One of the main advertised features of Mitaka is that it is easy to mask. This was done by proposing new algorithms for performing masked operations. One such algorithm is GaussShareByShare (Algorithm 3), which performs Gaussian sampling over the integers efficiently and with no A2B or B2A conversion. Mitaka comes complete with a claimed security proof in the $t$-probing model [EFG+22, Theorems 4 and 5], for $t < d$.

### 1.1 Our contribution

We show that Mitaka is insecure in the $t$-probing model. More precisely, by targeting one specific call to GaussShareByShare, and probing $t < d$ specific values inside that execution, a $t$-probing attacker can compute a vector that is correlated to the private key $\mathbf{b}_0$. By combining sufficiently many of these vectors, the attacker can compute an estimator $\widehat{\mathbf{b}_0}$ that is a noisy version of the $\mathbf{b}_0$, which can then be recovered by lattice reduction attacks, or simple rounding, depending on the number of probes $t$, masking order $d$ and number of executions $N$.

Concretely, we are able to recover the private key with $N = 2^{21}$ executions of the signing algorithm and, for each execution, the values of the probed variables, which we call traces. The efficiency behavior of our attack is illustrated in Fig. 1.

More worryingly, our attack remains feasible even if $d$ is polynomially large and $t$ is constant, since we only need the number of traces $N$ to be linear in $d/t$. A generic countermeasure against our attack is to replace GaussShareByShare by more classical conversion-based techniques, but we expect this to incur a significant overhead on the computational cost of Mitaka.

As part of our attack, we propose in Section 5.4 a simple trick which speeds up considerably the recovery of $\mathbf{b}_0$ from the estimator $\widehat{\mathbf{b}_0}$ in many relevant regimes. This trick also applies to a recently proposed power analysis on Falcon [GMRR22], and may have other applications as well.

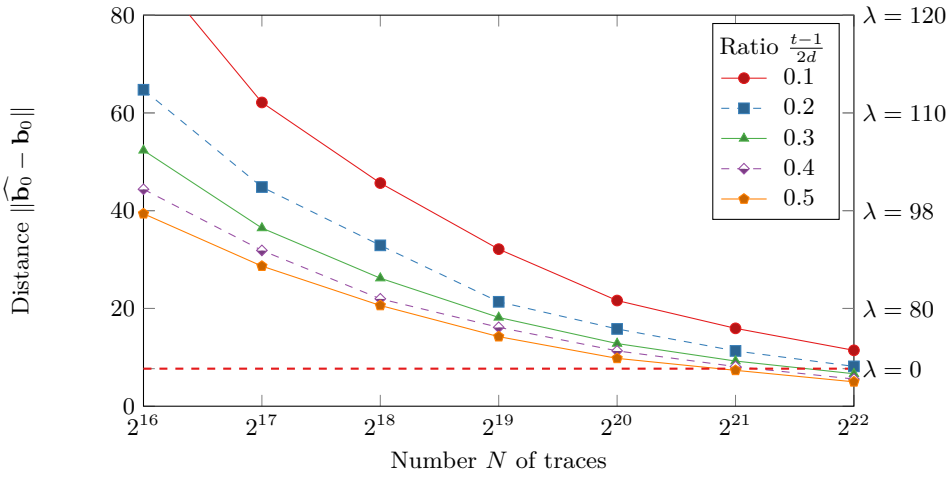



Fig. 1: Distance of the estimator $\widehat{\mathbf{b}_0}$ to the private key $\mathbf{b}_0$ as a function of the number of traces ($x$-axis) and the ratio $\frac{t-1}{2d}$. The marks $\{\lambda = x\}$ on the right side indicate the core-SVP hardness of the lattice problem we need to solve. Under the line {- - -}, $\mathbf{b}_0$ can be recovered in polynomial time (Section 5.4).

## 2 Preliminaries

Given $n \in \mathbb{N}, n > 0$, we may note $[n] = \{0, \ldots, n-1\}$.

### 2.1 Operators and relations

As a mnemonic device, we note $\mathsf{out} := f(\mathsf{in})$ (resp. $\mathsf{out} \leftarrow f(\mathsf{in})$ and $f(\mathsf{in}) \rightarrow \mathsf{out}$) to indicate that $\mathsf{out}$ is a deterministic (resp. randomized) function of $\mathsf{in}$.

We assume familiarity with the asymptotic notation: $O(\cdot), o(\cdot), \Theta(\cdot)$, Knuth's $\Omega(\cdot)$ and so on. We use the notation $x \sim y$ as shorthand for $x - y = o(x)$.

We employ the notation $x \overset{s}{\sim} X$ to indicate that the distribution of $x$ is statistically close to $X$. Finally, $F \simeq G$ indicates that $F$ and $G$ are isomorphic.

### 2.2 Cyclotomic fields

For efficiency reasons, schemes such as Falcon and Mitaka work over cyclotomic number fields. Given $n \in \mathbb{N}$ a fixed power-of-two and $\zeta \in \mathbb{C}$ a primitive $2n$-root of unity, we define the cyclotomic field $\mathcal{K}$ and its corresponding ring of integers $\mathcal{R} \subset \mathcal{K}$:

$$\begin{aligned} \mathcal{K} &= \mathbb{Q}(\zeta) \simeq \mathbb{Q}[x]/(x^n + 1) \\ \mathcal{R} &= \mathbb{Z}[\zeta] \ \simeq \mathbb{Z}[x]/(x^n + 1) \end{aligned}$$

It is often convenient to think of and represent elements of $\mathcal{K}$ and $\mathcal{R}$ as polynomials modulo $(x^n + 1)$. We can embed $\mathcal{K}$ (and thus $\mathcal{R}$) with an inner product:

$$\langle a, b \rangle_{\mathcal{K}} = a^* \cdot b,$$

where $a^*$ is the adjoint of $a$, that is the unique $a^* \in \mathcal{K}$ such that $a^*(\zeta^k) := \overline{a(\zeta^k)}$ for all odd values of $k$, where $\bar{\cdot}$ denotes the complex conjugation in $\mathbb{C}$. We note $\mathbb{R}^{++} = \{x \in \mathbb{R} | x > 0\}$, and $a \in \mathcal{K}^{++}$ if $a^*(\zeta^k) \in \mathbb{R}^{++}$ for all odd values of $k$.

The polynomial representation of elements in $\mathcal{K}$ naturally entails a mapping $\mathcal{K} \to \mathbb{R}^n$, which allows to define, for $a, b \in \mathcal{K}$, the dot product $\langle a, b \rangle_{\mathbb{R}}$ as the usual dot product of their vectors of coefficients. We note that $\langle a, a \rangle_{\mathbb{R}} > 0$, so we can likewise define the norm $\|a\|_{\mathbb{R}} = \sqrt{\langle a, a \rangle_{\mathbb{Q}}}$.

### 2.3 Vectors and matrices

We note vectors (resp. matrices) with entries in $\mathbb{Q}$ or $\mathcal{K}$ using lowercase (resp. uppercase) bold letters, for example $\mathbf{v}$ (resp. $\mathbf{M}$). We use the column convention for matrices.

We also note $\mathbf{x}^*$ the transposition of the coefficient-wise adjoint of $\mathbf{x}$. We extend the inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ to vectors $\mathbf{a} = (a_i), \mathbf{b} = (b_i) \in \mathcal{K}^m$:

$$\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{K}} = \sum_i \langle a_i, b_i \rangle_{\mathcal{K}}$$

Likewise, we extend the notations $\langle \cdot, \cdot \rangle_{\mathbb{R}}$, $\| \cdot \|_{\mathbb{R}}$, and the notion of self-adjointness to vectors. We say that $\mathbf{a}$ and $\mathbf{b}$ are $\mathcal{K}$-orthogonal if $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{K}} = 0_{\mathcal{K}}$. Given a full-rank matrix $\mathbf{B} \in \mathcal{K}^{k \times \ell}$, the Gram-Schmidt orthogonalization of $\mathbf{B}$ is the unique pair $(\mathbf{U}, \tilde{\mathbf{B}})$ such that $\mathbf{U} \in \mathcal{K}^{\ell \times \ell}$ is upper triangular with 1's on the diagonal, the columns of $\tilde{\mathbf{B}} \in \mathcal{K}^{k \times \ell}$ are pairwise orthogonal and:

$$\mathbf{B} = \tilde{\mathbf{B}} \cdot \mathbf{U}. \tag{1}$$

We say that $\mathbf{M} \in \mathbb{K}^{m \times m}$ is self-adjoint if the matrix obtained by transposing $\mathbf{M}$, followed by entry-wise application of the adjoint operator, is $\mathbf{M}$. $\mathbf{M}$ is positive definite if (i) it is self-adjoint, and (ii) $\langle \mathbf{a}, \mathbf{M} \cdot \mathbf{a} \rangle_{\mathcal{K}} \in \mathcal{K}^{++}$ for any non-zero $\mathbf{a} \in \mathcal{K}^m$.

### 2.4 Lattices and Gaussians

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$. Given a full-rank matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$, the set $\mathcal{L}(\mathbf{B}) \coloneqq \mathbf{B} \cdot \mathbb{Z}^n$ is a lattice. This representation is useful for algorithmic purposes. We can generalize this definition and define structured lattices by replacing $(\mathbb{R}, \mathbb{Z})$ in the definitions above with $(\mathcal{K}, \mathcal{R})$.

Structured lattices are convenient due to their compact representation, however they can also be interpreted as standard lattices since $\mathcal{R}$ is a $\mathbb{Z}$-module of rank $n$. More concretely, given $a \in \mathcal{K}$, we note $\mathcal{A}(a)$ the matrix $\mathcal{A}(a) = [\mathbf{a}_0, \ldots, \mathbf{a}_{n-1}]$ where each column $\mathbf{a}_i$ is the vector of coefficients of $x \cdot a$. Note

that $\mathcal{A}(a)$ is the matrix representation of the endomorphism $f \mapsto a \cdot f$ in the canonical basis of $\mathcal{K}$. In addition, $\mathcal{A} : a \in \mathcal{K} \mapsto \mathcal{A}(a) \in \mathbb{Q}^{n \times n}$ is a ring morphism.

Given a positive definite $\Sigma \in \mathcal{K}^{m \times m}$, we note $\rho_{\sqrt{\Sigma}}$ the Gaussian function defined over $\mathcal{K}^m$ as

$$\rho_{\sqrt{\Sigma}}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}^* \cdot \Sigma^{-1} \cdot \mathbf{x}\|_{\mathbb{R}}^2}{2}\right). \tag{2}$$

We may note $\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x}) = \rho_{\sqrt{\Sigma}}(\mathbf{x} - \mathbf{c})$. When $\Sigma$ is of the form $\sigma \cdot \mathbf{I}_m$, where $\sigma \in \mathcal{K}^{++}$ and $\mathbf{I}_m$ is the identity matrix, we note $\rho_{\sigma,\mathbf{c}}$ as shorthand for $\rho_{\sqrt{\Sigma},\mathbf{c}}$. For any countable set $S \subset \mathcal{K}^m$, we note $\rho_{\sqrt{\Sigma},\mathbf{c}}(S) = \sum_{\mathbf{x} \in \mathcal{K}^m} \rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x})$ whenever this sum converges. Finally, when $\rho_{\sqrt{\Sigma},\mathbf{c}}(S)$ converges, the discrete Gaussian distribution $D_{S,\mathbf{c},\sqrt{\Sigma}}$ is defined over $S$ by its probability distribution function:

$$D_{S,\sqrt{\Sigma},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x})}{\rho_{\sqrt{\Sigma},\mathbf{c}}(S)}. \tag{3}$$

We may also work with *continuous* Gaussians. Given $\sigma \in \mathcal{K}^{++}$, we note $\mathcal{N}_{\mathcal{K},\sigma}$ the unique distribution over $\mathcal{K}$ which probability distribution function is proportional to $\rho_\sigma(x)$. When $\sigma = 1$, we may omit it from the subscript. We note that $\sigma_2 \cdot \mathcal{N}_{\mathcal{K},\sigma_1} \sim \mathcal{N}_{\mathcal{K},\sigma_1 \cdot \sigma_2}$.

### 2.5 Masking

Given a finite field $\mathbb{F}$, masking a value $x \in \mathbb{F}$ consists of splitting it as:

$$x = \sum_{j \in [d]} x_j \tag{4}$$

We say that $(x_j)_{j \in [d]}$ is a valid $d$-sharing of $x$, and note it $[\![x]\!]$, if $x$ and $(x_j)_{j \in [d]}$ satisfy (4). Note that for all $x \in \mathbb{F}$, there exist $|\mathbb{F}|^{d-1}$ valid $d$-sharings of $x$. We also note $\mathsf{Decode}$ the algorithm that maps a valid sharing $(x_j)_{j \in [d]} \in \mathbb{F}^d$ of $x$ to the plain value $x = \sum_{j \in [d]} x_j$.

The $t$-probing model stipulates that for each execution of an algorithm $\mathsf{Alg}$, the adversary can select $t$ intermediate variables $(v_i)_{i \in [t]}$ inside $\mathsf{Alg}$ and is able to learn the values of $(v_i)_{i \in [t]}$ during this execution. Masked security proofs for Mitaka are realized in the $t$-probing model, using a modular proof framework which we informally refer to the SNI (strong non-interference) model. For the purpose of this paper, it suffices to focus on one notion of the SNI framework called $t$-NIo, which we recall in Definition 1.

**Definition 1 ($t$-NIo, [BBE+18]).** *A masked algorithm (gadget) with public outputs $X$ is $t$-NIo (Non-Interfering with public outputs) if and only if every set of at most $t$ intermediate variables can be perfectly simulated with the public outputs and at most $t$ shares of each input.*

## 3 Description of Mitaka

The GPV framework [GPV08] proposed a blueprint for obtaining lattice-based signatures in the hash-then-sign paradigm. Mitaka instantiates the GPV framework with NTRU lattices.

### 3.1 Private and public keys

In Mitaka, the private key is a structured matrix:

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_0 \ \mathbf{b}_1 \end{bmatrix} = \begin{bmatrix} f & F \\ g & G \end{bmatrix} \tag{5}$$

where $f, g, F, G \in \mathcal{R}$ satisfy the NTRU equation in $\mathcal{R}$:

$$f \cdot G - g \cdot F = q \tag{6}$$

Concretely, this quadruple can be generated by first sampling $\mathbf{b}_0 = \begin{bmatrix} f \\ g \end{bmatrix}$, then resolving (6), which can be done efficiently [PP19]. For security, $f, g, F, G$ are required to have small coefficients.

The public key is $h = g \cdot f^{-1} \bmod q$. If we note $\mathbf{A} = \begin{bmatrix} -h & 1 \end{bmatrix}$, we can see that $\mathbf{A} \cdot \mathbf{B} = 0 \bmod q$.

### 3.2 Signing procedure

Algorithm 1 describes the signing procedure of Mitaka. In practice, only the first half $s_1$ of the short vector $\mathbf{s} = (s_1, s_2)$ is actually output by Algorithm 1. However, $s_2$ can be re-computed from a valid signature, so we can assume without loss of generality that $\mathbf{s}$ is output entirely.

**Algorithm 1** $\mathsf{Signing}(\mathsf{sk}, \mathsf{msg}) \to \mathsf{sig}$

**Require:** A message $\mathsf{msg}$, a signing key $\mathsf{sk}$, a bound $\gamma$
**Ensure:** A signature $\mathsf{sig}$ of $\mathsf{msg}$ under $\mathsf{sk}$

1: **repeat**
2: $\quad \mathsf{salt} \leftarrow \{0,1\}^k$
3: $\quad \mathbf{c} := (0, H(\mathsf{salt} \| \mathsf{msg}))$
4: $\quad \mathbf{v} \leftarrow \mathsf{HybridSampler}(\mathsf{sk}, \mathbf{c})$ ▷ Algorithm 2
5: $\quad \mathbf{s} := \mathbf{c} - \mathbf{v}$ ▷ By construction, $\mathbf{s}$ is short
6: **until** $\|\mathbf{s}\| \leq \gamma$
7: **return** $\mathsf{sig} := (\mathsf{salt}, \mathbf{s})$

Algorithm 2 ($\mathsf{HybridSampler}$) is at the core of the signing procedure. Given a target vector $\mathbf{c}$ and a short basis $\mathbf{B}$ of a lattice $\mathcal{L}$, it outputs a lattice point $\mathbf{v} \in \mathcal{L}$ close to $\mathbf{c}$.

Algorithm 2 is designed so that $\mathbf{v}$ is distributed statistically close to $D_{\mathcal{L},\mathbf{c},\sigma}$. This ensures that $\mathbf{v}$ leaks no information about the short basis $\mathbf{B}$. In order to achieve this, continuous Gaussians (Line 3) and discrete Gaussians (Line 4) are employed in a careful manner. Our attack will learn some intermediate variables such that, conditioned on the values of these variables , the distribution of $\mathbf{v}$ is no longer independent of $\mathbf{B}$.

**Algorithm 2** $\mathsf{HybridSampler}(\mathbf{B}, r, \mathbf{c}) \to \mathbf{v}$

**Require:** A target center $\mathbf{c} \in \mathcal{K}^2$, a matrix $\mathbf{B} = [\mathbf{b}_0, \mathbf{b}_1]$
**Precompute:** The Gram-Schmidt orthogonalization $\tilde{\mathbf{B}} = [\tilde{\mathbf{b}}_0, \tilde{\mathbf{b}}_1]$ of $\mathbf{B}$. Standard deviations $\sigma_i = \sqrt{\frac{\sigma^2}{\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle_{\mathcal{K}}} - r^2} \in \mathcal{K}^{++}$ for $i \in \{0, 1\}$ and a fixed parameter $\sigma$
**Ensure:** $\mathbf{v} \overset{s}{\sim} D_{\mathcal{L}(\mathbf{B}),\mathbf{c},\sigma}$
1: $(\mathbf{c}_2, \mathbf{v}_2) := (\mathbf{c}, \mathbf{0})$
2: **for** $i \in \{1, 0\}$ **do**
3: $\quad d_i \leftarrow \frac{\langle \tilde{\mathbf{b}}_i, \mathbf{c}_{i+1} \rangle_{\mathcal{K}}}{\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle_{\mathcal{K}}} - \sigma_i \cdot \mathcal{N}_{\mathcal{K}}$
4: $\quad z_i \leftarrow D_{\mathbb{Z}, d_i, r}$ ▷ When masked, use Algorithm 3
5: $\quad (\mathbf{c}_i, \mathbf{v}_i) := (\mathbf{c}_{i+1}, \mathbf{v}_{i+1}) + z_i \cdot (-\mathbf{b}_i, \mathbf{b}_i)$
6: **end for**
7: **return** $\mathbf{v}_0$

In a masked setting, the masked generation of $[\![z_i]\!]$ from $[\![d_i]\!]$ in Line 4 of Algorithm 2 is performed by Algorithm 3 ($\mathsf{GaussShareByShare}$). Whereas a generic approach would perform this step by leveraging costly A2B and B2A conversions, Algorithm 3 foregoes this approach in favor of a more efficient one, by sampling each share of $[\![z_i]\!]$ independently and in parallel.

**Algorithm 3** $\mathsf{GaussShareByShare}([\![c]\!], r) \to [\![z]\!]$

**Require:** A standard deviation $r$, an arithmetic masking $[\![c]\!]$ for $c \in \frac{1}{C} \cdot \mathbb{Z}$, $B = \left\lceil \sqrt{2d} \right\rfloor$.
**Ensure:** An arithmetic masking $[\![z]\!]$, where $z \overset{s}{\sim} D_{\mathbb{Z},c,r}$
1: **repeat**
2: $\quad$ **for** $j \in [d]$ **do**
3: $\quad\quad z_j \leftarrow D_{\frac{1}{B} \cdot \mathbb{Z}, c_j, \frac{r}{\sqrt{d}}}$
4: $\quad$ **end for**
5: $\quad \mathsf{acc} := \mathsf{Decode}\left((z_j \bmod 1)_{j \in [d]}\right)$
6: **until** $\mathsf{acc} = 0$
7: **return** $[\![z]\!] := (z_j)_{j \in [d]}$

### 3.3 The proof outline of Mitaka and its flaw

We refer to [EFG+22] for the full security proof of Mitaka, which is quite extensive due to the constraints of the $t$-probing model. The relevant part for us is [EFG+22, Lemma 3], which claims that Algorithm 3 is $t$-NIo (Definition 1). While no formal proof for [EFG+22, Lemma 3] is given, [EFG+22] informally argues that it follows from the fact that the input $[\![c]\!] = (c_i)_{i\in[d]}$ is uniform and each share is processed independently and in parallel. We illustrate this reasoning in Fig. 2; any subset $(c_i)_{i\in S}$ is perfectly uniform as long as $|S| < d$, and similarly for $(z_i)_{i\in S}$.

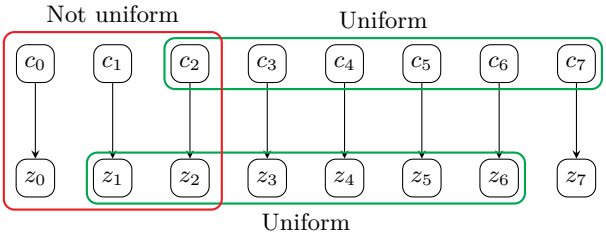



Fig. 2: Illustrating Algorithm 3 (Lines 2 to 4). Probing exclusively the input or output values yields a perfectly uniform subset (Green), but probing them conjointly does not (Red).

Unfortunately, there is a flaw in this reasoning: while it is true that any set of $t < d$ shares of $[\![c]\!]$ or $[\![z]\!]$ would look uniform, the joint distribution of any subset of input values $(c_i)_{i\in S}$ and the corresponding output values $(z_i)_{i\in S}$ is not uniform. Indeed, for any $j \in [d]$, $c_j - z_j$ follows a Gaussian distribution. Moreover, we show in the next section that the observed value of this Gaussian is statistically correlated to the private key, and turn this observation into an attack.

## 4 Our attack

At its heart, our attack is a simple statistical, averaging-based attack.

In Section 4.1, we show in that by probing the appropriate values in Algorithm 3, we are able, for each execution $i$ of Algorithm 1, to compute a scalar $w_i \in \mathbb{R}$ such that $w_i \cdot \mathbf{b}_0$ correlates positively with the signature vector $\mathbf{s}_i$.

Once sufficiently many pairs $(\mathbf{s}_i, w_i)_i$ are collected, we show in Section 4.2 how we can compute a noisy estimator of $\mathbf{b}_0$, then recover $\mathbf{b}_0$ exactly via lattice-reduction (Section 5.3), pure rounding (Section 5.2), or guessing plus linear algebra (Section 5.4), depending on the regime.

### 4.1 Placing the probes

Suppose Algorithm 2 is used to sample $\mathbf{v} \overset{s}{\sim} D_{\mathcal{L}(\mathbf{B}),\mathbf{c},\sigma}$. Let us note $\mathbf{v} = \mathbf{B} \cdot \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \end{bmatrix}$, and let $c \in \mathbb{R}$ the first coefficient of $\mathbf{z}_0$. We target the execution of Algorithm 3 when it is used in Algorithm 2 with $[\![c]\!]$ as input. We target $c$ in particular because the signature $\mathbf{s}$ contains $c \cdot \mathbf{b}_0$ as an additive term, so learning $\mathbf{s}$ plus information about $c$ provides information about $\mathbf{b}_0$. As illustrated in Fig. 2 (Red), during that specific execution of Algorithm 3, we probe:

1. the first $t_1$ coefficients $(c_j)_{j \in [t_1]}$ of $[\![c]\!]$;
2. the first $t_1$ coefficients $(z_j)_{j \in [t_1]}$ of $[\![z]\!]$;
3. the Boolean value $\mathsf{acc}$.

As long as $t := 2 \cdot t_1 + 1 < d$, this is consistent with what is allowed within the $t$-probing model. Note that our choice of probes requires $d \geq 4$. Once $\mathsf{acc} = 0$, we know that $[\![z]\!] = (z_j)_{j \in [d]}$ is output and incorporated in the signature. We compute:

$$w = \sum_{j \in [t_1]} (c_j - z_j) \tag{7}$$

We say that the trace associated to a given execution is $\mathsf{trace} = (\mathbf{s}, w)$, where $\mathbf{s}$ is the vector such that $\mathbf{A} \cdot \mathbf{s} = H(\mathsf{salt} \| \mathsf{msg})$, which is output as part of the signing procedure.

### 4.2 Recovering the signing key

We show how to exploit traces in order to recover the private key $\mathbf{b}_0$. Since the product $w \cdot \mathbf{b}_0$ is an additive component of the signature vector $\mathbf{s}$, there is a slight but exploitable correlation between $\mathbf{s}$ and $w \cdot \mathbf{b}_0$, more precisely the dot product $\langle \mathbf{s}, w \cdot \mathbf{b}_0 \rangle_{\mathbb{R}} = \langle w \cdot \mathbf{s}, \mathbf{b}_0 \rangle_{\mathbb{R}}$ will tend to be slightly larger than zero. We formalize this intuition by computing a real-valued estimator for $\mathbf{b}_0$ from a set of $N$ traces $(\mathsf{trace}_i = (\mathbf{s}_i, w_i))_{i \in [N]}$:

$$\widehat{\mathbf{b}}_0 = \frac{1}{\left(\sum_{i \in [N]} w_i^2\right)} \cdot \left( \sum_{i \in [N]} w_i \cdot \mathbf{s}_i \right). \tag{8}$$

We now study the distribution of signatures, conditioned on additional information. A valid signature $\mathbf{s}$ satisfies $\mathbf{s} \overset{s}{\sim} D_{\mathcal{L}-\mathbf{c},\sigma}$. If we note $V = \mathrm{Span}_{\mathbb{R}}(\mathbf{b}_0)$, we can decompose $\mathbf{s}$ over $V \oplus V^{\perp}$:

$$\mathbf{s} = \bar{\mathbf{s}} + \overset{\perp}{\mathbf{s}}, \qquad \text{where} \qquad \begin{cases} \bar{\mathbf{s}} \overset{s}{\sim} D_{\mathrm{Proj}(\{\mathcal{L}-\mathbf{c}\},V),\sigma} \\ \overset{\perp}{\mathbf{s}} \overset{s}{\sim} D_{\mathrm{Proj}(\{\mathcal{L}-\mathbf{c}\},V^{\perp}),\sigma} \end{cases} \tag{9}$$

Since $\overset{\perp}{\mathbf{s}} \perp \mathbf{b}_0$, the distribution of $\overset{\perp}{\mathbf{s}}$ is independent of $w$. On the other hand, we use the following heuristic for the conditional distribution of $\bar{\mathbf{s}}$:

$$\bar{\mathbf{s}} | w \quad \overset{s}{\sim} \quad w \cdot \mathbf{b}_0 + D_{\mathrm{Proj}(\{\mathcal{L}-\mathbf{c}-w\cdot\mathbf{b}_0\},V),\sigma^*}, \quad \text{where} \quad \sigma^* = \sqrt{\sigma^2 - \frac{t_1}{d} \cdot r^2} \tag{10}$$

Let us note $\mathbf{w} = (w_i)_{i\in[N]}$. Summing the equation above for all traces, we obtain:

$$\sum_{i\in[N]} w_i \cdot \mathbf{s}_i \quad \overset{s}{\sim} \quad \sum_{i\in[N]} w_i \cdot \bar{\mathbf{s}}_i \quad + \quad \sum_{i\in[N]} w_i \cdot \overset{\perp}{\mathbf{s}}_i \tag{11}$$

$$\overset{s}{\sim} \|\mathbf{w}\|^2 \cdot \mathbf{b}_0 \tag{12}$$

$$+ D_{\mathrm{Proj}\left(\left\{\sum_i w_i(\mathcal{L}-\mathbf{c}_i-w_i\cdot\mathbf{b}_0)\right\},V\right),\sigma^*\|\mathbf{w}\|} \tag{13}$$

$$+ D_{\mathrm{Proj}\left(\left\{\sum_i w_i(\mathcal{L}-\mathbf{c}_i)\right\},V^\perp\right),\sigma\cdot\|\mathbf{w}\|} \tag{14}$$

Dividing everything by $\|\mathbf{w}\|^2$ gives the distribution of our estimator $\widehat{\mathbf{b}}_0$:

$$\widehat{\mathbf{b}}_0 \quad \overset{s}{\sim} \quad \mathbf{b}_0 + X, \tag{15}$$

where $X$ is the random variable corresponding to summing (13) and (14), then dividing the result by $\|\mathbf{w}\|^2$. $X$ is subgaussian for the Gaussian parameter $\sigma/\|\mathbf{w}\|$, so we model $X$ in a way that is simpler, more conservative for an attacker, and essentially tight in our context:

$$X \overset{s}{\sim} D_{\frac{1}{\|\mathbf{w}\|^2}\left\{\sum_i w_i(\mathcal{L}-\mathbf{c}_i-w_i\cdot\mathbf{b}_0)\right\},\sigma_X}, \tag{16}$$

where $\sigma_X = \sigma/\|\mathbf{w}\|$. Since we modeled each $w_i$ as a Gaussian of standard deviation $r\sqrt{\frac{t_1}{d}}$, $\|\mathbf{w}\|^2$ is a $\chi^2$ distribution with $N$ degrees of freedom, scaled by a factor $\frac{r^2\cdot t_1}{d}$. This implies that with probability $\Omega(1)$:

$$\sigma_X \leq \sigma \cdot \sqrt{\frac{d}{r^2 \cdot t_1 \cdot N}} \tag{17}$$

For a continuous $2n$-dimensional Gaussian $Z$ of parameter $\sigma_X$, the probability that $\|Z\|_\infty \leq t$ is lower bounded as follows:

$$\mathbb{P}[\|Z\|_\infty \leq t] \geq \left(1 - 2e^{-t^2/2\sigma_X^2}\right)^{2n} \tag{18}$$

While $X$ is discretized, we assume for the rest of our analysis that it behaves like a continuous Gaussian: $X \sim \mathcal{N}_{\mathbb{R}^{2n},\sigma_X}$. In this case, (18) guarantees that $\|X\|_\infty \leq 1/2$ with probability $\geq 1/2$ if:

$$\sigma_X \leq \frac{1}{\sqrt{8 \cdot \log_2(4 \cdot n)}}. \tag{19}$$

Combining (17) with (19) gives the following success condition:

$$N \;\geq\; \frac{8 \cdot \log_2(4 \cdot n) \cdot d \cdot \sigma^2}{t_1 \cdot r^2} \tag{20}$$

If (20) is satisfied, then with good probability $\lfloor \widehat{\mathbf{b}}_0 \rceil = \mathbf{b}_0$ and we can recover $\mathbf{b}_0$. The second private basis vector $\mathbf{b}_1$ can be recovered by solving (6).

Note that (20) indicates that even if the masking order $d$ is polynomially high and the number of probes per execution $t = 2 \cdot t_1 + 1$ is constant, a polynomial number of traces $N$ suffices to ensure key recovery with $\Omega(1)$ probability.

## 5 Concrete results

We tested the viability of our attack via experiments. To the best of our knowledge, there is no masked implementation of Mitaka, including private ones. We instead rely on an *unmasked* C implementation [Esp22] of Mitaka.

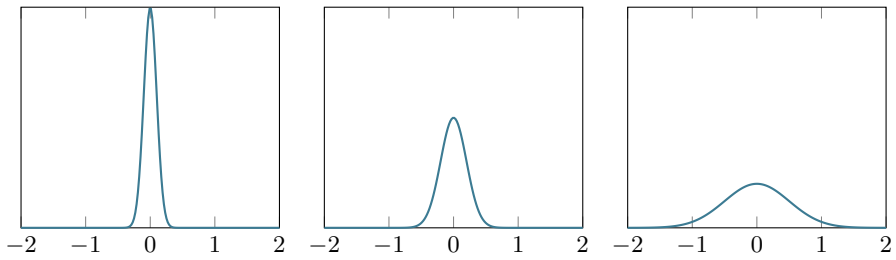
### 5.1 Simulating the leakage

The implementation of [Esp22] does not use Algorithm 3 to sample $[\![z_i]\!]$. Instead, it directly samples $z_i \leftarrow D_{\mathbb{Z},d_i,r}$. We can nevertheless simulate the computation of the value $w$. Let $X, Y$ be two independent Gaussians of center 0 and standard deviation $\sigma_X, \sigma_Y$. Given the sum $Z = X+Y$, it is well-known that the conditional distribution of $X$ given the realization $Z = z$ is distributed as a Gaussian of mean $z \cdot \frac{\sigma_X^2}{\sigma_X^2+\sigma_Y^2}$ and variance $\frac{\sigma_X^2 \cdot \sigma_Y^2}{\sigma_X^2+\sigma_Y^2}$. This provides a simple way to simulate the computation of $w$ in Algorithm 3:

1. Sample $z \leftarrow D_{\mathbb{Z},d,r}$ corresponding to the $[\![z]\!]$ output by Algorithm 3. This sample is easily obtained from the C implementation.
2. Compute $w = (c - z) \cdot \frac{t_1}{d} + r\sqrt{\frac{t_1(d-t_1)}{d^2}} \cdot \mathcal{N}_{\mathbb{R}}$.

One subtlety that this simulation does not capture is that each share of $z$ belongs to $\frac{1}{B} \cdot \mathbb{Z}$, whereas our simulated $w$ is not discretized in any way. This seems unimportant as the discretization (or lack thereof) of $w$ does not seem to have an influence on the feasibility of our attack.

With this method of simulating the computation of $w$, we can now compute our estimator $\widehat{\mathbf{b}_0}$ using (8). Following (15), the difference $\widehat{\mathbf{b}_0} - \mathbf{b}_0$ follows an isotropic continuous Gaussian distribution $X$ of standard deviation $\sigma_X$ given by (17). We distinguish three regimes for $X$: low-, moderate- and high-noise, see Fig. 3. We cover each regime in a distinct section (Sections 5.2 to 5.4), since we employ (slightly) different strategies for each setting.




(a) Low-noise (Section 5.2)

(b) Moderate-noise (Section 5.4)

(c) High-noise (Section 5.3)

Fig. 3: Three regimes for the coefficient-wise distribution of $(\widehat{\mathbf{b}_0} - \mathbf{b}_0)$

### 5.2 Low-noise regime: $\|\widehat{\mathbf{b_0}} - \mathbf{b_0}\|_\infty < 1/2$

Following our analysis in Section 4.2, satisfying (20) guarantees with probability $\Omega(1)$ that we fall into this regime. In this case, $\lfloor \widehat{\mathbf{b}_0} \rceil = \mathbf{b}_0$.

Although the required number of samples $N$ is polynomial, this number may be moderately large in practice. In our experiments, we found that for MITAKA-512 and for $\frac{t_1}{d} \approx \frac{1}{2}$, setting $N \approx 2^{22}$ provides a good chance of success.

### 5.3 High-noise regime: $\|\widehat{\mathbf{b_0}} - \mathbf{b_0}\| < \sqrt{q}$

If $N$ is large but does not satisfy (20), we can still recover $\mathbf{b}_0$ from the estimator $\widehat{\mathbf{b}_0}$ via lattice reduction methods. We first recall the Gaussian Heuristic.

**Definition 2.** *Let $gh(\mathcal{L})$ be the expected first minimum of a lattice $\mathcal{L}$ according to the Gaussian Heuristic. For a lattice $\mathcal{L} \subset \mathbb{R}^m$ generated by a full-rank matrix full-rank $\mathbf{M} \in \mathbb{R}^{m \times m}$, it is given by:*

$$gh(\mathcal{L}) = \sqrt{\frac{m}{2\pi e}} \cdot \det(\mathbf{M})^{1/m}. \tag{21}$$

Consider the rounded estimator: $\breve{\mathbf{b}}_0 = \lfloor \widehat{\mathbf{b}_0} \rceil \in \mathcal{R}^2$. If we note $\mathbf{e} = \breve{\mathbf{b}}_0 - \mathbf{b}_0$, it holds that $\|\mathbf{e}\|^2 \leq \|\widehat{\mathbf{b}_0} - \mathbf{b}_0\|^2 + n^2/2$. On the other hand, in the parameter regime of MITAKA, $\mathbf{b}_0$ is the shortest vector in the NTRU lattice: $\|\mathbf{b}_0\| \approx 2\sqrt{q}$.[1] Since $\|\widehat{\mathbf{b}_0} - \mathbf{b}_0\| < \sqrt{q}$, we can expect $\mathbf{e}$ to be much shorter than the shortest vector in the NTRU lattice. This allows us to use Kannan's embedding; a good reference for this technique is [AGVW17], which methodology we follow here. We first generate the matrix $\mathbf{M}$:

$$\mathbf{M} = \left[\begin{array}{cc|c} \mathbf{I}_n & & \breve{\mathbf{b}}_0 \\ \mathbf{H} & q\mathbf{I}_n & \\ \hline & & 1 \end{array}\right] \in \mathbb{Z}^{d \times d} \tag{22}$$

where $d = 2n + 1$ and $\mathbf{H} = \mathcal{A}(h)$. By construction, we expect $\begin{bmatrix} \mathbf{e} \\ 1 \end{bmatrix}$ to be the shortest vector of $\mathbf{M}$. Therefore, we apply the BKZ lattice reduction algorithm to $\mathbf{M}$ with blocksize $\beta$ in order to recover $\mathbf{e}$. Under the geometric series assumption, $\mathbf{e}$ can be found if:

$$\sqrt{\frac{\beta}{d}} \cdot \sqrt{\|\mathbf{e}\|^2 + 1} \leq \delta_\beta^{2\cdot\beta - d} \cdot \det(\mathbf{M})^{1/d}, \tag{23}$$

where $\delta_\beta = \left(\frac{(\pi\beta)^{1/\beta} \cdot \beta}{2\pi e}\right)^{1/(2(\beta-1))}$ [Che13, Eq. (4.2)]. The corresponding core-SVP hardness $\lambda$ for our key-recovery attack can be determined by computing $\lambda = \lfloor 0.292 \cdot \beta \rfloor$ for the minimal value of $\beta$ such that (23) is satisfied. Alternatively, one may also use the nearest-colattice algorithm of [EK20].

[1] In [EFG+22], it is shown that $\|\mathbf{b}_0\| \leq \alpha\sqrt{q}$, with $\alpha \approx 2.04$ for MITAKA-512 and $\alpha \approx 2.33$ for MITAKA-1024.

### 5.4 Moderate-noise regime: $\|\widehat{\mathbf{b_0}} - \mathbf{b_0}\|_\infty < 1$

If it is the case that:

$$\frac{1}{2} \;<\; \|\widehat{\mathbf{b}_0} - \mathbf{b}_0\|_\infty <\; 1,$$

then we are in an paradoxical situation: $\widehat{\mathbf{b}_0}$ is very close to $\mathbf{b}_0$, but rounding its coefficients will return a different vector from $\mathbf{b}_0$. Worse, several dozens of coefficients may be erroneous, and an exhaustive search of these coefficients may be expensive in practice. Similarly, lattice reduction as in Section 5.3 may be expensive. We now describe a simple trick that allows to recover $\mathbf{b}_0$ with high probability and little to no computation effort.

*Observation 1.* By construction, when interpreted as a vector in $\mathbb{Z}^{2n}$, $\mathbf{b}_0$ satisfies:

$$\mathbf{A} \cdot \mathbf{b}_0 = \mathbf{0} \bmod q, \tag{24}$$

Recall that $\widehat{\mathbf{b}_0}$ is equal to $\mathbf{b}_0$ plus Gaussian noise of standard deviation $\sigma_X$. The fact $\|\widehat{\mathbf{b}_0} - \mathbf{b}_0\|_\infty < 1$ implies that $\sigma_X$ is small (concretely, $\sigma_X \leq 0.25$ if $n = 512$). This in turn implies that errors close to 1 in absolute value are likely to be rare, which leads to our first key observation:

> If a coefficient of $\widehat{\mathbf{b}_0}$ is close to an integer, then the corresponding coefficient of $\mathbf{b}_0$ is highly likely to be equal to this integer.

*Observation 2.* We now observe that recovering half of the coefficients of $\mathbf{b}_0$ is sufficient to recover it entirely. Suppose we have guessed $n$ of the $2{\cdot}n$ entries of $\mathbf{b}_0$. We can rearrange the entries of $\mathbf{b}_0$ as $\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$, where $\mathbf{y}$ corresponds to coefficients of $\mathbf{b}_0$ that were successfully guessed, and $\mathbf{x}$ are the remaining ones. By rearranging the columns of $\mathbf{A}$ in the same way, (24) becomes:

$$\begin{bmatrix}\mathbf{A}_1 \; \mathbf{A}_2\end{bmatrix} \cdot \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathbf{0} \bmod q, \tag{25}$$

If $\mathbf{A}_1$ is invertible, then we can recover $\mathbf{x}$ by computing $\mathbf{x} = -\mathbf{A}_1^{-1} \cdot \mathbf{A}_2 \cdot \mathbf{y}$. In practice, we observe that $\mathbf{A}_1$ is invertible more often than not.

*Example 1.* We illustrate this strategy with a toy example over $\mathbb{Z}_q$, $q = 19$. Let:

$$\mathbf{A} = \begin{bmatrix}12 & 9 & 10 & 5\\ 9 & 7 & 5 & 15\end{bmatrix} \in \mathbb{Z}_q^{2\times 4} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix}7 & 6 & 1 & 18\end{bmatrix}^t.$$

One can check that $\mathbf{A}{\cdot}\mathbf{b} = \mathbf{0} \bmod q$. Our estimator will be a noisy version of $\mathbf{b}$, for example $\widehat{\mathbf{b}} = \begin{bmatrix}7.1 & 6.4 & 1.6 & 18.1\end{bmatrix}^t$. Naively rounding $\widehat{\mathbf{b}}$ gives $\lfloor\widehat{\mathbf{b}}\rceil = \begin{bmatrix}7 & 6 & 2 & 18\end{bmatrix}^t \neq \mathbf{b}$. In contrast, guessing half of the coefficients (precisely, the half which are closer to an integer) gives $\mathbf{b} = \begin{bmatrix}7 & * & * & 18\end{bmatrix}^t$, and which point the remaining half of $\mathbf{b}$ can be computed by solving the linear system $\mathbf{A} \cdot \mathbf{b} = \mathbf{0} \bmod q$.

*Success probability.* Recall that $X = \widehat{\mathbf{b}_0} - \mathbf{b}_0$. This attack succeeds if there exists $\epsilon > 0$ such that, with probability $\Omega(1)$:

1. No coefficient of $X$ is larger in absolute norm than $1 - \epsilon$.
2. At least half of the coefficients of $X$ are in $[-\epsilon, \epsilon]$;

Item 1 ensures that "guessing to the nearest integer" all coefficient of $\widehat{\mathbf{b}_0}$ that are $\epsilon$-close to an integer will indeed return the correct coefficient of $\mathbf{b}_0$, and is true if $\mathbb{P}[\|X\|_\infty < 1 - \epsilon] \leq 1/2$. Item 2 ensures there are $n$ such coefficients. Following our modelization of $X$ as a $(2 \cdot n)$-dimensional Gaussian of standard deviation $\sigma_X$, the conditions above can be expressed, using Gaussian tail bounds, as:

$$\left(1 - 2 \cdot e^{-\frac{(1-\epsilon)^2}{2 \cdot \sigma_X^2}}\right)^{2n} < \frac{1}{2}, \tag{26}$$

$$\text{where} \quad \epsilon = \min\left\{\epsilon^* \mid \mathcal{N}_{\mathbb{R},\sigma_X}([-\epsilon^*, \epsilon^*]) \geq \frac{1}{2}\right\}. \tag{27}$$

For $n = 512$, our attack is effective when $\sigma_X \lesssim 0.214$. In contrast, for this value of $\sigma_X$, pure rounding (Section 5.2) succeeds with probability[2] $\leq 2^{-29}$. Similarly, we expect on average 21 coefficients of $\widehat{\mathbf{b}_0}$ to round incorrectly, so that a pure lattice reduction approach (Section 5.3) would require a blocksize $\beta = 196$ and be costly to carry out. In comparison, our guessing-based approach is inexpensive and succeeds with high probability. Concretely, it allows us to decrease $N$ to $2^{21}$.

*Refinement.* This "smart guessing" technique can be refined to remain effective even if we guess less than half of the coefficients of $\mathbf{b}_0$. Suppose that with probability $1/2$, we can guess $k$ of the $2n$ coefficients of $\mathbf{b}_0$. This is the case if (26) is satisfied, and by replacing (27) by this relaxed condition:

$$\epsilon = \min\left\{\epsilon^* \mid \mathcal{N}_{\mathbb{R},\sigma_X}([-\epsilon^*, \epsilon^*]) \geq \frac{k}{2n}\right\}. \tag{28}$$

We then rewrite (24) as (25), except that now $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times (2n-k)}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{n \times k}$. We put $\mathbf{A}_1$ in systematic form ($\mathbf{M} \times \mathbf{A}_1 = \begin{bmatrix}\mathbf{I}_n & \bar{\mathbf{A}}_1\end{bmatrix}$) so that (25) becomes:

$$\bar{\mathbf{A}}_1 \cdot \mathbf{x}_2 + \mathbf{x}_1 = \mathbf{z}, \quad \text{where} \quad \begin{cases}\mathbf{z} = -\mathbf{M} \cdot \mathbf{A}_2 \cdot \mathbf{y} \in \mathbb{Z}_q^n \\ (\mathbf{x}_2, \mathbf{x}_1) \in \mathbb{Z}^{n-k} \times \mathbb{Z}^n\end{cases} \tag{29}$$

(29) can be interpreted as an LWE problem with a secret of dimension $n - k$, which indeed becomes vacuous when $k = n$. Unfortunately, due to Gaussian cut-off effects, this optimization does not seem to significantly increase the range of $\sigma_X$ covered by our technique. We still provide it here for reference.

[2] Alternatively, (19) implies that pure rounding requires $\sigma_X \lesssim 0.1066$ to be practical. Hence it is applicable on a more narrow range than our guessing-based approach.

*Remark 1.* We expect the guessing trick to also apply to a recent power analysis attack on Falcon by Guerreau et al. [GMRR22]. Similarly to our attack, their attack recovers a noisy estimator of $\mathbf{b}_0$, where the noise decreases with the number of traces. It then recovers $\mathbf{b}_0$ either by rounding (as in Section 5.2) or via lattice reduction (as in Section 5.3). Our guessing-based approach is applicable in regimes that are out of reach for pure rounding, but for which the cost of lattice reduction remains prohibitive.

*Remark 2.* After the initial completion of this work, we realized that a similar guessing trick was described in [DDGR20, §6.1], although with a different perspective (the "*LWE with side information*" framework). We invite the interested reader to read [DDGR20, §6.1] for a complementary point of view.

## 6 Conclusion

We have proposed a key-recovery attack against Mitaka in the $t$-probing model. Given a masked implementation of Mitaka with $d \geq 4$ shares, an attacker with the capability of probing $t < d$ variables per execution can recover the private key efficiently with $N = 2^{21}$ executions of the signing algorithm. More generally, our attack can be carried as long as $N = \Omega(d/t)$.

As part of our attack, we proposed a guessing-based trick which significantly reduces the computational cost of our attack for many relevant regimes.

## Acknowledgements

I would like to thank Mélissa Rossi, Thomas Espitau, Alexandre Wallet, Morgane Guerreau and Eamonn Postlethwaite for useful discussions about [EFG+22], [GMRR22], and the attack presented in this paper. I am particularly grateful to my PQShield colleagues Rafaël del Pino and Fabrice Mouhartem for discussing the subtleties of lattice attacks with me. Finally, I would like to thank the anonymous reviewers of PKC 2023 for their insightful comments.

## References

AA04. Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *2004 IEEE Symposium on Security and Privacy*, pages 3–11. IEEE Computer Society Press, May 2004.

AGVW17. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 297–322. Springer, Heidelberg, December 2017.

BBE+18. Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi. Masking the GLP lattice-based signature scheme at any order. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 354–384. Springer, Heidelberg, April / May 2018.

Che13. Yuanmi Chen. *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe*. PhD thesis, 2013. https://archive.org/details/PhDChen13.

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. Cryptology ePrint Archive, Report 2020/292, 2020. https://eprint.iacr.org/2020/292.

DFS19. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. *Journal of Cryptology*, 32(4):1263–1297, October 2019.

EFG+22. Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 222–253. Springer, Heidelberg, May / June 2022.

EK20. Thomas Espitau and Paul Kirchner. The nearest-colattice algorithm. Cryptology ePrint Archive, Report 2020/694, 2020. https://eprint.iacr.org/2020/694.

Esp22. Thomas Espitau. Supporting code for mitaka signature (eurocrypt 2022). GitHub, 2022. https://github.com/espitau/Mitaka-EC22.

GMO01. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer, Heidelberg, May 2001.

GMRR22. Morgane Guerreau, Ange Martinelli, Thomas Ricosset, and Mélissa Rossi. The hidden parallelepiped is back again: Power analysis attacks on falcon. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(3):141–164, Jun. 2022.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

GST14. Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 444–461. Springer, Heidelberg, August 2014.

ISW03. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Heidelberg, August 2003.

KJJ99. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Heidelberg, August 1999.

Koc96. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Heidelberg, August 1996.

LDK+22. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

NIS22. NIST. Nistir 8413 - status report on the third round of the nist post-quantum cryptography standardization process, 2022. https://doi.org/10.6028/NIST.IR.8413.

PFH+22. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

PP19. Thomas Pornin and Thomas Prest. More efficient algorithms for the NTRU key generation using the field norm. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 504–533. Springer, Heidelberg, April 2019.