MITAKA: A Simpler, Parallelizable, Maskable Variant of FALCON

Thomas Espitau[1], Akira Takahashi[2], Mehdi Tibouchi[1], and Alexandre Wallet[3]

[1] NTT Corporation, Japan
{thomas.espitau.ax,mehdi.tibouchi.br}@hco.ntt.co.jp
[2] Aarhus University, Denmark
takahashi@cs.au.dk
[3] Inria, Rennes Bretagne-Atlantique Center, France
alexandre.wallet@inria.fr

Abstract. This work describes the MITAKA signature scheme: a new hash-and-sign signature scheme over NTRU lattices which can be seen as a variant of NIST finalist FALCON. It achieves comparable efficiency but is considerably simpler, online/offline, and easier to parallelize and protect against side-channels, thus offering significant advantages from an implementation standpoint. It is also much more versatile in terms of parameter selection.

We obtain this signature scheme by replacing the FFO lattice Gaussian sampler in FALCON by the "hybrid" sampler of Ducas and Prest, for which we carry out a detailed and corrected security analysis. In principle, such a change can result in a substantial security loss, but we show that this loss can be largely mitigated using new techniques in key generation that allow us to construct much higher quality lattice trapdoors for the hybrid sampler relatively cheaply. This new approach can also be instantiated on a wide variety of base fields, in contrast with FALCON's restriction to power-of-two cyclotomics.

In addition, we describe a provably secure masking of MITAKA at any order at much lower cost than previous masking techniques for Gaussian sampling-based signature schemes, for cheap and dependable side-channel protection.

1 Introduction

The third round finalists for signatures in the NIST postquantum standardization process consist of just three candidates: Rainbow [11, 10], a multivariate scheme, Dilithium [13, 32], a lattice-based scheme in the Fiat–Shamir with aborts framework, and FALCON [43], a hash-and-sign signature over NTRU lattices. They occupy fairly different positions within the design space of post-quantum signature schemes, and it is therefore important to understand, for each of them, to what extent they could possibly be improved by exploring similar designs that overcome some of their limitations. This paper aims at doing so for the FALCON signature scheme.

Hash-and-sign lattice-based signatures. FALCON fits within the long and hectic history of hash-and-sign signatures based on lattices. In those schemes, the signing key is a "good" representation of a lattice, the *trapdoor*, which makes it possible, given an arbitrary point in the ambient space, to find lattice points that are relatively close to it (i.e. solve the *approximate closest vector* problem, ApproxCVP[4]); the verification key, on the other hand, is a "bad" representation: it allows anyone to check if a point is in the lattice, but not to solve ApproxCVP. In order to sign a message, it is then hashed to a random point in the ambient space, and the signature is a lattice point close to it, obtained using the trapdoor. To verify, one checks that the signature is in the lattice and sufficiently close to the hash digest.

Early constructions along those lines, such as the GGH signature scheme [21] and multiple iterations of NTRUSign [23, 22], were later shown to be insecure due to a common critical vulnerability: the lattice points obtained as signatures would leak information about the trapdoor used to compute them, which could then be recovered using more or less advanced statistical techniques [37, 15]. One of the first round NIST candidates was in fact broken using the same idea [44].

[4] Sometimes, this is also seen as a *bounded distance decoding* problem, BDD, but with large enough decoding bound that there are exponentially many solutions, instead of a unique one as is typically the case in the traditional formulation of BDD.

It is thus crucial for security to prove that signatures are sampled according to a distribution that is *statistically independent* of the trapdoor. The first approach to do so, which remains the state of the art,[5] is due to Gentry, Peikert and Vaikuntanathan (GPV) [19]: sample the ApproxCVP solution according to a discrete Gaussian distribution centered at the target point and supported over the lattice, with covariance independent from the trapdoor (usually spherical). This type of lattice discrete Gaussian sampling can be carried out by randomizing known deterministic algorithms for ApproxCVP, like Babai rounding and Babai's nearest plane algorithm. Gaussians play an essential role because of their stable nature: the linear combination of two discrete Gaussian random vectors is still statistically close to a discrete Gaussian (at least for standard deviations above the so-called smoothing parameter).

Within the overall GPV framework, specific signature schemes vary according to the lattices over which they are instantiated, the construction of the corresponding trapdoors, and the lattice Gaussian sampling algorithms they rely on based on those trapdoors. The security level achieved by such a scheme is then essentially determined by the *quality* of the trapdoor and of the Gaussian sampling algorithm, defined as the minimal standard deviation achievable in Gaussian sampling, while still preserving the statistical independence of the output.

A complete overview of existing proposals for each of those elements is beyond the scope of the current work. We focus instead on the particular case of NTRU lattices with the usual NTRU trapdoors first considered in NTRUSign, as those lattices appear to offer the most efficient implementations by a significant margin, thanks to their compact trapdoors.

Hash-and-sign signatures over NTRU lattices. NTRU lattices are, in essence, free rank 2 module lattices over cyclotomic rings, and the NTRU designers showed how to construct good trapdoors for them, even though the original signature schemes based on them proved insecure.

They were brought within the GPV framework (and thus gained provable security) thanks to the work of Ducas, Lyubashevsky and Prest (DLP) [14], who combined them with the Gaussian sampling algorithm obtained by randomizing Babai's nearest plane algorithm (this randomization is sometimes called the *Klein sampler* for lattice Gaussians). They analyzed the security of the construction and provided what became the first reasonably efficient implementation of a signature scheme in the GPV framework.

This DLP signature scheme offers relatively compact keys and signatures, but suffers from a relatively long signing time, quadratic in the $\mathbb{Z}$-rank of the underlying lattice. This is because the nearest plane computation is carried out after descending to $\mathbb{Z}$, essentially ignoring the module structure of the lattice.

Falcon is a direct improvement of this scheme, obtained by replacing this quadratic Gaussian sampling by a quasilinear one, derived from the quasilinear nearest plane algorithm described in the Fast Fourier Orthogonalization paper of Ducas and Prest [16] (and refining the parameter selection using a tighter statistical analysis based on the Rényi divergence). The computation still ultimately descends to $\mathbb{Z}$, but takes advantage of the tower field structure of the underlying number field (assumed to be a power-of-two cyclotomic) to achieve a better complexity.

These two approaches are equivalent in terms of the quality of the resulting Gaussian sampler, which is essentially the best possible for the given NTRU lattice. However, DLP does so at the cost of a slow signing algorithm, whereas Falcon, while fast, suffers from a very complex signing algorithm that is hard to implement, poorly suited for parallelization and difficult to protect against side-channel attacks. On the last point, both schemes have been shown to suffer from potential vulnerabilities with respect to side-channel leakage [18], and even though the most recent implementation of Falcon appears to be protected against timing attacks [39, 24], countermeasures against stronger side-channel attacks like DPA seem difficult to achieve. Falcon is also limited to NTRU lattices over power-of-two cyclotomic fields, which limits its flexibility in terms of parameter selection. That latter limitation can be overcome to some extent by extending the construction to higher rank modules, as done in ModFalcon [9], but the other drawbacks remain.

Another possibility is to instantiate the randomized ApproxCVP algorithm directly over the underlying ring, instead of doing so over $\mathbb{Z}$. For the randomized version of Babai rounding, this gives rise to (the ring version of) Peikert's sampler, as introduced in [38]. This can also be done for Babai's nearest plane algorithm, leading to what Ducas and Prest call the *hybrid sampler*. The resulting algorithms consist of a constant number of ring multiplications, so that quasilinear complexity is obtained "for free" as long as the

[5] Other techniques have been proposed that avoid Gaussian distributions, as in [33], but they tend not to be competitive.

underlying ring has a fast multiplication algorithm (as certainly holds for arbitrary cyclotomics). This makes them highly versatile in terms of parameter selection. They are also much simpler than FALCON, easy to parallelize, and support fairly inexpensive masking for side-channel protection.

Their downside, however, is the lower quality of the corresponding samplers compared to FALCON and DLP. Indeed, by not descending to $\mathbb{Z}$ but only to the ring itself, the ApproxCVP algorithm achieves less tight of a bound compared to the Klein sampler, and hence the Gaussian sampling has a larger standard deviation. This is analyzed in details in Prest's Ph.D. thesis [41] (although certain heuristic assumptions are incorrect), and results in a substantially lower security level than FALCON and DLP.

Our contributions: the MITAKA *signature scheme.* In this work, we revisit in particular the hybrid sampler mentioned above, and show that the security loss compared to FALCON can be largely mitigated using a novel technique to generate higher quality trapdoors. The resulting scheme, MITAKA,[6] offers an attractive alternative to FALCON in many settings since:

- it is considerably simpler from an algorithmic and an implementation standpoint, while keeping the same complexity (in fact, it is likely faster at the same dimension due to better cache locality);
- signature generation is almost embarrassingly parallel(izable);
- like Peikert's sampler, it has an online/offline structure, with the online part requiring only one-dimensional discrete Gaussian sampling with very small, constant standard deviation and simple linear operations;
- it can be instantiated over arbitrary cyclotomic fields[7], which makes it quite versatile in terms of parameter selection;
- it is easier to protect against side-channels and can be cheaply masked even at high order, at the cost of a small loss in security depending on the masking order.

The main idea that allows us to achieve higher security than previously expected is as follows. It is well-known that, given NTRU generators (f, g), it is easy to compute the quality of the corresponding NTRU trapdoor for the hybrid sampler (in particular, it can be done without computing the whole trapdoor). It is thus very easy to check whether a given (f, g) reaches a certain threshold in terms of bit security, and as a result, the costly part of key generation is the sampling of the random ring elements f and g themselves (with discrete Gaussian coefficients). One can therefore achieve a greatly improved security level at the same cost in terms of randomness and not much more computation time if one can "recycle" already sampled ring elements f and g.

We propose several ways of doing so. The simplest one is to generate lists $\{f_i\}$, $\{g_j\}$ of candidate elements for f and g, and test the pairs (f_i, g_j): this increases the space of candidates quadratically, instead of linearly, in the amount of generated randomness. One can also generate the f_i's, g_j's themselves as sums of Gaussians with smaller standard deviation (as long as it remains above the smoothing parameters), and consider the Galois conjugates of a given g_j. By combining these techniques appropriately, we achieve a substantial security increase, of around 15 bits for typical parameter sizes. Concretely, we achieve the same security level as Dilithium–II [13] (which was argued to reach NIST Level–I) in dimension $d = 512$, and attain roughly NIST Level–V in dimension $d = 1024$, with intermediate parameter settings possible.

We also provide a detailed security analysis of our construction, and while most of the presentation focuses on power-of-two cyclotomics for simplicity's sake and easier comparison with previous work, we also show that intermediate NIST security levels can be conveniently achieved using other base fields, e.g. of dimension 648 (same security as FALCON–512), 768 (NIST Level–II), 864 (NIST Level–III) and 972 (NIST Level–IV).

Finally, we introduce a new, concrete approach to mask the signature generation algorithms efficiently. In previous work, efficiently masking signature schemes using Gaussian sampling has proved quite challenging: even for the case of 1-dimensional *centered* discrete Gaussians, as in the BLISS signature scheme [12], this is far from straightforward [5]. Since MITAKA and MITAKA$_{\mathbb{Z}}$, like FALCON and DLP, require discrete Gaussian sampling with *variable* centers, a naive approach to masking is unlikely to yield fast results. Instead, we propose to carry out all online Gaussian sampling operations share-by-share. This allows us to completely

[6] Trivia: Mitaka is a neighborhood in Tokyo, Japan whose name means "the three falcons". It sounded fitting considering the maskable, parallelizable nature of our scheme and its strong points compared to FALCON.

[7] In principle, even more general number fields are possible as well, provided a good basis is known for their canonical embedding. The corresponding security analysis is cumbersome, however.

avoid masking Gaussian sampling operations in the online phase: this works for a masked center, because picking a uniform center in $[0, M)$ with fixed denominator and sampling a discrete Gaussian around that center results in a close to uniform distribution modulo M. Carrying out this share-by-share sampling directly causes a slight decrease in the quality of the resulting sampler (depending on the number of shares). Since larger MITAKA parameters offer a comfortable security margin, we regard this slight reduction in security level as an acceptable compromise to achieve very efficient side-channel protection.

2 Preliminaries

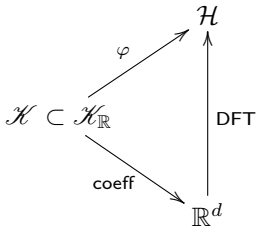
A lattice $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^m$. If it is generated by (the columns of) $\mathbf{B} \in \mathbb{R}^{m\times d}$, we also write $\mathscr{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^d\}$. If $\mathbf{B}$ has full column rank, then we call $\mathbf{B}$ a basis and d the rank of $\mathscr{L}$. The volume of $\mathscr{L}$ is $\mathrm{Vol}(\mathscr{L}) = \det(\mathbf{B}^T\mathbf{B})^{\frac{1}{2}}$ for any basis $\mathbf{B}$.

2.1 Power-of-two cyclotomic fields.

For the rest of this article, we let $d = 2^\ell$ for some integer $\ell \geqslant 1$ and ζ_d to be a $2d$-th primitive root of 1. Then for a fixed d, $\mathscr{K} := \mathbb{Q}(\zeta_d)$ is the d-th power-of-two cyclotomic field, and comes together with its ring of algebraic integers $\mathscr{R} := \mathbb{Z}[\zeta_d]$. We are also interested in the field automorphism $\zeta_d \mapsto \zeta_d^{-1} = \overline{\zeta_d}$, which corresponds to the complex conjugation. We call *adjoint* the image f^* of f under this automorphism.

We also have $\mathscr{K} \simeq \mathbb{Q}[x]/(x^d + 1)$ and $\mathscr{R} \simeq \mathbb{Z}[x]/(x^d + 1)$, so that elements in cyclotomic fields can be seen as polynomials. In other words, each $f = \sum_{i=0}^{d-1} f_i\zeta_d^i \in \mathscr{K}$ can be identified with its coefficient vector $(f_0, \ldots, f_{d-1})$. In this representation the adjoint is given by $f^* = (f_0, -f_{d-1}, \ldots, -f_1)$. The real algebra $\mathscr{K}_\mathbb{R} := \mathscr{K} \otimes \mathbb{R}$ corresponds algebraically to $\mathbb{R}[x]/(x^d + 1)$; in other words it extends coefficients of polynomials to real numbers. The adjoint operation extends naturally to $\mathscr{K}_\mathbb{R}$, and we denote by $\mathscr{K}_\mathbb{R}^+$ the subspace of conjugation invariant elements, i.e. satisfying $f^* = f$. We let $\mathscr{K}_\mathbb{R}^{++}$ be the subset of $\mathscr{K}_\mathbb{R}^+$ which have all positive coordinates in the canonical embedding. Square root extends over this set by taking the coordinate-wise (real) square root. The cyclotomic field $\mathscr{K}$ comes with d complex field embeddings $\varphi_i : \mathscr{K} \to \mathbb{C}$ which map f seen as a polynomial to its evaluations at the odd powers of ζ_d. This leads to the so-called *canonical embedding* of $\mathscr{K}$ is $\varphi(f) := (\varphi_1(f), \ldots, \varphi_d(f))$. Note that $\varphi(fg) = (\varphi_i(f)\varphi_i(g))_{i\leqslant d}$ and that we have $\sqrt{d}\|f\| = \|\varphi(f)\|$, where both norms are ℓ_2 norm over the coefficients of the elements. The canonical embedding extends straightforwardly to $\mathscr{K}_\mathbb{R}$.

Representation of cyclotomic numbers. For this work, there are two useful representations of $\mathscr{K}_\mathbb{R}$, and embeddings of $\mathscr{K}$ extend straighforwardly to this algebra. The coefficient embedding directly identifies a real polynomial in $\mathbb{R}[x]/(x^d + 1)$ with its coefficient vector in $\mathbb{R}^d$ endowed with the standard dot product, while the embedding φ identifies it to the space $\mathcal{H} = \{\mathbf{v} \in \mathbb{C}^d : v_i = \overline{v_{d/2+i}}, 1 \leqslant i \leqslant d/2\}$. The standard Hermitian product on $\mathbb{C}^d$ induces a real inner product on $\mathcal{H}$. The Discrete Fourier Transform (DFT) gives a linear isomorphism between the coefficient embedding space and $\mathcal{H}$, which is depicted by the commutative diagram on the right. In practice, it can be computed in quasilinear time using the Fast Fourier Transform (FFT) algorithm.



2.2 Matrices of algebraic numbers and NTRU modules.

2.2.1 2×2 $\mathscr{K}$-valued matrices. This work deals with free $\mathscr{R}$-modules of rank 2 in $\mathscr{K}^2$, or in other words, groups of the form $\mathscr{R}\mathbf{x} + \mathscr{R}\mathbf{y}$ where $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$ span $\mathscr{K}^2$. There is a natural $\mathscr{K}$-bilinear form over $\mathscr{K}^2$ defined by $\langle \mathbf{x}, \mathbf{y}\rangle_\mathscr{K} := x_1^*y_1 + x_2^*y_2 \in \mathscr{K}$. It can be checked that for all $\mathbf{x} \in \mathscr{K}^2$, $\langle \mathbf{x}, \mathbf{x}\rangle_\mathscr{K} \in \mathscr{K}_\mathbb{R}^{++}$. This form comes with a corresponding notion of orthogonality, and in particular, the well-known Gram-Schmidt orthogonalization procedure can be defined for a pair of linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathscr{K}^2$ by

$$\widetilde{\mathbf{b}}_1 := \mathbf{b}_1,\ \widetilde{\mathbf{b}}_2 := \mathbf{b}_2 - \frac{\langle \mathbf{b}_1, \mathbf{b}_2\rangle_\mathscr{K}}{\langle \mathbf{b}_1, \mathbf{b}_1\rangle_\mathscr{K}} \cdot \widetilde{\mathbf{b}}_1.$$

One readily checks that $\langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}} = 0$. From a computational perspective and thanks to the canonical embedding, this process can be thought of as $d/2$ simultaneous Gram-Schmidt orthogonalizations. The Gram-Schmidt matrix with columns $\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2$ is denoted by $\widetilde{\mathbf{B}}$ and we have $\det \widetilde{\mathbf{B}} = \det \mathbf{B}$.

For $\Sigma \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$, we write Σ^* its conjugate-transpose, where $*$ is the conjugation in $\mathscr{K}$. We extend the notion of positive definiteness for matrices with entries in $\mathscr{K}_{\mathbb{R}}$: we say that $\Sigma \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$ is positive definite when it is symmetric and all the d matrices induced by the canonical embedding are positive definite. We then write $\Sigma \succ 0$. For $s > 0$, we abuse notation and write $\Sigma \succ s$ if $\Sigma - s\mathbf{I} \succ 0$. Positive definite matrices admit "square roots", that is, matrices $\sqrt{\Sigma}$ such that $\sqrt{\Sigma}\sqrt{\Sigma}^* = \Sigma$.

In this work, we consider two notions of "quality" for a given $\mathbf{B} \in \mathscr{K}^{2\times 2}$, as described in [41]. The first is $|\mathbf{B}|_{\mathscr{K}} := \max_{1\leqslant i\leqslant 2} \|\varphi(\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i \rangle_{\mathscr{K}})\|_\infty^{1/2}$. For the second, one checks that $\mathbf{B}^*\mathbf{B}$ is a positive definite matrix, so that its eigenvalues λ_1, λ_2 all in $\mathscr{K}^{++}$ and coordinate-wise square roots are well-defined. The largest singular value of (the embeddings of) $\mathbf{B}$ is recovered as $s_1(\mathbf{B}) := \max_{1\leqslant i\leqslant 2} \|\varphi(\sqrt{\lambda_i})\|_\infty$.

NTRU Modules Given $f, g \in \mathscr{R}$ such that f is invertible modulo some prime $q \in \mathbb{Z}$, we let $h = f^{-1}g \bmod q$. The NTRU module determined by h is $\mathscr{L}_{\mathrm{NTRU}} = \{(u, v) \in \mathscr{R}^2 \,:\, uh - v = 0 \bmod q\}$. Two bases of this free module are of particular interest for cryptography:

$$\mathbf{B}_h = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \text{ and } \mathbf{B}_{f,g} = \begin{bmatrix} f & F \\ g & G \end{bmatrix},$$

where $F, G \in \mathscr{R}$ are such that $fG - gF = q$.

This module is usually seen as a lattice of volume q^d in $\mathbb{R}^{2d}$ thanks to the coefficient embedding. In this work we also consider its version under the embedding φ, which can be described thanks to the matrices

$$\varphi_i(\mathbf{B}_{f,g}) := \begin{bmatrix} \varphi_i(f) & \varphi_i(F) \\ \varphi_i(g) & \varphi_i(G) \end{bmatrix} \text{ and } \varphi(\mathbf{B}_{f,g}) := \begin{bmatrix} \varphi_1(\mathbf{B}_{f,g}) & & \\ & \ddots & \\ & & \varphi_d(\mathbf{B}_{f,g}) \end{bmatrix} \in \mathbb{C}^{2d\times 2d}.$$

Noting that $\det \varphi_i(\mathbf{B}_{f,g}) = \varphi_i(\det \mathbf{B}_{f,g}) = q$, the lattice $\varphi(\mathbf{B}_{f,g}\mathscr{R}^2)$ has volume $q^d\Delta_K$ in $\mathcal{H}$.

Let $\mathbf{B}_{f,g} = [\mathbf{b}_1 | \mathbf{b}_2]$ and $\widetilde{\mathbf{b}}_2 = (\widetilde{F}, \widetilde{G})$ the second Gram-Schmidt vector of $\mathbf{B}_{f,g}$. For all $i \leqslant d/2$, note that $\varphi_i(\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_{\mathscr{K}}) = |\varphi_i(f)|^2 + |\varphi_i(g)|^2$ and $\varphi_i(\langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}) = |\varphi_i(\widetilde{F})|^2 + |\varphi_i(\widetilde{G})|^2$. In particular, the Euclidean norm of the largest Gram-Schmidt vector of $\varphi(\mathbf{B}_{f,g})$ is $|\mathbf{B}_{f,g}|_{\mathscr{K}}$. From the explicit description of $\mathbf{B}_{f,g}$, one can derive formulae for the associated quality parameters.

Lemma 1 ([41], adapted). *Let $\mathbf{B}_{f,g}$ be a basis of an NTRU module. We have $\sqrt{q} \leqslant |\mathbf{B}_{f,g}|_{\mathscr{K}} \leqslant s_1(\mathbf{B}_{f,g})$ and :*

$$|\mathbf{B}_{f,g}|_{\mathscr{K}}^2 = \max\left(\|\varphi(ff^* + gg^*)\|_\infty, \left\| \frac{q^2}{\varphi(ff^* + gg^*)} \right\|_\infty \right),$$

$$s_1(\mathbf{B}_{f,g})^2 = \frac{1}{2} \|\varphi\big(ff^* + gg^* + FF^* + GG^* + \sqrt{(ff^* + gg^* + FF^* + GG^*)^2 - 4q^2}\big)\|_\infty.$$

Proof. With $\mathbf{B}_{f,g} = [\mathbf{b}_1 | \mathbf{b}_2]$, we have $\widetilde{\mathbf{b}}_1 = (f, g)$, we have $ff^* + gg^* = \langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle_{\mathscr{K}}$. We also have $\det \widetilde{\mathbf{B}}^* \widetilde{\mathbf{B}} = q^2 = \langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle_{\mathscr{K}} \langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}$. Both the left side of the claimed inequality and the expression of $|\mathbf{B}_{f,g}|_{\mathscr{K}}$ follow.

Let $\mathbf{e}_1, \ldots, \mathbf{e}_{2d}$ be the canonical basis of $\mathbb{C}^{2d}$. We have $\|\varphi(\mathbf{B}_{f,g})\mathbf{e}_{2i-1}\|^2 = |\varphi_i(f)|^2 + |\varphi_i(g)|^2$ and $\|\varphi(\mathbf{B}_{f,g})\mathbf{e}_{2i}\|^2 = |\varphi_i(F)|^2 + |\varphi_i(G)|^2$, so that $\max(\|\varphi(\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_{\mathscr{K}})\|_\infty, \|\varphi(\langle \mathbf{b}_2, \mathbf{b}_2 \rangle_{\mathscr{K}})\|_\infty) \leqslant s_1(\mathbf{B})^2$ by definition of the operator norm. We check that $\varphi_i(\widetilde{\mathbf{b}}_2)$ is the Gram-Schmidt orthogonalization of $\varphi_i(\mathbf{b}_2)$ with respect to $\varphi_i(\mathbf{b}_1)$, hence we have $\|\varphi_i(\widetilde{\mathbf{b}}_2)\|^2 = \varphi_i(\langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}) \leqslant \|\varphi_i(\mathbf{b}_2)\|^2 = \varphi_i(\langle \mathbf{b}_2, \mathbf{b}_2 \rangle_{\mathscr{K}})$, for all i. This implies the right side of the inequality. Next, the trace of $\mathbf{B}_{f,g}^* \mathbf{B}_{f,g}$ is $T = ff^* + gg^* + FF^* + GG^* \in \mathscr{K}^{++}$, so that its characteristic polynomial is $\chi = X^2 - TX + q^2$ with totally real discriminant $\Delta = T^2 - 4q^2$. If there was an embedding $\mathscr{K} \to \mathbb{C}$ where Δ is negative, this would contradict that the corresponding embedding of $\mathbf{B}_{f,g}^* \mathbf{B}_{f,g}$ is positive definite in the usual sense. Hence $\Delta \in \mathscr{K}^{++}$, and this also means that (each embedding of) $T + \sqrt{\Delta}$ is larger than (the corresponding embedding of) $T - \sqrt{\Delta}$. The last claim follows.

Algorithm 1: RingPeikert sampler

Input: A matrix $\mathbf{B} \in \mathscr{K}^{2\times 2}$ such that $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$ and a target center $\mathbf{c} \in \mathscr{K}_{\mathbb{R}}^2$;
Result: $\mathbf{z} \in \mathscr{L}$ with distribution negligibly far from $D_{\mathscr{L},\mathbf{c},\Sigma}$

Precomputed: a parameter $r \geqslant \eta_\varepsilon(\mathscr{R}^2)$, and $\Sigma_0 \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$ such that $\Sigma_0\Sigma_0^* = \Sigma - r^2\mathbf{B}\mathbf{B}^*$
$\mathbf{x} \leftarrow \Sigma_0 \cdot (\mathcal{N}_{\mathscr{K}_{\mathbb{R}},1})^2$
$\mathbf{z} \leftarrow \lceil \mathbf{B}^{-1}(\mathbf{c} - \mathbf{x}) \rfloor_r$
return $\mathbf{Bz}$

2.3 Gaussians over rings.

The spherical Gaussian function on $\mathbb{R}^d$ centered at $\mathbf{c}$ and with covariance matrix $\Sigma \succ 0$ is defined as $\rho_{\mathbf{c},\Sigma}(\mathbf{x}) = \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{c})\Sigma^{-1}(\mathbf{x}-\mathbf{c}))$. If $\Sigma = s^2\mathbf{I}_d$, we write also $\rho_{\mathbf{c},s} = \exp(-\|\mathbf{x}-\mathbf{c}\|^2/(2s^2))$ and we omit $\mathbf{c}$ if it is $\mathbf{0}$. The normal distribution $\mathcal{N}_\Sigma$ of covariance Σ then has density probability function $((2\pi)^d \cdot \det\Sigma)^{-1/2}\rho_\Sigma$. When we write $\mathcal{N}_{\mathscr{K}_{\mathbb{R}},s}$, we mean that $(z_1,\ldots,z_d) \leftarrow (\mathcal{N}_s)^d$ is sampled and $(z_1+iz_2,\ldots,z_{d-1}+iz_d, z_1 - iz_2,\ldots,z_{d-1}-iz_d) \in \mathcal{H}$ is outputted.

The discrete Gaussian distribution over a full rank lattice $\mathscr{L}$, centered at $\mathbf{c}$ and with covariance matrix $\Sigma \succ 0$ has density function given by

$$\forall\, \mathbf{x} \in \mathscr{L}, D_{\mathscr{L},\mathbf{c},\Sigma}(\mathbf{x}) = \frac{\rho_{\mathbf{c},\Sigma}(\mathbf{x})}{\rho_{\mathbf{c},\Sigma}(\mathscr{L})}.$$

For $c \in \mathscr{K}_{\mathbb{R}}$ and $s > 0$, we also use the notation $\lfloor c \rceil_s$ to denote the distribution $D_{\varphi(\mathscr{R}),\varphi(c),s}$. It extends coordinate-wise to vectors in $\mathscr{K}_{\mathbb{R}}^2$. For $\varepsilon > 0$, the smoothing parameter of a lattice $\mathscr{L}$ is $\eta_\varepsilon(\mathscr{L}) = \min\{s > 0 : \rho_{1/s}(\mathscr{L}^\vee) < 1+\varepsilon\}$, where $\mathscr{L}^\vee$ is the dual lattice. Its exact definition[8] is not needed in this work, and when $\mathscr{L} = \mathscr{L}(\mathbf{B})$, it is enough to know the matrix $\mathbf{B}^{-*}$ describes it. We use the following bound.

Lemma 2 (Adapted from [19]). *Let $\mathbf{B}\mathscr{R}^2$ be free $\mathscr{R}$-module, and let $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$ be the associated rank d lattice in $\mathcal{H}^2$. For all $\varepsilon > 0$, we have*

$$\eta_\varepsilon(\mathscr{L}) \leqslant |\mathbf{B}|_{\mathscr{K}} \cdot \frac{1}{\pi}\sqrt{\frac{\log(4d(1+1/\varepsilon))}{2}}.$$

The rightmost quantity is also an upper bound on $\eta_\varepsilon(\mathscr{R}^2)$, which we usually use as a placeholder in this work, for the sake of readability.

2.4 Sampling discrete Gaussians in rings.

In this section, we recall different approaches toward Gaussian sampling in integer rings following [41]. We give pseudo-codes to describe the main steps.

2.4.1 Randomized rounding in rings. In [38], Peikert presented an efficient algorithm to sample discrete Gaussians in a target lattice, as long as the needed standard deviation parameter is large enough. It is essentially a randomized version of Babai's round-off algorithm, and can be formulated directly over the algebra $\mathscr{K}_{\mathbb{R}}$. The pseudo-code in Algorithm 1 outputs discrete Gaussians in a free rank 2 $\mathscr{R}$-module $\mathscr{L}$ described by a basis $\mathbf{B} \in \mathscr{K}^{2\times 2}$, with an arbitrary center in $\mathscr{K}_{\mathbb{R}}^2$. When $\Sigma \succ r^2\mathbf{B}\mathbf{B}^*$, the existence of Σ_0 below is guaranteed.

Theorem 1 ([38], adapted). *Let $\mathscr{D}$ be the output distribution of Algorithm 1. If $\varepsilon \leqslant 1/2$ and $\sqrt{\Sigma} \succ s_1(\mathbf{B}) \cdot \eta_\varepsilon(\mathscr{R}^2)$, then the statistical distance between $\mathscr{D}$ and $D_{\mathscr{L},\mathbf{c},\Sigma}$ is bounded by 2ε. Moreover, we have*

$$\sup_{\mathbf{x}\in\mathbf{B}\mathscr{R}^2} \left| \frac{\mathscr{D}(\mathbf{x})}{D_{\mathscr{L}(\mathbf{B}),\mathbf{c},\Sigma}(\mathbf{x})} - 1 \right| \leqslant 4\varepsilon.$$

[8] If the lattice corresponds to the canonical embedding of a free $\mathscr{R}$-module, it would involve the *co-different* $\mathscr{R}^\vee$.

Algorithm 2: RingPeikert, one-dimensional version

Input: A target center $c \in \mathscr{K}_{\mathbb{R}}$
Result: $z \in \mathscr{R}$ with distribution negligibly far from $D_{\mathscr{R},c,\Sigma}$

Precomputed: a parameter $r \geqslant \eta_\varepsilon(\mathscr{R})$, *and* $\sigma_0 \in \mathscr{K}_{\mathbb{R}}$ *such that* $\sigma_0^*\sigma_0 = \Sigma - r^2$

$x \leftarrow \mathcal{N}_{\mathscr{K}_{\mathbb{R}},\sigma_0}$
return $\lceil c - x \rfloor_r$

Algorithm 3: Hybrid Module Gaussian Sampler

Input: A target center $\mathbf{c} \in \mathscr{K}_{\mathbb{R}}^2$, a matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ such that $\mathscr{L} = \mathbf{B}\mathscr{R}^2$ and its GSO $[\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2]$ over $\mathscr{K}$, a covariance parameter $\Sigma \in \mathscr{K}_{\mathbb{R}}^{++}$.
Result: $\mathbf{z}$ with distribution negligibly far from $D_{\mathscr{L},\mathbf{c},\Sigma}$

$\mathbf{c}_2 \leftarrow \mathbf{c}, \mathbf{v}_2 \leftarrow 0$
$d_2 \leftarrow \frac{\langle \widetilde{\mathbf{b}}_2, \mathbf{c}_2 \rangle_{\mathscr{K}}}{\langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}}$
$\Sigma_2 \leftarrow \Sigma / \langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}$
$z_2 \leftarrow \mathsf{RingPeikert}_1(d_2, \Sigma_2, \sqrt{\Sigma_2 - r^2})$
$\mathbf{c}_1 \leftarrow \mathbf{c}_2 - z_2\mathbf{b}_2, \mathbf{v}_1 \leftarrow z_2\mathbf{b}_2$
$d_1 \leftarrow \frac{\langle \widetilde{\mathbf{b}}_1, \mathbf{c}_1 \rangle_{\mathscr{K}}}{\langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle_{\mathscr{K}}}$
$\Sigma_1 \leftarrow \Sigma / \langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle_{\mathscr{K}}$
$z_1 \leftarrow \mathsf{RingPeikert}_1(d_1, \Sigma_1, \sqrt{\Sigma_1 - r^2})$
$\mathbf{v}_0 \leftarrow \mathbf{v}_1 + z_1\mathbf{b}_1$
return $\mathbf{v}_0$

From Lemma 1 and 2, note that the condition in the statement ensure that we are above the smoothing parameter of the target lattice. Following standard Renyi divergence arguments as in [42, 5, 1], $\varepsilon = 2^{-39}$, resp. 2^{-40} ensures that using Algorithm 1 preserves up to 128, resp. 256 bits of security from an ideal sampler, which is quite enough for our purpose. In practice this algorithm involves floating point arithmetic. We analyze the necessary precision to preserve standard security levels in Appendix E. As it is used in the next section, we highlight in Algorithm 2 the one dimensional version, that is, outputting discrete Gaussians in $\mathscr{R}$.

2.4.2 Randomized nearest plane in rings. In [41], a so-called hybrid sampler is presented to output discrete Gaussians in free $\mathscr{R}$ modules of finite rank and with basis $\mathbf{B} \in \mathscr{K}^{2\times 2}$. On a high level, it follows Klein's approach, which is a randomized version of the Nearest Plane algorithm, but in the hybrid case the randomization subroutine happens "at the ring level" thanks to a ring Gaussian sampler. It leads to an efficiency-quality trade-off between Algorithm 1 and the Nearest Plane algorithm "at the integer level", that is, in the module seen as a lattice. We summarize it in Algorithm 3 for rank 2 modules and "spherical" covariance over $\mathscr{K}_{\mathbb{R}}$, which are our main interest in this work.

We note that in practice, $\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i \rangle_{\mathscr{K}}$, Σ_i and $\sqrt{\Sigma_i - r^2}$ can all be precomputed during the key-generation. The next result is proved in Appendix E.

Theorem 2 ([41], Theorem 5.10, adapted). *Let $\mathscr{D}$ be the output distribution of Algorithm 1. If $\varepsilon \leqslant 2^{-5}$ and $\sqrt{\Sigma} \succ |\mathbf{B}|_{\mathscr{K}} \cdot \eta_\varepsilon(\mathscr{R}^2)$, then the statistical distance between $\mathscr{D}$ and $D_{\mathscr{L},\mathbf{c},\Sigma}$ is bounded by 7ε. Moreover, we have*

$$\sup_{\mathbf{x}\in\mathbf{B}\mathscr{R}^2} \left| \frac{\mathscr{D}(\mathbf{x})}{D_{\mathscr{L}(\mathbf{B}),\mathbf{c},\Sigma}(\mathbf{x})} - 1 \right| \leqslant 14\varepsilon.$$

Algorithm 4: Hybrid Module Gaussian Sampler

Input: A target center $\mathbf{c} \in \mathscr{K}_{\mathbb{R}}^2$, a matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ such that $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$ and its GSO $[\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2]$ over $\mathscr{K}$, a parameter $\sigma > 0$ (corresponding to $(\sigma, \ldots, \sigma) \in \mathscr{K}_{\mathbb{R}}$).
Result: $\mathbf{z}$ with distribution negligibly far from $D_{\mathscr{L},\mathbf{c},\sigma^2\mathbf{I}_{2d}}$

Precomputed: $\sigma_i := \sqrt{\frac{\sigma^2}{\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i \rangle} - r^2} \in \mathscr{K}_{\mathbb{R}}^{++}$.
$\mathbf{c}_2 \leftarrow \mathbf{c}, \mathbf{v}_2 \leftarrow 0$
$d_2 \leftarrow \frac{\langle \widetilde{\mathbf{b}}_2, \mathbf{c}_2 \rangle_{\mathscr{K}}}{\langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}}_2 \rangle_{\mathscr{K}}}$
$u_2 \leftarrow \mathcal{N}_{1,\mathscr{K}_{\mathbb{R}}}$
$y_2 \leftarrow \sigma_2 \cdot u_2$
$x_2 \leftarrow \lfloor d_2 - y_2 \rceil_r$
$\mathbf{c}_1 \leftarrow \mathbf{c}_2 - x_2\mathbf{b}_2, \mathbf{v}_1 \leftarrow x_2\mathbf{b}_2$
$d_1 \leftarrow \frac{\langle \widetilde{\mathbf{b}}_1, \mathbf{c}_1 \rangle_{\mathscr{K}}}{\langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle_{\mathscr{K}}}$
$u_1 \leftarrow \mathcal{N}_{1,\mathscr{K}_{\mathbb{R}}}$
$y_1 \leftarrow \sigma_1 \cdot u_1$
$x_1 \leftarrow \lfloor d_1 - y_1 \rceil_r$
$\mathbf{v}_0 \leftarrow \mathbf{v}_1 + x_1\mathbf{b}_1$
return $\mathbf{v}_0$

More concretely, the hybrid sampler that we will consider in this work is presented in Algorithm 4. Among the differences with the previous and more general version, below the standard deviation parameter below is spherical and the "Peikert" steps are made explicit. In Appendix E, we also analyze the necessary floating point precision to preserve enough bits of security.

2.4.3 Asymptotic security of the samplers. Hash-and-sign signatures over lattices are constructed, following the GPV framework [19], by hashing a message to the ambient space of the lattice, and returning as a signature a lattice point close to that hash digest. This is done using a "good" representation of the lattice, called the *trapdoor*, that enables the signer to solve the ApproxCVP problem with a relatively small approximation factor. Moreover, to prevent signatures from leaking information about the secret trapdoor, the close lattice points need to be sampled according to a distribution that is statistically independent of the trapdoor: usually a spherical discrete Gaussian distribution supported over the lattice and centered at the hash digest. This is were the algorithms from the previous sections come into play.

The security of the resulting signature scheme depends on the standard deviation of the discrete Gaussian distribution output by the sampler: the smaller the standard deviation, the closer the distance to the hash digest, the harder the corresponding ApproxCVP problem, and hence the higher the security level. As we have seen, however, there is a lower bound (depending on the trapdoor) to how small of a standard deviation the sampler can achieve while remaining statistically close to the desired spherical Gaussian: lower than that, and the distribution may start to deviate from that Gaussians in ways that could expose information about the secret trapdoor, and thus compromise the security of the signing key.

In the case of NTRU lattices, the trapdoor is the secret basis:

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

and the standard deviation of the discrete Gaussian obtained from this trapdoor varies depending on the sampling algorithm, as discussed in particular in [41, §6]. It can be written as:

$$\sigma = \alpha \cdot \eta_\varepsilon(\mathscr{R}^2) \cdot \sqrt{q}$$

where the factor $\alpha \geqslant 1$, which we call the *quality*, depends on the sampler for a given trapdoor.

Table 1. Comparison of the best achievable trapdoor quality α for the various Gaussian samplers over NTRU lattices.

Sampler	Complexity	Expression of $\alpha\sqrt{q}$	Best achievable α
Peikert	$O(d \log d)$	$s_1(\mathbf{B}_{f,g})$	$O(d^{1/4}\sqrt{\log d})$ [41, §6.5.2]
Hybrid	$O(d \log d)$	$\lvert\mathbf{B}_{f,g}\rvert_{\mathscr{K}}$	$O(d^{1/8}\sqrt[4]{\log d})$ [Appendix A]
Klein (FALCON)	$O(d \log d)$	$\lVert\mathbf{B}_{f,g}\rVert_{\mathrm{GS}}$	$O(1)$ [41, §6.5.1]

For the so-called Klein sampler used in DLP and FALCON, $\alpha\sqrt{q}$ is the Gram–Schmidt norm $\lVert\mathbf{B}_{f,g}\rVert_{\mathrm{GS}} := \max_{1\leqslant i\leqslant 2d} \lVert\widetilde{\mathbf{b}}_i^{\mathbb{Z}}\rVert_2$ of $\mathbf{B}_{f,g}$ *over the integers*. For the Peikert sampler over $\mathscr{K}$, Theorem 1 shows that $\alpha\sqrt{q} = s_1(\mathbf{B}_{f,g})$. And finally, for the hybrid sampler, Theorem 2 shows that $\alpha\sqrt{q} = \lvert\mathbf{B}_{f,g}\rvert_{\mathscr{K}}$.

For a given sampler, the generators f, g should be sampled appropriately in order to minimize the corresponding α. In his thesis [41], Prest analyzed the optimal choices both theoretically (under suitable heuristics) and experimentally. The resulting optimal choices for α are as follows (after correcting the flawed heuristic analysis of Prest in the case of the hybrid sampler: see our discussion in Appendix A):

- the Peikert sampler heuristically statisfies $\alpha = O(d^{1/4}\sqrt{\log d})$ [41, §6.5.2];
- for the hybrid sampler, following Appendix A, we show $\alpha = O(d^{1/8}\sqrt[4]{\log d})$ (and not $O(\sqrt{\log d})$ contrary to what was claimed in [41, §6.5.2] based on flawed heuristics);
- for the Klein sampler (used in DLP, and in modified form, FALCON), the heuristic analysis in [41, §6.5.1] show that it can be taken as low as $\sqrt{e/2} \approx 1.17$ independently of the dimension, and in particular $\alpha = O(1)$.

These properties are summarized in Table 1.

3 Improved Trapdoor Generation

The Peikert, hybrid and FALCON samplers for an NTRU basis $\mathbf{B}_{f,g}$ all have essentially the same complexity, and the first two are significantly simpler, easier to implement, slightly faster in the same dimension, and offer better avenues for parallelization and side-channel resistance (see Section 5). It would therefore be desirable to adopt one of the first two for practical implementations.

However, as seen in Section 2.4.3, the FALCON sampler has a substantial advantage in terms of security, since its Gaussian standard deviation is proportional to $\lVert\mathbf{B}_{f,g}\rVert_{\mathrm{GS}}$, whereas the Peikert and hybrid samplers are proportional to $s_1(\mathbf{B}_{f,g})$ and $\lvert\mathbf{B}_{f,g}\rvert_{\mathscr{K}}$ respectively, which are both larger. This results in a significant difference in asymptotic terms, as shown in Table 1, and also in bit security terms as will become apparent in the next section.

In order to increase the security level achievable using the first two samplers, and in particular the hybrid sampler, we propose a new technique to significantly improve the quality of NTRU trapdoors. We note in passing that it also applies to FALCON: while it cannot yield significant improvements in terms of security, since the standard deviation it achieves is already a very small factor away from the theoretical optimum, it can be used to speed up key generation substantially. The idea is as follows.

Recall that NTRU trapdoor generation for FALCON, say, works by sampling f, g with discrete Gaussian coefficient, computing the $\lVert\mathbf{B}_{f,g}\rVert_{\mathrm{GS}}$ of the resulting NTRU basis, and checking if this value is below the desired quality threshold. If not, we try again, and if so, the NTRU basis is completed and kept as the secret key. Trapdoor sampling for the hybrid sampler is similar. (On the other hand, for Peikert, completion has to be recomputed at each step to evaluate s_1).

In this process, the costly operations are, on the one hand, the generation of the discrete Gaussian randomness, which has to be repeated several dozen times over in order to reach the desired threshold (this is not explicitly quantified by the authors of FALCON, but experiments suggest that, in many instances, at least 50 iterations are necessary), and, on the other hand, the completion of the basis (still costly despite recent optimizations [40]), which is only carried out once at the end and not for each iteration[9].

[9] This is the case at least for FALCON and for the hybrid sampler, as for both of them, one can compute the quality of the trapdoor given only (f, g). This is especially fast for the hybrid sampler. For the Peikert sampler, however, doing so without also obtaining (F, G) seems difficult, and is left as an open problem.

To optimize the process, our idea is to amortize the cost of discrete Gaussian sampling, by constructing several candidate trapdoors from the same randomness. We propose three main ideas to do so.

Lists of candidates for f and g. The usual key generation algorithm for Falcon, as already mentioned, normally ends up generating many pairs (f_i, g_i), and tests each of them as a candidate first vector for the NTRU lattice.

Since we are generating Gaussian vectors f_i and g_i anyway, we can easily recycle this generated randomness by testing all the mixed pairs (f_i, g_j) instead: this results in a set of possible candidates which increases quadratically with the number of random vectors we generate, instead of just linearly.

Generating the Gaussian vectors as linear combinations. Independently, one can generate each candidate vector f as a linear combination $\sum_{k=1}^{m} f^{(k)}$ where each $f^{(k)}$ is sampled from a discrete Gaussian of standard deviation $\sigma_0/\sqrt{m}$, for σ_0 the desired standard deviation of f. It is well-known that this results in the correct distribution provided that $\sigma_0/\sqrt{m}$ remains above the smoothing parameter [36]. In fact, the Falcon implementation already does so for $d = 512$, where the candidate vectors are sums of two Gaussians vectors of standard deviation $\sqrt{2}$ times lower.

Now, when generating several f_i's, one obtains m lists $L_k = \{f_i^{(k)}\}_i$ of Gaussian vectors. It is again possible to recycle this generated randomness by mixing and matching among those lists, and constructing candidates f of the form $\sum f_{i_k}^{(k)}$ for varying indices i_k, so that the total set of candidates is in bijection with $\prod_k L_k$. Its size increases like the m-th power of the size of the lists.

Using the Galois action. Finally, one can expand the set of candidates for g, say, by applying the action of the Galois group. In principle, other unitary transformations of g, even less structured ones like randomly permuting the coefficients in the power basis, could also be considered, but the Galois action in particular is convenient as it is expressed as a circular permutation on the embeddings $\varphi_i(g)$ of g (i.e., the Fourier coefficients), and for the hybrid sampler, the computation of the quality is entirely carried out in the Fourier domain.

Concretely, recall from Lemma 1 that the quality parameter α of the hybrid sampler associated with $\mathbf{B}_{f,g}$ satisfies:

$$\alpha^2 = \frac{|\mathbf{B}_{f,g}|_{\mathscr{K}}^2}{q} = \max\left(\frac{\max_i z_i}{q}, \frac{q}{\min_i z_i}\right)$$
$$\text{where } z_i = \varphi_i(ff^* + gg^*) = |\varphi_i(f)|^2 + |\varphi_i(g)|^2 \in \mathbb{R}^+.$$

It is easy to compute the embeddings z_i^τ associated to $\mathbf{B}_{f,\tau(g)}$ for some Galois automorphism τ of $\mathscr{K}$ simply by applying the corresponding permutation on the components of $\varphi(g)$. Moreover, we see from this representation that the conjugation $\tau_* \colon g \mapsto g*$ leaves this quality invariant, so the relevant Galois elements to consider are a set of representatives of $\mathrm{Gal}(\mathscr{K}/\mathbb{Q})/\langle\tau_*\rangle$. For power-of-two cyclotomics, one can for example use τ_5^k for $k = 0, \ldots, d/2 - 1$, where $\tau_5(\zeta_d) = \zeta_d^5$.

Security considerations. The techniques above can potentially skew the distribution of f and g somewhat compared to the case when each tested (f, g) that fails to pass the security threshold is thrown away. However, this is not really cause for concern: the precise distribution of f and g is not known to affect the security of the signature scheme other than in two ways:

- the extent to which it affects the geometry of the trapdoor, as encoded in the quality parameter α already; and
- the length of (f, g) itself as it affects key recovery security, but this length is always at least as large in our setting as in Falcon.

This indicates that our optimized secret keys do not weaken the scheme.

Algorithm 5: MITAKA optimized key generation

Input: Desired standard deviation σ_0 of f and g, target quality α of the Gaussian, number of samples m to generate, set G of Galois automorphisms to apply. The total search space is of size $\#G \cdot m^4$ for $4m$ generated discrete Gaussian vectors.

Result: NTRU first trapdoor vector (f, g) with quality better than α.

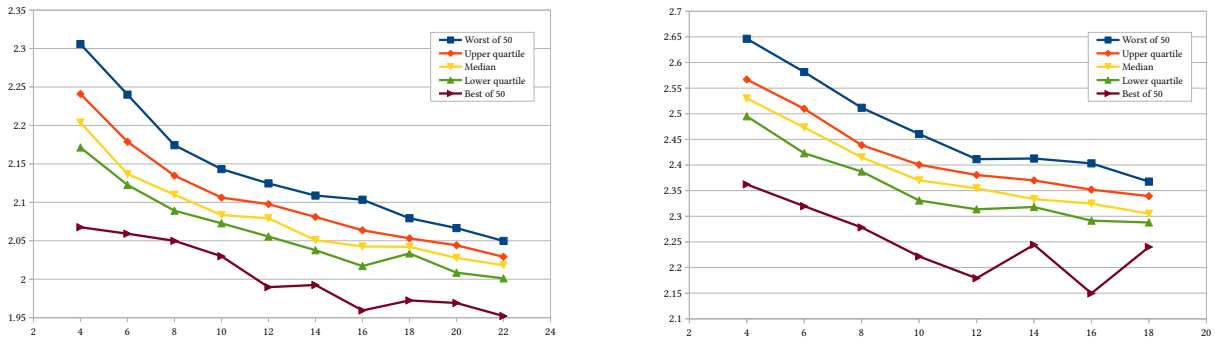
for $i \in [1, m]$ **do**
  $f'_i \leftarrow D_{\mathscr{R},\sigma_0/\sqrt{2}}$, $f''_i \leftarrow D_{\mathscr{R},\sigma_0/\sqrt{2}}$
  $g'_i \leftarrow D_{\mathscr{R},\sigma_0/\sqrt{2}}$, $g''_i \leftarrow D_{\mathscr{R},\sigma_0/\sqrt{2}}$
end for
$L_f \leftarrow \{f'_i + f''_j \mid i, j \in [1, m]\}$
$L_g \leftarrow \{\tau(g'_k + g''_\ell) \mid k, \ell \in [1, m], \tau \in G\}$
$L_u \leftarrow \{(f, \varphi(ff^*)) \mid f \in L_f\}$
$L_v \leftarrow \{(g, \varphi(gg^*)) \mid g \in L_g\}$
for $(f, u) \in L_u, (g, v) \in L_v$ **do**
  $z \leftarrow u + v$
  if $q/\alpha^2 \leqslant z_i \leqslant \alpha^2 q$ *for all* i **then return** (f, g)
end for
restart

Concrete example. In Algorithm 5, we describe an example key generation procedure that combines all three techniques presented above: we construct lists of candidates for f and g and test all possible pairs. Moreover, each f and g itself is sampled as a sum of two narrower Gaussians, and the list of g's is expanded using the Galois action. Of course, different combinations of the techniques are also possible, but this particular one offers a good balance between efficiency and achievable security.

Using this approach, as shown in Fig. 1, we are able to efficiently generate trapdoors with $\alpha \leqslant 2.05$ for $d = 512$, and $\alpha \leqslant 2.35$ for $d = 1024$ by $m \approx 14$ (corresponding to generating 56 narrow Gaussian vectors to select one candidate (f, g), largely in line with FALCON).

4 Security analysis of MITAKA

Concrete security. In order to assess the concrete security of our signature scheme, we proceed using the usual cryptanalytic methodology of estimating the complexity of the best attacks against *key recovery attacks*



(a) Reachable α in dimension 512

(b) Reachable α in dimension 1024

Fig. 1. Quality α reached by the optimized sampler of Algorithm 5 for various choices of m (50 trials each, $\sigma_0 = 1.17\sqrt{q/2d}$, G coset representatives of $\mathrm{Gal}(\mathscr{K}/\mathbb{Q})/\langle\tau_*\rangle$).

Table 2. Concrete values for sampler's quality and corresponding bit security level.

	$d = 512$				$d = 1024$			
	Quality α	Classical	Quantum	NIST Level	Quality α	Classical	Quantum	NIST Level
Falcon	1.17	120	108	I	1.17	280	252	V
Naive Hybrid[a]	3.03	88	80	below I	3.58	216	196	IV
Mitaka **(this work)**	2.05	**100**	**90**	I[b]	2.35	**233**	**210**	V

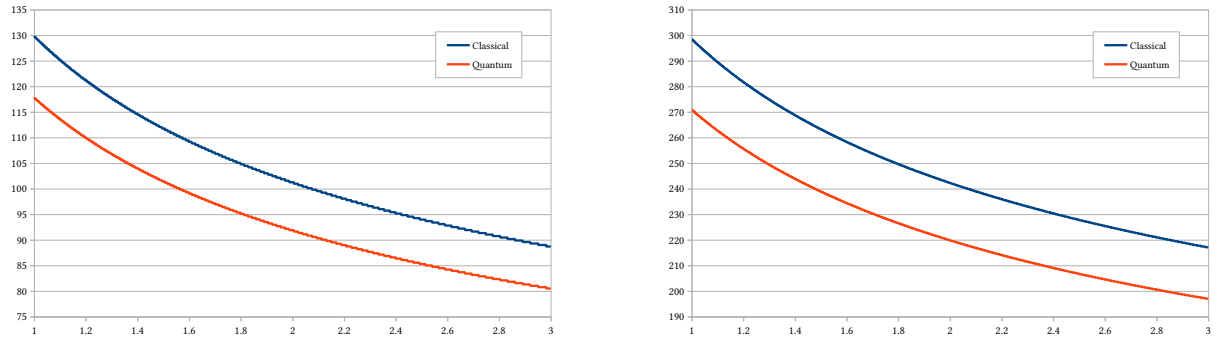
[a] Key generation with the same median amount of randomness as Mitaka Alg. 5 with $m = 14$, but without the optimizations of Section 3.

[b] Taking into account the heavy memory cost of sieving. This is the same level as e.g. Dilithium–II; see the discussion in [31, §5.3].

on the one hand, and *signature forgery* on the other. The detailed analysis is carried out in Appendix C. For the parameter choices relevant to our scheme (in which the vectors of the trapdoor basis are not unusually small), key recovery is always harder than signature forgery, and therefore the cost of signature forgery is what determines the security level. Using the condition of Eq. (5), we see that the security of the forgery is a function of the standard deviation of the lattice Gaussian sampler used in the signature function, which itself depends on the quality α of the trapdoor, as discussed in Section 2.4.3. This analysis translates into concrete bit-security estimates following the methodology of NewHope [2], sometimes called "core-SVP methodology". In this model [6, 29], the bit complexity of lattice sieving (which is asymptotically the best SVP oracle) is taken as $\lfloor 0.292d \rfloor$ in the classical setting and $\lfloor 0.262d \rfloor$ in the quantum setting. The resulting security in terms of α is given in Fig. 2 in dimensions 512 and 1024. This allows us to compare Mitaka with Falcon as well as with a "naive" version of the hybrid sampler that would omit the optimizations of Section 3; the results are presented in Table 2.

In addition, as mentioned earlier, our construction can be instantiated over more general base fields than power-of-two cyclotomics, which enables us to choose security level in a much more flexible way than Falcon. This is analyzed in Appendix B. Example security levels which can be reached in this way are presented in Table 3.

Asymptotic security. As for all signature schemes in the GPV framework, the EUF–CMA security of our scheme in an asymptotic sense reduces, both in the classical [19] and quantum random oracle models [7], to the SIS problem in the underlying lattice (in this case, an instance of Module–SIS [30]). However, as is the case for Falcon (and as holds, similarly, for Dilithium), the SIS bound in Euclidean norm for the standard parameter choice ($q = 12289$) makes the underlying assumption vacuous. This is not known to lead to any



(a) Security for forgery in dimension 512

(b) Security for forgery in dimension 1024

Fig. 2. Security (classical and quantum) against forgery as a function of the quality $1 \leqslant \alpha \leqslant 3$ of the lattice sampler.

Table 3. Suggested intermediate parameters and security levels for MITAKA using other base fields.

	$d = 512$	$d = 648$	$d = 768$	$d = 864$	$d = 972$	$d = 1024$
Cyclotomic Conductor	2^{10}	$2^3 \cdot 3^5$	$2^8 \cdot 3^2$	$2^5 \cdot 3^4$	$2^2 \cdot 3^6$	2^{11}
Quality α	2.05	2.15	2.25	2.30	2.33	2.35
Classical/Quantum Security	100/90	121/110	149/134	172/156	200/180	233/210
NIST Level	I ($\approx$ Dilithium-II)	I ($\approx$ FALCON-512)	II	III	IV	V

attack, and can be addressed by increasing q if so desired, or reducing to the SIS problem in infinity norm instead.

5 Masking Gaussian samplers

5.1 Preliminaries on masking countermeasure

Below we fix some ring $(\mathcal{R}, 0, 1, +, -, \cdot)$. Following the seminal work due to Ishai, Sahai, and Wagner [26], we aim to protect Gaussian samplers described in Section 2.4 from the so-called *t-probing side-channel adversary*, who is able to peek at up to t intermediate variables per each invocation of an algorithm. The side-channel *masking* is a technique to mitigate such attacks, by additively *secret-sharing* every sensitive variables into $T > t$ values in $\mathcal{R}$. The integer t is often referred to as *masking order* an T is usually set to be $t+1$. Clearly, higher masking order allows a side-channel protected implementation to tolerate stronger probing attacks with larger t. For a ring element $a \in \mathcal{R}$, we say that a vector $(a_i)_{i\in[1,t+1]} \in \mathcal{R}^{t+1}$ is a $t+1$-*sharing* of $a := \sum_{i\in[1,t+1]} a_i$. For readability, we write $[\![a]\!]_m := (a_i)_{i\in[1,t+1]}$ for a ring element $a \in \mathbb{Z}/m\mathbb{Z}$. Furthermore, we introduce rather unusual secret sharing over a quotient group $(1/C \cdot \mathbb{Z})/m\mathbb{Z}$ for some fixed integer C: for an element $b \in (1/C\cdot\mathbb{Z})/m\mathbb{Z}$, we write as its additive secret sharing $\langle\!\langle b \rangle\!\rangle_m := (b_i)_{i\in[1,t+1]} \in ((1/C\cdot\mathbb{Z})/m\mathbb{Z})^{t+1}$. Note that $1/C \cdot \mathbb{Z}$ is not a ring, and thus the multiplication is not well-defined. This is not an issue in our application, since we never carry out multiplication of two sharings in the latter form.

The most basic security notion for a masking countermeasure is the *t-privacy* of a gadget G [26]. This notion guarantees that any set of at most t intermediate variables observed during the computation is independent of the secret input. While the idea behind the notion is relatively simple, t-private gadgets are unfortunately not composable, meaning that a gadget consisting of multiple t-private sub-gadgets may not be necessarily secure. Hence in this work we rely on the following more handy security notions introduced by Barthe et al. [3]. We remark that the definition below is a slightly weaker variant of the original one, in which *perfect* simulation of probes is assumed. We require this generalization as probes on one of our sub-gadgets are only *statistically* simulatable, but this clearly does not affect the composition property.

Definition 1 (t-NI, t-SNI). *Let G be a gadget with inputs $(x_i)_{i\in[1,t+1]} \in \mathcal{R}^{t+1}$ and outputs $(y_i)_{i\in[1,t+1]} \in \mathcal{R}^{t+1}$. Suppose that for any set of t_1 intermediate variables and any subset of $O \subset [1, t+1]$ of output indices with $t_1 + |O| \leqslant t$, there exists a subset of indices $I \subset [1, t+1]$ such that the output distribution of the t_1 intermediate variables and the output variables $(y_i)_{i\in O}$ is statistically simulatable from $(x_i)_{i\in I}$. Then*

1. *if $|I| \leqslant t_1 + |O|$ we say G is* t-non-interfering (t-NI)*, and*
2. *if $|I| \leqslant t_1$ we say G is* t-strong-non-interfering (t-SNI)*.*

The above notion can be naturally extended for a gadget with multiple input and output sharings. Note that *linear* operations performed share-wise (such as addition of two sharings, or multiplication by a constant) are trivially t-NI, as each computation on share i can be simulated from the input share x_i. Building blocks satisfying either NI or SNI can be easily composed with each other, by inserting the RefreshM gadgets (see Algorithm 11) at suitable locations to re-randomize shares [3, Proposition 4]. It is also internally used in the Unmask gadget before taking the sum of shares, so that a probe on any partial sum doesn't leak more information than one input share [4].

Typically, the non-interference notions only deal with gadgets where all of the inputs and outputs are sensitive. To also handle public, non-sensitive values, a weaker notion called *NI with public output* (t-NIo) was proposed in [4]. As stated in [4, Lemma 1], if a gadget G is t-NI secure it is also t-NIo secure for any public outputs.

5.2 Masking Peikert and Hybrid samplers

Algorithm 7 (resp. Algorithm 8) corresponds to our masked version of the RingPeikert sampler of Algorithm 1 (resp. the Hybrid sampler of Algorithm 3). Although masked MITAKA is instantiated with the MaskedHybrid sampler, we also include MaskedRingPeikert for completeness because the former can be essentially obtained by extending the basic masking paradigm outlined in the latter.

Unlike previous solutions to masking Gaussian samplers (e.g., [20, 5]), our approach essentially generates samples *share-by-share*, instead of masking all the expensive non-linear operations during the online phase (see Algorithm 6). This gadget is (statistically) NI-secure, as it operates individually on each uniform share of the masked center and therefore any t probes can be simulated with the corresponding t shares of the center up to negligible statistical distance. Although this in turn increases the standard deviation by a factor of $\sqrt{t+1}$, where t is the masking order, we are able to achieve a quite simple masked algorithm, consisting of basic sub-gadgets such as the ISW multiplier (Algorithm 12) and SNI-secure share refresh (Algorithm 11).

One may think that we here need to mask floating-point arithmetic. However, we can avoid it by representing each sensitive variable from $\mathscr{K}_{\mathbb{R}}$ as a fixed-point number. Concretely, an element $x \in \mathscr{K}_{\mathbb{R}}$ is approximated by $\tilde{x} \in \mathscr{K}_{\mathbb{R}}$ such that every coefficient of $q^k\tilde{x}$ is an integer, where k is a parameter determining the precision. Then we can secret-share $q^k\tilde{x}$ in $\mathbb{Z}_M^d$ for $M \gg q^k$. Since we do not perform many multiplication operations, an accumulated scaling factor does not break the correctness of sampler if we choose sufficiently large M.

We also remark that a secret-shared center in fixed-point representation must be divided by a scaling factor q^k for the following 1-dimensional discrete Gaussian sampling to work share-by-share (i.e. division of $[\![q^k v_{i,j}]\!]$ at line 9-12 of Alg. 7, and of $[\![q^k z_{i,j}]\!]$ at lines 12-14 and 19-21 of Alg. 8, respectively). This division can be performed in floating-point arithmetic in practice. For the sum of shares to represent the correct center in the MaskedRingPeikert sampler, we further set $M = q^{\ell+k}$ for some $\ell > 0$. The resulting shares after division form a sharing of $\mathbf{v} = [(v_{1,j})_{j\in[0,d-1]}, (v_{2,j})_{j\in[0,d-1]}]$ over $((1/q^k \cdot \mathbb{Z})/q^\ell\mathbb{Z})^{2d}$. Assuming $r/\sqrt{2(t+1)}$ exceeds the smoothing parameter (according to [34]), Algorithm 6 outputs mod-q additive shares of an integer sample statistically close to $D_{\mathbb{Z},v_{i,j},r}$ for $i = 1, 2$ and $j \in [0, d-1]$. Since the output values of the signature are defined mod q we can further map the shares to $[\![\cdot]\!]_q$ and the remaining computations can be performed mod q.

Since we invoke the above routine twice in the MaskedHybrid sampler, the initial masking modulus needs to be increased so that no wrap-around occurs during the masked computation of the second nearest plane. Concretely, the first nearest plane operations are computed with modulus $M = q^{\ell+2k}$; the second nearest plane operations are performed on $[\![\cdot]\!]_{q^{\ell+k}}$, with corresponding arithmetic shares of sensitive inputs; the output values can be represented in $[\![\cdot]\!]_q$ as in the MaskedRingPeikert.

Finally, we briefly discuss possible options to instantiate other sub-gadgets.

- SNI-secure PolyMul and MatMul can be implemented in a rather straightforward manner by combining linear operations and the standard ISW multiplier (Algorithm 12). For example, one can compute the canonical embedding (resp. the coefficient embedding) by recursively invoking `mergefft` and `split` (resp. `splitfft` and `merge`) routine of FALCON share-wise in fixed-point arithmetic. Then the result is obtained by calling the ISW multiplier coordinate-wise. When these gadgets are called back-to-back, one should of course maintain the FFT representation and bring it back to the coefficient domain right before the division by a scaling factor. For further optimization, we can in practice rely on masked NTT instead of FFT. Notice that the masked instances only deal with a multiplication between polynomials mapped to $\mathbb{Z}_M[x]/(x^d+1)$ (or $\mathbb{Z}_{q^{\ell+k}}[x]/(x^d+1)$ during the second nearest plane of the Hybrid sampler) thanks to the fixed-point representation. One caveat is that in the current setting M is restricted to a *power* of q, but we are able to show such a choice is indeed NTT-friendly. Recall that the prime q is usually chosen such that $q = 1 \bmod 2d$, so that $x^d + 1$ has exactly d roots $(\zeta, \zeta^3, \ldots, \zeta^{2d-1})$ over $\mathbb{Z}_q$. Now thanks to the Hensel lifting, one can construct another set of d roots $(\omega, \omega^3, \ldots, \omega^{2d-1})$ over $\mathbb{Z}_{q^2}$, such that $\omega = \zeta \bmod q$. By iterating this procedure until the roots for a sufficiently large modulus M are obtained, we can indeed utilize the NTT for evaluating $f(x) \in \mathbb{Z}_M[x]/(x^d+1)$ on the primitive $2d$-th roots of unity.
- The offline MaskedContGauss gadget can be implemented using an NI-secure CDT sampling gadget that generates a masked sample following a tabulated Gaussian distribution of the 0-center and a fixed standard deviation r. The table values are not sensitive so they are the same as for the unmasked implementation. This masked CDT algorithm was introduced in [5, 20] and proved t-NI. One could instantiate the offline sampling in a different manner, depending on the required precision of samples. One plausible option would be to employ a secure gadget computing the Box–Muller transform [8]. Since

Algorithm 6: GaussShareByShare$_r$: Share-by-share discrete Gaussian sampling

Input: $t+1$-shared center $\langle\!\langle c \rangle\!\rangle_m \in ((1/C \cdot \mathbb{Z})/m\mathbb{Z})^{t+1}$ for fixed integer C and m; a Gaussian parameter r
Result: $t+1$-shared discrete Gaussian sample $[\![z]\!]_m \in (\mathbb{Z}/m\mathbb{Z})^{t+1}$, where z follows $D_{\mathbb{Z},c,r} \bmod m$

1 $(c_1, \ldots, c_{t+1}) \leftarrow \langle\!\langle c \rangle\!\rangle_m$
2 **for** $i \in [1, t+1]$ **do**
3 $\quad z_i \leftarrow D_{\mathbb{Z},c_i,r/\sqrt{t+1}}$
4 **end for**
5 $[\![z]\!]_m \leftarrow (z_1, \ldots, z_{t+1})$
6 **return** $[\![z]\!]_m$

masking the Box–Muller transform amounts to evaluating a few non-linear functions (i.e. $\cos, \sin, \log, \sqrt{\cdot}$) on random Boolean shares representing values in $\mathbb{R} \cap [0,1]$, one could easily achieve such a gadget using the existing polynomial approximation techniques [5].

5.3 Masking security proof

We are able to prove that both masked samplers meet the standard security notion (t-NIo) for masked signature schemes.

Theorem 3. *Assuming t-NI security of* GaussShareByShare, MatMul *and* MaskedContGauss, *and t-NIo security of* Unmask, *the* MaskedRingPeikert *sampler (Alg. 7) is t-NIo secure with public output* $\mathbf{z}$.

Proof. Let us assume that an attacker has access to $\delta \leqslant t$ observations on the whole sampler. Our goal is to prove that all these δ observations can be perfectly simulated with at most δ shares of each secret among $[\![\mathbf{B}]\!]_q$, $[\![q^k \mathbf{B}^{-1}]\!]$, and $[\![q^{k_2} \Sigma_1]\!]$. We consider an attacker who peeks at internal computations as follows: δ_1 observations during line 15; δ_2 observations during line 14; δ_3 observations during line 9-12; δ_4 observations during line 8; δ_5 observations during line 7; δ_6 observations during line 6; δ_7 observations during line 1-4. Suppose $\sum_i \delta_i \leqslant \delta$.

- Since Unmask is t-NIo secure with public output $\mathbf{z}$, all the observations at line 15 can be simulated with at most δ_1 shares of $[\![\mathbf{z}]\!]_q$ and $\mathbf{z}$.
- Since MatMul is t-NI secure, all the observations at line 14 can be simulated with at most $\delta_1 + \delta_2$ shares of $[\![\mathbf{z}]\!]_q$ and $[\![\mathbf{B}]\!]_q$.
- Since line 9-12 consists of independent local operations on each share of $[\![q^k \mathbf{v}]\!]$ and GaussShareByShare is t-NI secure, all the observations during line 9-12 can be simulated with at most $\delta_1 + \delta_2 + \delta_3$ shares of $[\![q^k \mathbf{v}]\!]$.
- Since line 8 consists of linear operations on input shares, all the observations during line 8 can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_4$ shares of $[\![q^k \mathbf{B}^{-1}\mathbf{c}]\!]$ and $[\![q^k \mathbf{x}]\!]$.
- Since MatMul is t-NI secure, all the observations at line 7 can be simulated with at most $\delta_1+\delta_2+\delta_3+\delta_4+\delta_5$ shares of $[\![q^k \mathbf{B}^{-1}]\!]$.
- Since MatMul is t-NI secure, all the observations at line 6 can be simulated with at most $\delta_1+\delta_2+\delta_3+\delta_4+\delta_6$ shares of $[\![q^{k_2} \Sigma_1]\!]$ and $[\![q^{k_1} \mathbf{y}]\!]$.
- Since MaskedContGauss is t-NI secure, all the observations during line 1-4 can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_6 + \delta_7$ shares of randomness as input to MaskedContGauss.

Clearly, the number of total observations for each sensitive input does not exceed δ.

Theorem 4. *Assuming t-NI security of* GaussShareByShare, MatMul, PolyMul *and* MaskedContGauss, *t-SNI security of* RefreshM, *and t-NIo security of* Unmask, *the* MaskedHybrid *sampler (Alg. 8) is t-NIo secure with public output* $\mathbf{v}_0$.

Algorithm 7: MaskedRingPeikert sampler

Input: An arithmetic masking modulus $M = q^{\ell+k}$ such that $k = k_1 + k_2$ and $\ell > 0$; a masked secret bases $[\![\mathbf{B}]\!]_q, [\![q^k\mathbf{B}^{-1}]\!]_{q^{\ell+k}}$ such that $\mathbf{B} \in \mathscr{K}^{2\times 2}$, $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$; a target center $\mathbf{c} \in \mathscr{R}$; a precomputed parameter $r \geqslant \eta_\varepsilon(\mathscr{R}^2)$ and masked matrix $[\![q^{k_2}\Sigma_1]\!]_{q^{\ell+k}}$ such that $\Sigma_1 = \mathbf{B}^{-1}\Sigma_0$, where $\Sigma_0 \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$ is such that $\Sigma_0\Sigma_0^* = \Sigma - r^2\mathbf{B}\mathbf{B}^*$.

Result: $\mathbf{z} \in \mathscr{L}$ with distribution negligibly far from $D_{\mathscr{L}(\mathbf{B}),\mathbf{c},\Sigma}$.

Offline

1 **for** $j \in [0, d-1]$ **do**
2 $\quad [\![q^{k_1}y_{1,j}]\!]_{q^{\ell+k}} \leftarrow \mathsf{MaskedContGauss}(t, q^{\ell+k})$
3 $\quad [\![q^{k_1}y_{2,j}]\!]_{q^{\ell+k}} \leftarrow \mathsf{MaskedContGauss}(t, q^{\ell+k})$
4 **end for**
5 $[\![q^{k_1}\mathbf{y}]\!]_{q^{\ell+k}} := (([\![q^{k_1}y_{1,j}]\!]_{q^{\ell+k}})_{j\in[0,d-1]}, ([\![q^{k_1}y_{2,j}]\!]_{q^{\ell+k}})_{j\in[0,d-1]})$
6 $[\![q^k\mathbf{x}]\!]_{q^{\ell+k}} \leftarrow \mathsf{MatMul}([\![q^{k_2}\Sigma_1]\!]_{q^{\ell+k}}, [\![q^{k_1}\mathbf{y}]\!]_{q^{\ell+k}})$

Online

7 $[\![q^k\mathbf{B}^{-1}\mathbf{c}]\!]_{q^{\ell+k}} \leftarrow \mathsf{MatMul}([\![q^k\mathbf{B}^{-1}]\!]_{q^{\ell+k}}, \mathbf{c})^a$
8 $[\![q^k\mathbf{v}]\!]_{q^{\ell+k}} \leftarrow [\![q^k\mathbf{B}^{-1}\mathbf{c}]\!]_{q^{\ell+k}} - [\![q^k\mathbf{x}]\!]_{q^{\ell+k}}$
9 **for** $j \in [1, d]$ **do**
10 $\quad [\![z_{1,j}]\!]_{q^\ell} \leftarrow \mathsf{GaussShareByShare}_r\left(\frac{[\![q^k v_{1,j}]\!]_{q^{\ell+k}}}{q^k}\right)$
11 $\quad [\![z_{2,j}]\!]_{q^\ell} \leftarrow \mathsf{GaussShareByShare}_r\left(\frac{[\![q^k v_{2,j}]\!]_{q^{\ell+k}}}{q^k}\right)$
12 **end for**
13 $[\![\mathbf{z}]\!]_q := [\![\mathbf{z}]\!]_{q^\ell}$ /* compute every share mod q */
14 $[\![\mathbf{z}]\!]_q \leftarrow \mathsf{MatMul}([\![\mathbf{B}]\!]_q, [\![\mathbf{z}]\!]_q)$
15 $\mathbf{z} \leftarrow \mathsf{Unmask}([\![\mathbf{z}]\!]_q)$
16 **return** $\mathbf{z}$

[a] As the input center is not sensitive this is in practice a simpler variant of MatMul where the second input is unmasked.

Proof. Below we omit the subscripts of masked variables denoting modulus for the sake of readability. Let us assume that an attacker has access to $\delta \leqslant t$ observations on the whole sampler. Our goal is to prove that all these δ observations can be perfectly simulated with at most δ shares of each secret among $[\![\mathbf{b}_1]\!]$, $[\![\mathbf{b}_2]\!]$, $[\![q^{k_2}\sigma_1]\!]$, $[\![q^{k_2}\sigma_2]\!]$, $[\![q^k\boldsymbol{\beta}_1]\!]$ and $[\![q^k\boldsymbol{\beta}_2]\!]$. We consider an attacker who peeks at internal computations as follows: δ_1 at line 25; δ_2 during addition at line 24; δ_3 during MatMul at line 24; δ_4 at line 23; δ_5 during line 19-21; δ_6 at line 18; δ_7 during addition at line 17; δ_8 during first PolyMul at line 17; δ_9 during second PolyMul at line 17; δ_{10} at line 15; δ_{11} during line 12-14; δ_{12} at line 11; δ_{13} during addition at line 10; δ_{14} during first PolyMul at line 10; δ_{15} during second PolyMul at line 10; δ_{16} at line 7; δ_{17} at line 8; δ_{18} during line 1-4; Suppose $\sum_i \delta_i \leqslant \delta$.

- Since Unmask is t-NIo secure with public output $\mathbf{v}_0$, the observations during line 25 can be simulated with at most δ_1 shares of $[\![\mathbf{v}_0]\!]$ and $\mathbf{v}_0$.
- Due to the linear operations, the observations during addition at line 24 can be simulated with at most $\delta_1 + \delta_2$ shares of $[\![\mathbf{v}_1']\!]$ and $[\![x_1\mathbf{b}_1]\!]$.
- Since MatMul is t-NI secure, the observations during MatMul at line 24 can be simulated with at most $\delta_1 + \delta_2 + \delta_3$ shares of $[\![x_1]\!]$ and $[\![\mathbf{b}_1]\!]$.
- Since RefreshM is t-SNI secure, the observations during line 23 can be simulated with at most δ_4 shares of $[\![\mathbf{v}_1]\!]$.
- Since line 19-21 consists of independent local operations on each share of $[\![q^k z_1]\!]$ and GaussShareByShare is t-NI secure, all the observations during line 19-21 can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_5$ shares of $[\![q^k z_1]\!]$.

- Due to the linear operations, the observations during addition at line 18 can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_5 + \delta_6$ shares of $[\![q^k d_1]\!]$ and $[\![q^k y_1]\!]$.
- Due to the combination of linear operations and t-NI secure PolyMul, the observations during line 17 can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_5 + \delta_6 + \delta_7 + \delta_8$ shares of $[\![q^k \beta_{1,1}]\!]$ and $[\![c_{1,1}]\!]$, and $\delta_1 + \delta_2 + \delta_3 + \delta_5 + \delta_6 + \delta_7 + \delta_9$ shares of $[\![q^k \beta_{1,2}]\!]$ and $[\![c_{1,2}]\!]$.
- Since MatMul is t-NI secure, the observations at line 15 can be simulated with at most $\sum_{i\in[1,10]} \delta_i$ shares of $[\![x_2]\!]$ and $[\![\mathbf{b}_2]\!]$.
- Since line 12-14 consists of independent local operations on each share of $[\![q^k z_2]\!]$ and GaussShareByShare is t-NI secure, all the observations during line 12-14 can be simulated with at most $\sum_{i\in[1,11]} \delta_i$ shares of $[\![q^k z_2]\!]$.
- Due to the linear operations, the observations during addition at line 11 can be simulated with at most $\sum_{i\in[1,12]} \delta_i$ shares of $[\![q^k d_2]\!]$ and $[\![q^k y_2]\!]$.
- Due to the combination of linear operations and t-NI secure PolyMul, the observations during line 10 can be simulated with at most $\sum_{i\in[1,14]} \delta_i$ shares of $[\![q^k \beta_{2,1}]\!]$, and $\sum_{i\in[1,13]} \delta_i + \delta_{15}$ shares of $[\![q^k \beta_{2,2}]\!]$.
- Since PolyMul is t-NI secure, the observations during PolyMul at line 7 (resp. line 8) can be simulated with at most $\delta_1 + \delta_2 + \delta_3 + \delta_5 + \delta_6 + \delta_{16}$ shares of $[\![q^{k_2} \sigma_1]\!]$ and $[\![q^{k_1} u_1]\!]$ (resp. $\sum_{i\in[1,12]} \delta_i + \delta_{17}$ shares of $[\![q^{k_2} \sigma_2]\!]$ and $[\![q^{k_1} u_2]\!]$).
- Since MaskedContGauss is t-NI secure, all the observations during line 1-4 can be simulated with at most $\sum_{i\in[1,12]} \delta_i + \delta_{16} + \delta_{17} + \delta_{18}$ shares of randomness as input to MaskedContGauss.

Clearly, the number of total observations for each sensitive input does not exceed δ.

References

[1] Abboud, M., Prest, T.: Cryptographic divergences: New techniques and new applications. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. LNCS, vol. 12238, pp. 492–511. Springer, Heidelberg (Sep 2020)

[2] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: Holz, T., Savage, S. (eds.) USENIX Security 2016. pp. 327–343. USENIX Association (Aug 2016)

[3] Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.A., Grégoire, B., Strub, P.Y., Zucchini, R.: Strong non-interference and type-directed higher-order masking. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 116–129. ACM Press (Oct 2016)

[4] Barthe, G., Belaïd, S., Espitau, T., Fouque, P.A., Grégoire, B., Rossi, M., Tibouchi, M.: Masking the GLP lattice-based signature scheme at any order. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 354–384. Springer, Heidelberg (Apr / May 2018)

[5] Barthe, G., Belaïd, S., Espitau, T., Fouque, P.A., Rossi, M., Tibouchi, M.: GALACTICS: Gaussian sampling for lattice-based constant- time implementation of cryptographic signatures, revisited. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2147–2164. ACM Press (Nov 2019)

[6] Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Krauthgamer, R. (ed.) 27th SODA. pp. 10–24. ACM-SIAM (Jan 2016)

[7] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (Dec 2011)

[8] Box, G.E.P., Muller, M.E.: A Note on the Generation of Random Normal Deviates. The Annals of Mathematical Statistics 29(2), 610 – 611 (1958), https://doi.org/10.1214/aoms/1177706645

[9] 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)

[10] Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y., Kannwischer, M., Patarin, J.: Rainbow. Tech. rep., National Institute of Standards and Technology (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[11] Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 05. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (Jun 2005)
[12] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal Gaussians. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 40–56. Springer, Heidelberg (Aug 2013)
[13] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR TCHES 2018(1), 238–268 (2018), https://tches.iacr.org/index.php/TCHES/article/view/839
[14] 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, Heidelberg (Dec 2014)
[15] 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, Heidelberg (Dec 2012)
[16] Ducas, L., Prest, T.: Fast Fourier orthogonalization. Cryptology ePrint Archive, Report 2015/1014 (2015), https://eprint.iacr.org/2015/1014
[17] Espitau, T., Kirchner, P.: The nearest-colattice algorithm. IACR Cryptol. ePrint Arch. (2020)
[18] 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, Heidelberg (May 2020)
[19] 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)
[20] Gérard, F., Rossi, M.: An efficient and provable masked implementation of qtesla. In: Belaïd, S., Güneysu, T. (eds.) Smart Card Research and Advanced Applications - 18th International Conference, CARDIS 2019, Prague, Czech Republic, November 11-13, 2019, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11833, pp. 74–91. Springer (2019)
[21] 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, Heidelberg (Aug 1997)
[22] Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: Performance improvements and a baseline parameter generation algorithm for NTRUSign. Cryptology ePrint Archive, Report 2005/274 (2005), https://eprint.iacr.org/2005/274
[23] 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, Heidelberg (Apr 2003)
[24] 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, Heidelberg (2020)
[25] Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 150–169. Springer, Heidelberg (Aug 2007)
[26] Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (Aug 2003)
[27] Kirchner, P., Espitau, T., Fouque, P.A.: Fast reduction of algebraic lattices over cyclotomic fields. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 155–185. Springer, Heidelberg (Aug 2020)
[28] Kirchner, P., Fouque, P.A.: Revisiting lattice attacks on overstretched NTRU parameters. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 3–26. Springer, Heidelberg (Apr / May 2017)
[29] Laarhoven, T.: Randomized lattice sieving for the closest vector problem (with preprocessing). Cryptology ePrint Archive, Report 2016/888 (2016), https://eprint.iacr.org/2016/888
[30] Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. 75(3), 565–599 (2015)
[31] Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-DILITHIUM. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions

[32] 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 (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[33] Lyubashevsky, V., Wichs, D.: Simple lattice trapdoor sampling from a broad class of distributions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 716–730. Springer, Heidelberg (Mar / Apr 2015)

[34] Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg (Aug 2013)

[35] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007)

[36] Micciancio, D., Walter, M.: Gaussian sampling over the integers: Efficient, generic, constant-time. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 455–485. Springer, Heidelberg (Aug 2017)

[37] Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. Journal of Cryptology 22(2), 139–160 (Apr 2009)

[38] Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80–97. Springer, Heidelberg (Aug 2010)

[39] Pornin, T.: New efficient, constant-time implementations of Falcon. Cryptology ePrint Archive, Report 2019/893 (2019), https://eprint.iacr.org/2019/893

[40] Pornin, T., Prest, T.: More efficient algorithms for the NTRU key generation using the field norm. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 504–533. Springer, Heidelberg (Apr 2019)

[41] Prest, T.: Gaussian Sampling in Lattice-Based Cryptography. Ph.D. thesis, École Normale Supérieure, Paris, France (2015)

[42] Prest, T.: Sharper bounds in lattice-based cryptography using the Rényi divergence. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 347–374. Springer, Heidelberg (Dec 2017)

[43] 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 (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[44] Yu, Y., Ducas, L.: Learning strikes again: The case of the DRS signature scheme. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 525–543. Springer, Heidelberg (Dec 2018)

A Concentration bounds for Gamma distributions

Let $X \sim \Gamma(\alpha, \beta)$ be a Gamma-distributed random variable. Its logarithmic moment generating function is as follows for all $\lambda < \beta$:

$$\begin{aligned}
\Phi_X(\lambda) &= \log \mathbb{E}\big[e^{\lambda X}\big] \\
&= \log \int_0^{+\infty} e^{\lambda x} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} dx \\
&= \log \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^{+\infty} x^{\alpha-1} e^{(\lambda-\beta)x} dx \right) \\
&= \log \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^{+\infty} x^{\alpha-1} e^{(\lambda-\beta)x} dx \right) \\
&= \log \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^{+\infty} \frac{z^{\alpha-1}}{(\beta-\lambda)^{\alpha-1}} e^{-z} \frac{dz}{\beta-\lambda} \right) \\
&= \log \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \cdot \frac{\Gamma(\alpha)}{(\beta-\lambda)^\alpha} \right) \\
&= -\alpha \log \Big(1 - \frac{\lambda}{\beta}\Big).
\end{aligned}$$

In particular, $\Phi'_X(\lambda) = \frac{\alpha}{\beta-\lambda}$, and thus the map $\lambda \mapsto t\lambda - \Phi_X(\lambda)$ is maximal at λ_0 such that:

$$t = \frac{\alpha}{\beta - \lambda_0}, \quad \text{namely} \quad \lambda_0 = \beta - \frac{\alpha}{t}$$

(assuming $t > 0$; otherwise the function is unbounded on $(-\infty, \beta)$). Therefore, the Cramér transform of X is given by:

$$\Phi^*(t) = \sup_{\lambda<1} \big(t\lambda - \Phi_X(\lambda)\big) = t\lambda_0 - \Phi_X(\lambda_0) = \beta t - \alpha - \alpha \log \frac{\beta t}{\alpha}$$

for $t > 0$. Noting that $\mathbb{E}[X] = \Phi'_X(0) = \alpha/\beta$, the Cramér–Chernoff inequality then yields:

$$\Pr[X > t] \leqslant e^{\alpha-\beta t} \Big(\frac{\beta t}{\alpha}\Big)^\alpha \leqslant \Big(\frac{e\beta t}{\alpha}\Big)^\alpha \qquad \text{for } t > \frac{\alpha}{\beta}, \tag{1}$$

$$\Pr[X < t] \leqslant e^{\alpha-\beta t} \Big(\frac{\beta t}{\alpha}\Big)^\alpha \leqslant (2e)^\alpha e^{-\beta t/2} \qquad \text{for } 0 < t < \frac{\alpha}{\beta}, \tag{2}$$

where the second inequalities in the two cases follow from $e^{-\beta t} \leqslant 1$ and $\beta t/\alpha \leqslant 2e^{\beta t/2\alpha}$ respectively.

Now we consider $X_1, \ldots, X_k \sim \Gamma(\alpha, \beta)$ independent gamma-distributed random variables, and want to estimate the tail bound on $\max(X_1, \ldots, X_k, \gamma/X_1, \ldots, \gamma/X_k)$ for some constant $\gamma > 0$.

On the one hand, by the union bound and Eq. (2), we have, for $0 < \gamma/t < \alpha/\beta$:

$$\Pr\Big[\max\big(\frac{\gamma}{X_1}, \ldots, \frac{\gamma}{X_k}\big) > t\Big] = \Pr\Big[\max(X_1, \ldots, X_k) > \frac{\gamma}{t}\Big] \leqslant k\Big(\frac{e\beta\gamma}{\alpha t}\Big)^\alpha.$$

In particular, we have $\Pr[\max(\gamma/X_1, \ldots, \gamma/X_k) > t] \leqslant \delta_{\min}$ as soon as:

$$\begin{aligned}
& k\Big(\frac{e\beta\gamma}{\alpha t}\Big)^\alpha \leqslant \delta_{\min} \\
\iff \quad & \frac{e\beta\gamma}{\alpha t} \leqslant \Big(\frac{\delta_{\min}}{k}\Big)^{1/\alpha} \\
\iff \quad & t \geqslant \frac{e\beta\gamma}{\alpha}\Big(\frac{k}{\delta_{\min}}\Big)^{1/\alpha}.
\end{aligned}$$

On the other hand, by the union bound and Eq. (1), we also have, for $t > \alpha/\beta$:

$$\Pr[\max(X_1, \ldots, X_k) > t] \leqslant k(2e)^\alpha e^{-\beta t/2}.$$

In particular, we have $\Pr[\max(X_1, \ldots, X_k) > t] \leqslant \delta_{\max}$ as soon as:

$$\begin{aligned} & k(2e)^\alpha e^{-\beta t/2} \leqslant \delta_{\max} \\ \iff \quad & \alpha \log(2e) - \frac{\beta}{2} t \leqslant -\log \frac{k}{\delta_{\max}} \\ \iff \quad & t \geqslant \frac{2}{\beta}\Big(\log \frac{k}{\delta_{\max}} + \alpha \log(2e)\Big). \end{aligned}$$

This results in the following theorem.

Theorem 5. *Let $X_1, \ldots, X_k \sim \Gamma(\alpha, \beta)$ independent gamma-distributed random variables, and fix $\gamma > 0$, $\delta_{\min} \in (0, 1)$ and $\delta_{\max} \in (0, 1)$. We have:*

$$\Pr\left[\max\left(X_1, \ldots, X_k, \frac{\gamma}{X_1}, \ldots, \frac{\gamma}{X_k}\right) < \max(t_{\min}, t_{\max})\right] \leqslant \delta_{\min} + \delta_{\max}$$

where:

$$t_{\min} = \frac{e\beta\gamma}{\alpha}\Big(\frac{k}{\delta_{\min}}\Big)^{1/\alpha} \quad \textit{and} \quad t_{\max} = \frac{2}{\beta}\Big(\log \frac{k}{\delta_{\max}} + \alpha \log(2e)\Big).$$

This simply follows from the union bound applied to the previous inequalities, together with the observation that $t_{\min}$ and $t_{\max}$ always satisfy the required bounds $0 < \gamma/t_{\min} < \alpha/\beta$ and $t_{\max} > \alpha/\beta$.

We can then apply this result to estimate the quality of the hybrid sampler obtained from an NTRU basis $\mathbf{B}_{f,g}$, which is given by the norm $|\mathbf{B}_{f,g}|_{\mathscr{K}}$, where (as recalled in Lemma 1):

$$|\mathbf{B}_{f,g}|_{\mathscr{K}}^2 = \max\left(\|\varphi(ff^* + gg^*)\|_\infty, \left\|\frac{q^2}{\varphi(ff^* + gg^*)}\right\|_\infty\right). \tag{3}$$

The ring elements f, g have all their coefficients sampled independently according to the centered discrete Gaussian of standard deviation $\sigma_0 = \sqrt{\frac{\nu q}{4d}}$ for some $\nu > 0$. Then, all the embeddings $\varphi_i(f), \varphi_i(g) \in \mathbb{C}$ are (statistically close to) 2-dimensional discrete Gaussian of standard deviation $\sqrt{\nu q/4}$, which we heuristically assume behave like normal vectors of the same standard deviation.[10] Then, the $\varphi_i(ff^*), \varphi_i(gg^*)$ $(1 \leqslant i \leqslant d/2)$ are independent scaled $\chi^2(2)$ distributed random variables, or equivalently $\Gamma\big(1, \frac{2}{\nu q}\big)$-distributed random variables. As a result, the $X_i = \varphi_i(ff^* + gg^*)$ $(1 \leqslant i \leqslant d/2)$ are independent $\Gamma\big(2, \frac{2}{\nu q}\big)$-distributed random variables. Now, by Eq. (3), we have:

$$|\mathbf{B}_{f,g}|_{\mathscr{K}}^2 = \max\big(X_1, \ldots, X_{d/2}, \frac{q^2}{X_1}, \ldots, \frac{q^2}{X_{d/2}}\big),$$

which is of the form considered in Theorem 5 with $\alpha = 2$, $\beta = \frac{2}{\nu q}$, $\gamma = q^2$ and $k = d/2$. Therefore, it follows that for any choice of $\delta_{\min}, \delta_{\max} \in (0, 1)$, $\Pr\left[|\mathbf{B}_{f,g}|_{\mathscr{K}}^2 < \max(t_{\min}, t_{\max})\right] \leqslant \delta_{\min} + \delta_{\max}$, where:

$$t_{\min} = \frac{eq}{\nu}\sqrt{\frac{d}{2\delta_{\min}}} \quad \text{and} \quad t_{\max} = \nu q \log \frac{2e^2 d}{\delta_{\max}}.$$

We can moreover choose ν in key generation in such a way that $t_{\min} = t_{\max}$, by setting:

$$\nu = \sqrt{\frac{e}{\log(2e^2 d/\delta_{\max})}} \cdot \sqrt[4]{\frac{d}{2\delta_{\min}}}.$$

[10] We can do away with this heuristic assumption in the analysis by generalizing Theorem 5 to sub-Gamma random variables.

Picking e.g. $\delta_{\max} = \delta_{\min} = 1/4$, we get $\Pr[|\mathbf{B}_{f,g}|^2_{\mathscr{K}} < t] \leqslant 1/2$, where:

$$t = q\sqrt{e\log(8e^2 d)}\sqrt[4]{2d} = O(q \cdot d^{1/4}\log^{1/2} d).$$

Thus, the quality of the hybrid sampler scales as $\widetilde{O}(d^{1/8})$ (and not $\sqrt{\log d}$ as incorrectly claimed in [41]: the error was due to the overly optimistic assumption that the $q^2/\varphi_i(ff^* + gg^*)$ components had the same tail behavior as the $\varphi_i(ff^* + gg^*)$, which is not the case, since they are Inverse-Gamma distributed, and thus not sub-Gamma; in fact, they do not even have finite variance).

B Finer-grained selection parameter using cyclotomic fields of composite conductors

As mentioned earlier, a strength of the Mitaka scheme lies in the possibility to instantiate it over any number fields $\mathscr{K}$. To remain practically competitive, one needs to be sable to sample efficiently discrete Gaussian over its ring of integers $\mathcal{R} = \mathcal{O}_{\mathscr{K}}$. We saw that on cyclotomic of conductor 2^n, this is trivially the case as the power basis is orthogonal; and as a result sampling over $\mathcal{R}$ boils down to perform a coefficient wise sampling. In this section, we show that we can efficiently sample in cyclotomic fields of *smooth* enough conductor.

B.1 Geometry of the power basis

Let $\mathbb{Q}(\zeta_m)$ be the cyclotomic field of conductor $m = p^a 2^b$ for an odd prime number p and nonnegative integers a, b. Set $\mathbf{B} = (1, \zeta_m, \ldots, \zeta_m^{\varphi(m)-1})$. As $\mathbb{Q}(\zeta_m)$ is the compositum of the prime-power cyclotomic fields $\mathbb{Q}(\zeta_{2^a}$ and $\mathbb{Q}(\zeta_{2^b}$, its ring of integers is the tensor product of the ring of integers of these two fields, so that a routine computation ensures that the Gram-matrix of $\mathbf{B}$ in the canonical embedding is:

$$\frac{\varphi(m)}{p-1} G(p,b) \otimes \mathrm{Id}_{\frac{\varphi(m)}{p-1}},$$

where

$$G(p,b) = \begin{cases} \mathrm{Circ}_{p-1}(p-1,-1,\ldots,-1) & \text{if } b = 0 \\ \mathrm{Circ}_{p-1}(p-1,1,-1,\ldots,-1,1) & \text{if } b > 0 \end{cases},$$

for $\mathrm{Circ}_{p-1}(X)$ designating the circulant matrix of size $p-1 \times p-1$ and coefficients following the $(p-1)$-uple X. As an illustration, for the cyclotomic field of conductor $20 = 2^2 \times 5$, we have:

$$G = 2\mathrm{Circ}_4(4,1,-1,1) \otimes \mathrm{Id}_2$$
$$= \begin{pmatrix} 8 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & 8 & 0 & 2 & 0 & -2 & 0 & 2 \\ 2 & 0 & 8 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & 8 & 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 & 8 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 & 0 & 8 & 0 & 2 \\ 2 & 0 & -2 & 0 & 2 & 0 & 8 & 0 \\ 0 & 2 & 0 & -2 & 0 & 2 & 0 & 8 \end{pmatrix}$$

B.2 Sampling

As a consequence, we can use the hybrid sampler to reduce the sampling over $\mathcal{R}$ to a module sampling over a module of rank $p-1$, which Gram-matrix is $G(p,B)$, defined over a subring isometric to $\frac{\varphi(m)}{p-1}\mathbb{Z}^{\frac{\varphi(m)}{p-1}}$. Let us study this matrix in more details. Its principal minor G_i of order i is the circulant matrix $\mathrm{Circ}_i(p-1,-1,\ldots,-1)$. Elementary theory of Toeplitz-like matrices ensures that this minor have for spectrum:

$$\mathrm{Sp}(G_i) = \{p-i, \underbrace{p,\ldots,p}_{i-1-\text{times}}\},$$

so that its determinant is $\det(G_i) = (p-i)p^{i-1}$. Henceforth a direct induction ensures that the (square of the) diagonal of the Cholesky decomposition of $G(p,b)$ is

$$\left[p-1, \frac{(p-2)p}{p-1}, \frac{(p-3)p}{(p-2)}, \ldots, \frac{p}{2}\right],$$

which is a decreasing sequence. Hence the maximum value of the diagonal elements of the Cholesky decomposition of $G(p,b)$ is $\sqrt{p-1}$. As it is also the norm of the corresponding Gram-Schmidt vectors. As such, the hybrid sampler induces an additional $\sqrt{p-1}$ compared to the sampling over the power-of-two cyclotomic fields.

B.3 Practical impact on the parameter selction

Using this analysis, we get a wider range of parameters for instantiating the MITAKA scheme. In particular, the 3-smooth conductors are of particular interest as they only induce a loss of a factor $\sqrt{2}$ in the sampler quality and allows sampling which is asymptotically as fast as the sampling in power-of-two conductors, using the hybrid sampler. In Fig. 3, we show the impact of the base field on the bit-security with regards to the parameter α.

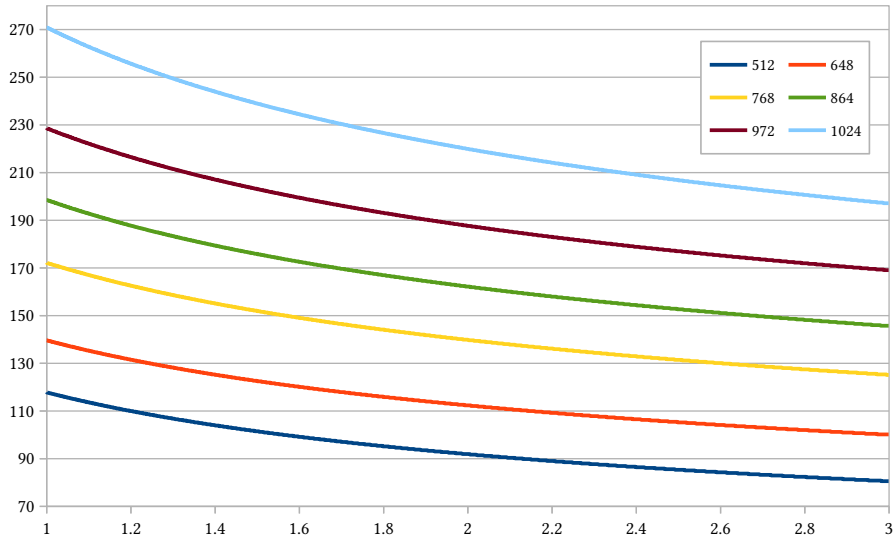


Fig. 3. Security of the MITAKA scheme for different choices of cyclotomic fields with 3−smooth conductors.

C On the security of MITAKA

In all of the following, we follow the so-called *Geometric series assumption* (GSA), asserting that a reduced basis sees its Gram-Schmidt vectors' norm decrease with geometric decay. More formally, it can be instantiated as follows for self-dual BKZ (DBKZ) reduction algorithm of Micciancio and Walter [36]: an output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ yielded by DBKZ algorithm with block size β on a lattice $\mathscr{L}$ of rank n satisfies

$$\|\widetilde{\mathbf{b}}_i\| = \delta_\beta^{n-2(i-1)} \operatorname{covol}(\mathscr{L})^{\frac{1}{n}}, \quad \text{where} \quad \delta_\beta = \left(\frac{(\pi\beta)^{\frac{1}{\beta}} \cdot \beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}.$$

C.1 Key recovery attack

The key recovery consists in finding the private secret key (i.e. $f, g \in \mathcal{R}^2$) from the sole data of the public elements q and h. The most powerful attacks are up-to-our-knowledge realized through lattice reduction. It consists in constructing the algebraic lattice over $\mathcal{R}$ spanned by the vectors $(q, 0)$ and $(h, 1)$ (i.e. the public basis of the NTRU key) and retrieve the lattice vector $\mathbf{s} = (\mathbf{g}, \mathbf{f})$ among all possible lattice vectors of norm bounded by $\|\mathbf{s}\| = \sqrt{2n}\sigma$. We make use of the so-called *projection trick* to avoid enumerating over all this sphere. More precisely we proceed as follows. Set β to be the block size parameter of the DBKZ algorithm and start by reducing the public basis with this latter algorithm. Call $[\mathbf{b}_1, \ldots, \mathbf{b}_{2n}]$ the resulting vectors. Then if we can recover the *projection* of the secret key onto $\mathcal{P}$, the orthogonal space to $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{2n-\beta-1})$, then we can retrieve in polynomial time the full key by *Babai nearest plane* algorithm to lift it to a lattice vector of the desired norm. Hence it suffices to be able find the projection of the secret key among the shortest vector of the lattice generated by the last β vectors projected onto $\mathcal{P}$. Classically, sieving on this projected lattice will recover all vectors of norm smaller than $\sqrt{\frac{4}{3}}\ell$, where ℓ is the norm of the $2n-\beta$-th Gram-Schmidt vector $\widetilde{b}_{2m-\beta}$ of the reduced basis. Under the GSA assumption we have:

$$\ell = \sqrt{q}\delta_\beta^{-2n+2\beta+2} \approx \left(\frac{\beta}{2\pi e}\right)^{1-\frac{n}{\beta}}.$$

Moreover, considering that $\mathbf{s}$ behaves as a random vector of norm $\sqrt{2n}\sigma$, and using the GSA to bound the norm of the Gram-Schmidt vectors $[\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{2n-\beta}]$, that the norm of its projection over $\mathcal{P}$ is roughly

$$\sqrt{\frac{\beta}{2n}}\|\mathbf{s}\| = \beta^{\frac{1}{2}}\sigma.$$

Hence, we will retrieve the projection among the sieved vectors if $\beta^{\frac{1}{2}}\sigma \leqslant \sqrt{\frac{4}{3}}\ell$, that is if the following condition is fulfilled:

$$\sigma^2 \leqslant \frac{4q}{3\beta}\delta_\beta^{4(\beta+1-n)} \tag{4}$$

C.2 Signature forgery by ApproxCVP reduction.

As a Hash-and-Sign paradigm signature, forging a signature stems to feeding a lattice point $\mathbf{v}$ at a bounded distance from a random space point $\mathbf{x}$. This ApproxCVP problem can be solved using the so-called *Nearest-Cospace* framework developed in [17]. Under the Geometric Series assumption, Theorem 3.3 of [17] states that under the condition: $\|\mathbf{x} - \mathbf{v}\| \leqslant \left(\delta_\beta^{2n} q^{\frac{1}{2}}\right)$, the decoding can be done in time $\mathrm{Poly}(n)$ calls to a CVP oracle in dimension β.

As mentioned in [9] a standard optimization of this attack consists only considering the lattice spanned by a subset of the vectors of the public basis and perform the decoding within this sublattice. The only interesting subset seems to consists in forgetting the $k \leqslant n$ first vectors. The dimension is of course reduced by k, at the cost of working with a lattice with covolume $q^{\frac{k}{2(2n-k)}}$ bigger. Henceforth the global condition of decoding becomes the (slightly more general) inequality:

$$\|\mathbf{x} - \mathbf{v}\| \leqslant \min_{k \leqslant n}\left(\delta_\beta^{2n-k} q^{\frac{n}{2n-k}}\right) \tag{5}$$

C.3 On the other attacks on Mitaka

In this section, we list the other possible type of attacks on the signature, which are nonetheless irrelevant for the set of parameters we are using.

C.3.1 Algebraic attacks As remarked in the design of NTRU-based schemes (such as for instance Falcon or ModFalcon signatures), there exists a rich algebraic structure in the modules over the convolution ring $\mathcal{R}$ used in Mitaka. However, there is no known way to improve all the algorithms previously mentioned with respect to their general lattice equivalent by more than polynomial factors (see for instance the speedup on lattice reduction of [27].

C.3.2 Overstretched NTRU-type As observed in [28], when the modulus q is significantly larger than the magnitudes of the NTRU secret key coefficients, the attack on the key based on lattice reduction recovers the secret key better than the results presented above. This so-called "overstretched NTRU" parameters occurs when $q > n^{2.83}$ for binary secrets, implying that, as it is the case for Falcon and other NTRU based NIST candidates, that even *very* significant improvements of this attack would still be irrelevant for the security of the scheme.

C.3.3 Hybrid attacks Odlyzko's meet on the middle attack, or more recently the hybrid attack of Howgrave-Graham [25] which combines a meet-in-the-middle algorithm with a key recovery by lattice reduction were used effectively against NTRU, mainly due to its design using sparse polynomials. As it is not the case (secrets are dense elements in the ring $\mathcal{R}$), their impact is not sufficient to be a problem on the parameter selection of MITAKA.

D Security arguments with Renyi divergence

Let $\mathscr{D}, \mathscr{D}'$ be two distributions sharing the same support. Their relative error is defined as $\Delta_{RE}(\mathscr{D}, \mathscr{D}') = \sup_{\mathbf{x} \in \mathrm{Supp}(\mathscr{D})} \left| \frac{\mathscr{D}(\mathbf{x})}{\mathscr{D}'(\mathbf{x})} - 1 \right|$.

Lemma 3 ([42], adapted). *Assume that for two distributions $\mathscr{D}, \mathscr{D}'$ with the same support, we have $\Delta_{RE}(\mathscr{D}, \mathscr{D}') \leqslant \delta$ for some $\delta > 0$. Then we have*

$$R_{2\lambda}(\mathscr{D} \| \mathscr{D}') \leqslant \left(1 + \frac{\lambda(2\lambda - 1)\delta^2}{(1-\delta)^{2\lambda+1}}\right)^{\frac{1}{2\lambda-1}}.$$

Additionally, if $\lambda \in \{128, 256\}$ and $\delta < 2^{-10}$, then:

$$R_{2\lambda}(\mathscr{D} \| \mathscr{D}') \leqslant 1 + 2\lambda\delta^2.$$

In practice, the parameter δ is quite smaller than the bound in the statement (which ensures the correctness of the second inequality). As argued in [42], when trying to solve a search problem, one cannot distinguish if one queried $\mathscr{D}$ or $\mathscr{D}'$ up to Q times as long as $2\lambda\delta^2 \leqslant (4Q)^{-1}$. In practice for signature algorithms, it is often the case that $Q = 2^{64}$ and λ is a target security level, e.g. 128 or 256. The value for Q can be larger if rejection sampling happens, as the target sampler will definitely be queried more time. Contrary to, say, BLISS, this does not happen for the lattice samplers considered in this work.

E Precision analysis for our samplers

We start this appendix with additional definitions and facts.

E.1 Smoothing parameter for general covariance parameters

Following [38], we extend the use of the smoothing parameter to covariance matrices: we say that $\sqrt{\Sigma} \geqslant \eta_\varepsilon(\mathscr{L})$ when $\rho_1(\sqrt{\Sigma}^* \mathscr{L}^\vee) = \rho_{\Sigma^{-1}}(\mathscr{L}^\vee) \leqslant 1 + \varepsilon$. In particular, one checks that $r\mathbf{B} \geqslant \eta_\varepsilon(\varphi(\mathbf{B}\mathscr{R}^2))$ when $r \geqslant \eta_\varepsilon(\mathscr{R}^2)$. The next lemmata are standards.

Lemma 4 ([35], implicit in Lemma 4.4). *Let $\mathscr{L}$ be a rank d lattice, and $\Sigma \succ 0$ such that $\sqrt{\Sigma} \geqslant \eta_\varepsilon(\mathscr{L})$. Then we have $\rho_{\mathbf{c},\Sigma}(\mathscr{L}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}; 1\right] \cdot \rho_\Sigma(\mathscr{L})$.*

Lemma 5 ([35]). *For any d dimensional lattice $\mathscr{L}$, any $\mathbf{c}$ and unit vector $\mathbf{u}$ in $\mathbb{R}^d$, and for all $0 < \varepsilon < 1$ and $r \geqslant 2\eta_\varepsilon(\mathscr{L})$, we have*

$$\left| \mathbb{E}_{\mathbf{x} \leftarrow D_{\mathscr{L},\mathbf{c},r}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u} \rangle] \right| \leqslant \frac{\varepsilon r}{1-\varepsilon}.$$

E.2 Analysis of the samplers

Smoothing parameter and floating point precision for the Peikert sampler

Proof (of Theorem 1). Let $\mathbf{Y}$ be a random variable of distribution $\mathcal{N}_{\Sigma_0\Sigma_0^*}$. If $\mathbf{x} \leftarrow D_{\mathscr{R}^2,\mathbf{B}^{-1}(\mathbf{c}-\mathbf{y}),r^2}$, then by composition $\mathbf{z} := \mathbf{B}\mathbf{x} \leftarrow D_{\mathbf{B}\mathscr{R}^2,\mathbf{c}-\mathbf{y},r^2\mathbf{B}\mathbf{B}^*}$.

Denoting by $\mathtt{Out}$ the output of Algorithm 2, then we have:

$$\begin{aligned}\mathbb{P}[\mathtt{Out} = \mathbf{z} \ \wedge \ \mathbf{Y} = \mathbf{y}] =& \frac{\rho_{\Sigma_0\Sigma_0^*}(\mathbf{y})}{\det \Sigma_0} \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c}-\mathbf{y},r^2\mathbf{B}\mathbf{B}^*}(\mathbf{z}) \\ =& \frac{\rho_{\Sigma_0\Sigma_0^*}(\mathbf{y})}{\det \Sigma_0} \cdot \frac{\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{z}-(\mathbf{c}-\mathbf{y}))}{\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{B}\mathscr{R}^2-(\mathbf{c}-\mathbf{y}))}.\end{aligned}$$

Next we use Fact 2.1 in [Peikert], with the parameters $\mathbf{x} = \mathbf{y}, \mathbf{c}_1 = \mathbf{0}, \mathbf{c}_2 = \mathbf{c}-\mathbf{z}, \Sigma_1 = \Sigma_0\Sigma_0^*, \Sigma_2 = r^2\mathbf{B}\mathbf{B}^*$ and $\Sigma_3 = \Sigma_2\Sigma^{-1}\Sigma_1, \mathbf{c}_3 = \Sigma_3\Sigma_2^{-1}\mathbf{c}_2$ to obtain the identity

$$\rho_{\Sigma_0\Sigma_0^*}(\mathbf{y})\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{z}-(\mathbf{c}-\mathbf{y})) = \rho_{\Sigma}(\mathbf{z}-\mathbf{c})\rho_{\Sigma_3}(\mathbf{y}-\mathbf{c}_3).$$

Set $\mathscr{D}(\mathbf{z})$ the probability of Algorithm 2 to output $\mathbf{z}$. Then:

$$\mathscr{D}(\mathbf{z}) = \frac{\rho_{\Sigma}(\mathbf{z}-\mathbf{c})}{\det \Sigma_0} \int_{\mathbb{R}^n} \frac{\rho_{\Sigma_3}(\mathbf{y}-\mathbf{c}_3)}{\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{B}\mathscr{R}^2-(\mathbf{c}-\mathbf{y}))}\mathrm{d}\mathbf{y}.$$

Because all matrices here are positive definite over $\mathscr{K}_{\mathbb{R}}$ we also have $\det(\Sigma_0^*) = \det(\Sigma_0)$ (it could have been equal to $(\det \Sigma_0)^*$). Note that this gives $\det(\Sigma_3) = \det(r\mathbf{B})^2 \det(\Sigma)^{-1} \det(\Sigma_0)^2$. As we are above $\eta_\varepsilon(\mathbf{B}\mathscr{R}^2)$ in the denominator of the integral, we get

$$\mathscr{D}(\mathbf{z}) \in \left[1, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot \frac{\det(r\mathbf{B})}{(\det \Sigma)^{1/2}} \cdot \frac{\rho_{\Sigma}(\mathbf{z}-\mathbf{c})}{\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{B}\mathscr{R}^2)},$$

By definition of discrete Gaussians, it is equivalent to

$$\mathscr{D}(\mathbf{z}) \in \left[1, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot \alpha \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{z}),$$

where we let $\alpha = \frac{\det(r\mathbf{B})}{(\det \Sigma)^{1/2}} \cdot \frac{\rho_{\Sigma}(\mathbf{B}\mathscr{R}^2-\mathbf{c})}{\rho_{r^2\mathbf{B}\mathbf{B}^*}(\mathbf{B}\mathscr{R}^2)}$. Summing both the left-hand side and right-hand side over all possible $\mathbf{z}$ we see that $\alpha \leqslant 1 \leqslant \frac{1+\varepsilon}{1-\varepsilon}\alpha$, or equivalently, that $\alpha \in [\frac{1-\varepsilon}{1+\varepsilon}, 1]$. We thus obtain

$$\mathscr{D}(\mathbf{z}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{z}). \tag{6}$$

Using that $\varepsilon \leqslant 1/2$, we see that the statistical distance between $\mathscr{D}$ and $D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}$ is bounded by 2ε, as well as

$$\left|\frac{\mathscr{D}(\mathbf{z})}{D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{z})} - 1\right| \leqslant 4\varepsilon.$$

Following the discussion in Appendix D, we see that the Renyi divergence of order 2λ is bounded by $32\lambda\varepsilon^2$. To preserve up to $\lambda = 128$, resp. 256 bits of security for $Q = 2^{64}$ queries, it is then enough to set $\varepsilon \leqslant 2^{-39}$, resp. 2^{-40}.

The following results and inequalities can be found in [41], but we rework them for the sake of diffusion. They are useful for the precision analysis of Algorithm 2.

Lemma 6 ([41], adapted from Lemma 3.10). *Let $k \in \mathbf{N}^*$ and $r > 0$. For fixed $\mathbf{t}, \hat{\mathbf{t}} \in \mathscr{K}_{\mathbb{R}}^k$, let $\psi(\mathbf{x}) := \frac{1}{2r^2}(\|\mathbf{x}-\hat{\mathbf{t}}\|^2 - \|\mathbf{x}-\mathbf{t}\|^2)$. For all $\mathbf{x} \in \mathscr{K}_{\mathbb{R}}^k$, we have $\frac{\rho_r(\mathbf{x}-\mathbf{t})}{\rho_r(\mathbf{x}-\hat{\mathbf{t}})} = \exp(\psi(\mathbf{x}))$, and also:*

$$\exp\left(-\mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\psi(\mathbf{x})]\right) \leqslant \frac{\rho_r(\mathscr{R}^k-\hat{\mathbf{t}})}{\rho_r(\mathscr{R}^k-\mathbf{t})} \leqslant \exp\left(-\mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^2,\hat{\mathbf{t}},r}}[\psi(\mathbf{x})]\right), \tag{7}$$

$$\exp\left(\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\psi(\mathbf{x})]\right) \leqslant \frac{D_{\mathscr{R}^k,\mathbf{t},r}(\mathbf{x})}{D_{\mathscr{R}^k,\hat{\mathbf{t}},r}(\mathbf{x})} \leqslant \exp\left(\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\hat{\mathbf{t}},r}}[\psi(\mathbf{x})]\right). \tag{8}$$

With $\mathbf{v} = \frac{\mathbf{t}-\hat{\mathbf{t}}}{\|\mathbf{t}-\hat{\mathbf{t}}\|}$*, we also have*

$$\left|\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\psi(\mathbf{x})]\right| \leqslant \frac{\|\hat{\mathbf{t}}-\mathbf{t}\|}{r^2}\cdot\left(\|\mathbf{x}-\mathbf{t}\| + \left|\mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\langle \mathbf{x}-\mathbf{t},\mathbf{v}\rangle]\right|\right),$$
$$\left|\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\hat{\mathbf{t}},r}}[\psi(\mathbf{x})]\right| \leqslant \frac{\|\hat{\mathbf{t}}-\mathbf{t}\|}{r^2}\cdot\left(\|\mathbf{x}-\hat{\mathbf{t}}\| + \left|\mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\hat{\mathbf{t}},r}}[\langle \mathbf{x}-\hat{\mathbf{t}},\mathbf{v}\rangle]\right|\right).$$

Proof. The claimed equality amounts to unrolling the definitions. Since $\mathbf{x},\mathbf{t},\hat{\mathbf{t}}$ have all their coordinates in $\mathscr{K}_{\mathbb{R}}$, their hermitian products are all real valued, so that $2r^2\psi(\mathbf{x}) = \|\mathbf{t}\|^2 + \|\hat{\mathbf{t}}\|^2 + 2\langle \mathbf{x}, \mathbf{t}-\hat{\mathbf{t}}\rangle$. In particular, ψ is an affine function of $\mathbf{x}$, and as such is convex, as well as the composition $\exp\circ\psi$. The left-hand side of the first inequality comes from

$$\frac{\rho_r(\mathbf{x}-\hat{\mathbf{t}})}{\rho_r(\mathscr{R}^k-\mathbf{t})} = \exp(-\psi(x))D_{\mathscr{R}^k,\mathbf{t},r}(\mathbf{x}),$$

summing over all $\mathbf{x}$'s and using Jensen's inequality. The right-hand side is obtained mutatis mutandis. The discrete Gaussian version of the inequality then amounts to unrolling the definition of the density function conjointly with Inequality (7) to bound the ratio of the total masses over $\mathscr{R}^k$. By linearity of the expectation, we then have

$$\begin{aligned} r^2\left(\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\psi(\mathbf{x})]\right) &= \langle \mathbf{x},\mathbf{t}-\hat{\mathbf{t}}\rangle - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}[\langle \mathbf{x},\mathbf{t}-\hat{\mathbf{t}}\rangle] \\ &= \langle \mathbf{x}-\mathbf{t},\mathbf{t}-\hat{\mathbf{t}}\rangle - \|\hat{\mathbf{t}}-\mathbf{t}\|\cdot\mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^k,\mathbf{t},r}}\left[\langle \mathbf{x}-\mathbf{t},\mathbf{v}\rangle\right]. \end{aligned}$$

Using Cauchy-Schwarz inequality, we thus obtain the next claim, and the other one follows by observing that the above can also be unrolled for $\mathbf{x}\leftarrow D_{\mathscr{R}^k,\hat{\mathbf{t}},r}$.

We now turn to the precision analysis of Algorithm 1. To make the analysis simpler, we in fact consider its variant in Algorithm 9, where one checks that $\Sigma_1 = \mathbf{B}^{-1}\Sigma_0$ is a valid choice.

Observe that in practice, $\mathbf{c}$ is usually an integer vector (outputted by some hash function with range in $\mathscr{R}^2$) and since $\mathbf{B} = \mathbf{B}_{f,g}$, $q\mathbf{B}^{-1}$ is in $\mathscr{R}^{2\times 2}$. Hence, we may assume that $\mathbf{B}^{-1}\mathbf{c}$ is known exactly. However Σ_0 and vectors sampled from $\mathcal{N}_{1,\mathscr{K}_{\mathbb{R}}^2}$ have real entries and therefore only approximations of their values can be known. In the statement below, one may think of the technical assumptions as analyzing alternative versions of Algorithm 9 which aborts if both the continuous and discrete Gaussian sampler output a large element. Assuming both these samplers are close to perfect (which can be done in practice), Gaussian tail bounds tell us that the probability that both output large elements can be made smaller than $2^{-\lambda}$, where λ is a target security level. Hence, for suitable parameters and by a hybrid argument, it makes no difference to consider such versions.

Proposition 1. *Let $r,\delta,\varepsilon > 0$, and let Σ_1 as in Algorithm 9. For $\mathbf{u}\in\mathscr{K}_{\mathbb{R}}^2$ with $\|\mathbf{u}\|\leqslant 2\sqrt{d}$, let $\mathbf{y} = \Sigma_1\mathbf{u}$. Assume that we are given $\hat{\mathbf{u}},\hat{\Sigma_1}$ and $\hat{\mathbf{y}} = \hat{\Sigma_1}\hat{\mathbf{u}}$ satisfying*

- $\|\mathbf{u}-\hat{\mathbf{u}}\| \leqslant \delta\cdot\|\mathbf{u}\|$;
- $s_1(\Sigma_1 - \hat{\Sigma_1}) \leqslant \delta\cdot s_1(\Sigma_1)$.
- $\max(\|\mathbf{x}-\mathbf{t}\|, \|\mathbf{x}-\hat{\mathbf{t}}\|) \leqslant 2r\sqrt{\pi d}$,

where we let $\mathbf{t} = \mathbf{B}^{-1}\mathbf{c} - \mathbf{y}, \hat{\mathbf{t}} = \mathbf{B}^{-1}\mathbf{c} - \hat{\mathbf{y}}$. Let $\Delta := \frac{15d\cdot\delta\cdot s_1(\Sigma_1)}{r}\cdot\left(1+\frac{\varepsilon}{2(1-\varepsilon)\sqrt{\pi d}}\right)$, then we have

$$\exp(-\Delta) \leqslant \frac{D_{\mathscr{R}^2,\mathbf{B}^{-1}\mathbf{c}-\mathbf{y},r}(\mathbf{x})}{D_{\mathscr{R}^2,\mathbf{B}^{-1}\mathbf{c}-\hat{\mathbf{y}},r}(\mathbf{x})} \leqslant \exp(\Delta).$$

We note that requiring a relative error at most δ on each complex embedding of u, that is, $|\varphi_i(u_j) - \varphi_i(\hat{u_j})| \leqslant \delta\cdot|\varphi_i(u_j)|$ foor $i\leqslant 2d$ and $j\leqslant 2$, implies the first relative error bound.

Proof. Note that by assumptions we also have $s_1(\hat{\Sigma}) \leqslant (1+\delta)s_1(\Sigma_1)$. Moreover, we have $\mathbf{y}-\hat{\mathbf{y}} = \Sigma_1\mathbf{u} - \hat{\Sigma_1}(\hat{\mathbf{u}}-\mathbf{u}) - \hat{\Sigma_1}\mathbf{u}$, which gives us

$$\begin{aligned} \|\hat{\mathbf{y}}-\mathbf{y}\| &\leqslant s_1(\hat{\Sigma_1}-\Sigma_1)\|\mathbf{u}\| + s_1(\hat{\Sigma_1})\|\hat{\mathbf{u}}-\mathbf{u}\| \\ &\leqslant (2+\delta)\delta\cdot s_1(\Sigma_1)\cdot\|\mathbf{u}\|. \end{aligned} \tag{9}$$

Lemma 6 states that

$$\exp\left(\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^2,\mathbf{t},r}}[\psi(\mathbf{x})]\right) \leqslant \frac{D_{\mathscr{R}^2,\mathbf{t},r}(\mathbf{x})}{D_{\mathscr{R}^2,\hat{\mathbf{t}},r}(\mathbf{x})} \leqslant \exp\left(\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\leftarrow D_{\mathscr{R}^2,\hat{\mathbf{t}},r}}[\psi(\mathbf{x})]\right).$$

Let us call $A(\mathbf{x})$ the quantity in the left-hand exponential in the inequality above. Thanks to Lemma 6, Lemma 5 and Inequality (9), we have

$$\begin{aligned}|A(\mathbf{x})| &\leqslant \frac{(2+\delta)\delta \cdot s_1(\Sigma_1)\cdot\|\mathbf{u}\|}{r^2}\cdot\left(\|\mathbf{x}-\hat{\mathbf{t}}\| + \frac{r\cdot\varepsilon}{1-\varepsilon}\right)\\ &\leqslant \frac{15\cdot d\cdot\delta\cdot s_1(\Sigma_1)}{r}\cdot\left(1+\frac{\varepsilon}{2(1-\varepsilon)\sqrt{\pi d}}\right),\end{aligned}$$

where our assumptions gives that $4\sqrt{\pi}(2+\delta)\leqslant 15$, which is used for the second line. The right-hand side is identical. This gives our claim.

We will now deduce the minimal precision δ needed for our finite precision samplers to be indistinguishable from the ideal one. The assumptions below reflect the practical situation for the NTRU lattices we consider. In the statement below, discrete Gaussian tail bounds tell us that only an exponentially small portion of $\mathbf{x}$'s are outside Ω. For example, the first sampled vector $\mathbf{u}$ is normal, its norm follows a chi-squared law of dimension $2d$. Hence the probability that $\|\mathbf{u}\| > 2\sqrt{d}$ is less[11] than $\exp(-d/5)$. The constant c depends on the what is achieved during the key generation algorithm.

Corollary 1. *Let ε,δ such that $0\leqslant\max(\varepsilon,\delta)\leqslant 2^{-40}$. Keep the notation of Theorem 1 with $\Sigma_1 = \mathbf{B}^{-1}\Sigma_0$. For $r=(1/\pi)\sqrt{(1/2)\log(4d(1+1/\varepsilon))}$, let $\Omega = \{\mathbf{x}\in\operatorname{Supp} D_{\mathscr{R}^2,\mathbf{t},r} : \|\mathbf{x}-\hat{\mathbf{t}}\|\leqslant 2r\sqrt{\pi d}\}$. Assume $\varphi(\mathbf{B}\mathscr{R}^2)$ is a lattice of rank $2d\geqslant 512$ with $s_1(\mathbf{B}) < c\sqrt{q}$ for some constants $1\leqslant c<16$ and $q>2^{10}$, and let $\Sigma = (cr)^2 q\mathbf{I}_2\in\mathscr{K}_{\mathbb{R}}^{2\times 2}$. Let $\mathrm{IP}(\mathbf{x})$ resp. $\mathrm{FP}(\mathbf{x})$ be the probability that the infinite, resp. the finite precision version of Algorithm 9 outputs $\mathbf{Bx}$. Then we have*

$$\sup_{\mathbf{x}\in\Omega}\left|\frac{\mathrm{IP}(\mathbf{x})}{\mathrm{FP}(\mathbf{x})}-1\right| \leqslant 17d\cdot c^2\cdot\delta.$$

Proof. We now want an upper bound on Δ. Let $s_+(\mathbf{B}^{-1}), s_-(\mathbf{B}^{-1})\in\mathscr{K}_{\mathbb{R}}$ be the singular values of $\mathbf{B}^{-1}$. They satisfy $s_+(\mathbf{B}^{-1})\cdot s_-(\mathbf{B}^{-1}) = 1/q$ so for each complex embedding we have $|\varphi_i(s_+(\mathbf{B}^{-1}))|\cdot|\varphi_i(s_-(\mathbf{B}^{-1}))| = 1/q$ as well. By construction, this means that there are d pairs of the singular values of $\varphi(\mathbf{B}^{-1})$ with a product equal to $1/q$. In particular, we deduce that

$$s_1(\mathbf{B}^{-1})\cdot s_{2d}(\mathbf{B}^{-1}) = s_1(\mathbf{B}^{-1})s_1(\mathbf{B})^{-1}\leqslant\frac{1}{q}.$$

Using properties of the spectral norm of matrices, we then obtain

$$s_1(\Sigma_1)\leqslant s_1(\mathbf{B}^{-1})s_1(\Sigma_0)\leqslant\frac{c\cdot s_1(\Sigma_0)}{\sqrt{q}}.$$

By assumptions, the spectrum of $\varphi(\mathbf{B}\mathbf{B}^*)$ is contained in the interval $(\frac{1}{c^2},c^2)q$, $\Sigma - r^2\mathbf{B}\mathbf{B}^* = \Sigma_0\Sigma_0^*$ and Σ and $\mathbf{B}\mathbf{B}^*$ commute, so we have $s_1(\Sigma_0)\leqslant cr\sqrt{q}$. With the definition of Δ and our assumptions again, we can now write

$$\Delta\leqslant 16d\cdot c^2\cdot\delta.$$

The result follows using that $\exp(16d\cdot c^2\delta)\leqslant 1+17d\cdot c^2\delta$ for our choice of parameters, Proposition 1 and the definition of the set Ω.

The last corollary follows with Lemma 3.

Corollary 2. *For $\varepsilon\leqslant 2^{-40}$ and $\delta\leqslant 2^{-51-2\log_2(c)}$ (resp. $\delta\leqslant 2^{-52-2\log_2(c)}$), Algorithm 9 in finite precision preserves up to 128 (resp. 256) bits of security if queried less than 2^{64} times over an NTRU lattice $\mathscr{L}(\mathbf{B})$ of rank 1024 (resp. 2048) and with $s_1(\mathbf{B}) = c\sqrt{q}$.*

[11] In practice, it is also less than $2^{-\lambda}$ as $d\geqslant 4\lambda$.

Smoothing parameter and floating point precision for the hybrid sampler

Proof (of Theorem 2). Let $\mathscr{D}(\mathbf{v})$ the probability that we obtain $\mathbf{v} = z_1\mathbf{b}_1 + z_2\mathbf{b}_2$ at Step 9, and $P(\mathbf{z}_i)$ be the probability that RingPeikert $_1$ outputs z_i at Steps 4 and 8 of Algorithm 3. By construction and Identity (6), we see that

$$\mathscr{D}(\mathbf{v}) = P(z_1)P(z_2) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2\right] \cdot D_{\mathscr{R}^2,d_1,\Sigma_1}(z_1)D_{\mathscr{R}^2,d_2,\Sigma_2}(z_2).$$

Note that the sampling covariances Σ_1 and Σ_2 "are above" $\eta_\varepsilon(\mathscr{R})$. Using the definitions of discrete Gaussians, our choices for the Σ_i's and Lemma 4, we obtain

$$\mathscr{D}(\mathbf{v}) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^4\right] \cdot \alpha \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{v}),$$

where we let $\alpha = \frac{\rho_\Sigma(\mathbf{B}\mathscr{R}^2-\mathbf{c})}{\rho_{\Sigma_1}(\mathscr{R})\rho_{\Sigma_2}(\mathscr{R})}$. Similarly as for Identity (6), we see that $\alpha \in [(\frac{1-\varepsilon}{1+\varepsilon})^4, (\frac{1+\varepsilon}{1-\varepsilon})^2]$, which leads us to

$$\mathscr{D}(\mathbf{v}) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^6, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^6\right] \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{v}). \tag{10}$$

We now differ mildly from [41] for the sake of parameter tuning. With our choice for ε, we see that $6(\log(1+\varepsilon) - \log(1-\varepsilon)) \leqslant 13\varepsilon$, and also that $(13\varepsilon)^2 \leqslant \varepsilon$ and $13\varepsilon \leqslant 1/2$, so that $\exp(13\varepsilon) - 1 \leqslant 14\varepsilon$. This means that

$$|\mathscr{D}(\mathbf{v}) - D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{v})| \leqslant 14\varepsilon \cdot D_{\mathbf{B}\mathscr{R}^2,\mathbf{c},\Sigma}(\mathbf{v}),$$

from which we get our claims.

Using the same method for the Peikert sampler, setting $\varepsilon \leqslant 2^{-40}$, resp. 2^{-41} preserves up to $\lambda = 128$, resp. 256 bits of security for $Q = 2^{64}$ queries.

Since the hybrid sampler relies on the ring version of the Peikert sampler, it is no surprise that the precision analysis is quite similar. As we are interested in the particular case of NTRU lattices, the situation is a bit simpler than in [41]. Indeed, here the different sampling centers are known exactly because in practice the starting center $\mathbf{c}$ and the $\langle\widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i\rangle$'s are in $\mathscr{K}$. Also, it is enough to target a scalar covariance matrix. We thus analyze Algorithm 4. Steps 4 to 6 and 9 to 11 are the explicit steps done in the Peikert sampler; in particular, x_2 and x_1 are essentially discrete Gaussians in $\mathscr{R}$ with standard deviation parameter σ.

Again, we implicitely consider versions of the algorithm that abort whenever the continuous Gaussian sampler or the discrete Gaussian sampler within the Peikert sampler output too large elements.

Proposition 2. *Let $r, \delta, \varepsilon > 0$, and keep the notation of Algorithm 4. For $u_1, u_2 \in \mathscr{K}_\mathbb{R}$ with $\|u_i\| \leqslant \sqrt{2d}$, let $y_i = \sigma_i u_i$. Assume that we are given $\hat{u}_i, \hat{\sigma}_i$'s and $\hat{y}_i = \hat{\sigma}_i\hat{u}_i$ such that for $i = 1, 2$:*

- $\|u_i - \hat{u}_i\| \leqslant \delta \cdot \|u_i\|$;
- $\|\sigma_i - \hat{\sigma}_i\|_\infty \leqslant \delta \cdot \|\sigma_i\|_\infty$.

Further, let $t_i = d_i - y_i$ and $\hat{t}_i = d_i - \hat{y}_i$. Assume that x_1, x_2 are such that $\max(\|x_i - t_i\|, \|x_i - \hat{t}_i\|) \leqslant r\sqrt{2\pi d}$ for $i = 1, 2$, and let $\Delta := \frac{15d\cdot\delta\cdot\max(\|\sigma_1\|_\infty, \|\sigma_2\|_\infty)}{r} \cdot \left(1 + \frac{\varepsilon}{(1-\varepsilon)\sqrt{2\pi d}}\right)$. Then we have

$$\exp(-\Delta) \leqslant \frac{\mathscr{D}(\mathbf{x})}{\hat{\mathscr{D}}(\mathbf{x})} \leqslant \exp(\Delta),$$

where $\mathscr{D}(\mathbf{x}) = D_{\mathscr{R},t_2,r}(x_2)D_{\mathscr{R},t_1,r}(x_1)$ and $\hat{\mathscr{D}}(\mathbf{x}) = D_{\mathscr{R},\hat{t}_2,r}(x_2)D_{\mathscr{R},\hat{t}_1,r}(x_1)$.

Proof. With our assumptions, we have

$$\begin{aligned}\|y_i - \hat{y}_i\| &\leqslant \|\hat{\sigma}_i - \sigma_i\|_\infty\|u_i\| + \|\sigma_i\|_\infty\|u_i - \hat{u}_i\| \\ &\leqslant \delta(2+\delta)\|\sigma_i\|_\infty \cdot \|u_i\|.\end{aligned}$$

Then the proof amounts to using twice Lemma 6 with $k = 1$ and our assumptions, in an identical way as in the proof of Proposition 1.

Corollary 3. *Let ε, δ such that $0 \leqslant \max(\varepsilon, \delta) \leqslant 2^{-40}$. Keep the notation of Proposition 2. For $r = (1/\pi)\sqrt{(1/2)\log(4d(1+1/\varepsilon))}$, let $\Omega_i = \{x_i \in \operatorname{Supp} D_{\mathscr{R},t_i,r} : \|x_i - \hat{t_i}\| \leqslant r\sqrt{2\pi d}\}$. Assume $\varphi(\mathbf{B}\mathscr{R}^2)$ is a lattice of rank $2d \geqslant 512$ with $|\mathbf{B}|_{\mathscr{K}} \leqslant c\sqrt{q}$ for some constants $1 \leqslant c < 16$ and $q > 2^{10}$, and let $\sigma = cr\sqrt{q} \in \mathbb{R}$. Let $\mathrm{IH}(x_1, x_2)$ resp. $\mathrm{FH}(x_1, x_2)$ be the probability that the infinite, resp. the finite precision version of Algorithm 4 outputs $\mathbf{v} = \mathbf{B}(x_1, x_2)$. Then we have*

$$\sup_{(x_1,x_2)\in\Omega_1\times\Omega_2} \left| \frac{\mathrm{IH}(x_1, x_2)}{\mathrm{FH}(x_1, x_2)} - 1 \right| \leqslant 17d \cdot c^2 \cdot \delta.$$

Proof. By construction and because $q^2 = \langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_1 \rangle \langle \widetilde{\mathbf{b}}_2, \widetilde{\mathbf{b}} \rangle_2$, we have $\|\sigma_i\|_\infty^2 \leqslant \frac{\sigma^2}{q^2}|\mathbf{B}|_{\mathscr{K}}^2 - r^2 \leqslant c^4 r^2$. With the definition of Δ and our assumptions again, we can now write

$$\Delta \leqslant 16d \cdot c^2 \cdot \delta,$$

and we conclude as in Corollary 1.

Corollary 4. *For $\varepsilon \leqslant 2^{-40}$ (resp. $\varepsilon \leqslant 2^{-41}$) and $\delta \leqslant 2^{-50-2\log_2(c)}$ (resp. $\delta \leqslant 2^{-52-2\log_2(c)}$), Algorithm 9 in finite precision preserves up to 128 (resp. 256) bits of security if queried less than 2^{64} times over an NTRU lattice $\mathscr{L}(\mathbf{B})$ of rank 1024 (resp. 2048) and with $|\mathbf{B}|_{\mathscr{K}} = c\sqrt{q}$.*

F Additional gadgets

In Algorithm 10,11,12 we recall basic gadgets in the literature. Note that Unmask corresponds to FullAdd from [4] and its NIo security is already proved there. The SNI security of Mul and RefreshM is proved in [3].

Algorithm 8: MaskedHybrid Gaussian sampler

Input: An arithmetic masking modulus $M = q^{\ell+2k}$ such that $k = k_1 + k_2$ and $\ell > 0$; a target center $\mathbf{c} \in \mathscr{R}^2$; a masked secret matrix $[[\![\mathbf{b}_1]\!]_q, [\![\mathbf{b}_2]\!]_{q^{\ell+k}}]$ such that $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$; a masked covariance $[\![q^{k_2}\sigma_1]\!]_{q^{\ell+k}}$ and $[\![q^{k_2}\sigma_2]\!]_{q^{\ell+2k}}$ such that $\sigma_i := \sqrt{\frac{\sigma^2}{\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i\rangle} - r^2} \in \mathscr{K}_{\mathbb{R}}^{++}$; a masked precomputed elements $[\![q^k\boldsymbol{\beta}_1]\!]_{q^{\ell+k}}$ and $[\![q^k\boldsymbol{\beta}_2]\!]_{q^{\ell+2k}}$ such that $\boldsymbol{\beta}_i = \frac{\widetilde{\mathbf{b}}_i^*}{\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i\rangle_{\mathscr{K}}}$, where $[\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2]$ is GSO of $\mathbf{B}$ over $\mathscr{K}$.

Result: $\mathbf{z}$ with distribution negligibly far from $D_{\mathscr{L},\mathbf{c},\sigma^2\mathbf{I}_{2d}}$.

Offline

for $j \in [0, d-1]$ **do**
  $[\![q^{k_1}u_{1,j}]\!]_{q^{\ell+k}} \leftarrow \mathsf{MaskedContGauss}(t, q^{\ell+k})$
  $[\![q^{k_1}u_{2,j}]\!]_{q^{\ell+2k}} \leftarrow \mathsf{MaskedContGauss}(t, q^{\ell+2k})$
end for
$[\![q^{k_1}u_1]\!]_{q^{\ell+k}} := ([\![q^{k_1}u_{1,j}]\!]_{q^{\ell+k}})_{j\in[0,d-1]}$
$[\![q^{k_1}u_2]\!]_{q^{\ell+2k}} := ([\![q^{k_1}u_{2,j}]\!]_{q^{\ell+2k}})_{j\in[0,d-1]}$
$[\![q^k y_1]\!]_{q^{\ell+k}} \leftarrow \mathsf{PolyMul}([\![q^{k_2}\sigma_1]\!]_{q^{\ell+k}}, [\![q^{k_1}u_1]\!]_{q^{\ell+k}})$
$[\![q^k y_2]\!]_{q^{\ell+2k}} \leftarrow \mathsf{PolyMul}([\![q^{k_2}\sigma_2]\!]_{q^{\ell+2k}}, [\![q^{k_1}u_2]\!]_{q^{\ell+2k}})$

Online

/* first nearest plane */
$\mathbf{c}_2 \leftarrow \mathbf{c}, \mathbf{v}_2 \leftarrow \mathbf{0}$
$[\![q^k d_2]\!]_{q^{\ell+2k}} \leftarrow \mathsf{PolyMul}([\![q^k\beta_{2,1}]\!]_{q^{\ell+2k}}, c_{2,1}) + \mathsf{PolyMul}([\![q^k\beta_{2,2}]\!]_{q^{\ell+2k}}, c_{2,2})$
$[\![q^k z_2]\!]_{q^{\ell+2k}} \leftarrow [\![q^k d_2]\!]_{q^{\ell+2k}} - [\![q^k y_2]\!]_{q^{\ell+2k}}$
for $j \in [0, d-1]$ **do**
  $[\![x_{2,j}]\!]_{q^{\ell+k}} \leftarrow \mathsf{GaussShareByShare}_r\left(\frac{[\![q^k z_{2,j}]\!]_{q^{\ell+2k}}}{q^k}\right)$
end for
/* second nearest plane */
$[\![\mathbf{v}_1]\!]_{q^{\ell+k}} \leftarrow \mathsf{MatMul}([\![x_2]\!]_{q^{\ell+k}}, [\![\mathbf{b}_2]\!]_{q^{\ell+k}})$
$[\![\mathbf{c}_1]\!]_{q^{\ell+k}} \leftarrow \mathbf{c}_2 - [\![\mathbf{v}_1]\!]_{q^{\ell+k}}$
$[\![q^k d_1]\!]_{q^{\ell+k}} \leftarrow \mathsf{PolyMul}([\![q^k\beta_{1,1}]\!]_{q^{\ell+k}}, [\![c_{1,1}]\!]_{q^{\ell+k}}) + \mathsf{PolyMul}([\![q^k\beta_{1,2}]\!]_{q^{\ell+k}}, [\![c_{1,2}]\!]_{q^{\ell+k}})$
$[\![q^k z_1]\!]_{q^{\ell+k}} \leftarrow [\![q^k d_1]\!]_{q^{\ell+k}} - [\![q^k y_1]\!]_{q^{\ell+k}}$
for $j \in [0, d-1]$ **do**
  $[\![x_{1,j}]\!]_{q^\ell} \leftarrow \mathsf{GaussShareByShare}_r\left(\frac{[\![q^k z_{1,j}]\!]_{q^{\ell+k}}}{q^k}\right)$
end for
$[\![x_1]\!]_q := [\![x_1]\!]_{q^\ell}, [\![\mathbf{v}_1]\!]_q := [\![\mathbf{v}_1]\!]_{q^{\ell+k}}$ /* compute every share $\bmod q$ */
$[\![\mathbf{v}_1']\!]_q \leftarrow \mathsf{RefreshM}([\![\mathbf{v}_1]\!]_q)$
$[\![\mathbf{v}_0]\!]_q \leftarrow [\![\mathbf{v}_1']\!]_q + \mathsf{MatMul}([\![x_1]\!]_q, [\![\mathbf{b}_1]\!]_q)$
$\mathbf{v}_0 \leftarrow \mathsf{Unmask}([\![\mathbf{v}_0]\!]_q)$
return $\mathbf{v}_0$

Algorithm 9: RingPeikert sampler, variant

Input: A matrix $\mathbf{B} \in \mathscr{K}^{2\times 2}$ such that $\mathscr{L} = \varphi(\mathbf{B}\mathscr{R}^2)$ and a target center $\mathbf{c} \in \mathscr{K}_{\mathbb{R}}^2$;
Result: $\mathbf{z} \in \mathscr{L}$ with distribution negligibly far from $D_{\mathscr{L},\mathbf{c},\Sigma}$

Precomputed: a parameter $r \geqslant \eta_\varepsilon(\mathscr{R}^2)$, and $\Sigma_1 \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$ such that $\Sigma_1\Sigma_1^* = \mathbf{B}^{-1}\Sigma\mathbf{B}^{-*} - r^2$.
$\mathbf{y} \leftarrow \Sigma_1 \cdot (\mathcal{N}_{\mathscr{K}_{\mathbb{R}},1})^2$
$\mathbf{x} \leftarrow \lceil \mathbf{B}^{-1}\mathbf{c} - \mathbf{y}\rfloor_r$
return $\mathbf{z} \leftarrow \mathbf{B}\mathbf{x}$

Algorithm 10: Unmask - share reconstruction (NIo secure)

Input: The share $[\![x]\!]_M = (x_1, \ldots, x_{t+1})$
Result: x such that $x = x_1 + \ldots + x_{t+1} \mod M$

$(x_1, \ldots, x_{t+1}) \leftarrow \mathsf{RefreshM}(x_1, \ldots, x_{t+1})$
$x \leftarrow x_1 + \ldots + x_{t+1}$
return x

Algorithm 11: RefreshM - multiplication-based share refresh (SNI secure)

Input: The share $[\![x]\!]_M = (x_1, \ldots, x_{t+1})$
Result: $[\![y]\!]_M = (y_1, \ldots, y_{t+1})$ such that $y_1 + \ldots + y_{t+1} = x_1 + \ldots + x_{t+1} \mod M$

$(y_1, \ldots, y_{t+1}) \leftarrow (x_1, \ldots, x_{t+1})$
for $i \in [1, t+1]$ **do**
  for $i < j \leqslant t+1$ **do**
    $r \leftarrow_{\$} \mathbb{Z}_M$
    $y_i \leftarrow y_i + r$
    $y_j \leftarrow y_j - r$
  end for
end for
return $(y_1, \ldots, y_{t+1})$

Algorithm 12: Mul - shared multiplication over $\mathbb{Z}_M$ (SNI secure)

Input: The shares $[\![a]\!]_M = (a_1, \ldots, a_{t+1})$ and $[\![b]\!]_M = (b_1, \ldots, b_{t+1})$
Result: $[\![c]\!]_M = (c_1, \ldots, c_{t+1})$ such that $c = ab \mod M$

for $i \in [1, t+1]$ **do**
  $c_i \leftarrow a_i b_i$
end for
for $i \in [1, t]$ **do**
  for $j \in [i+1, t+1]$ **do**
    $r_{i,j} \leftarrow_{\$} \mathbb{Z}_M$
    $c_i \leftarrow c_i + r_{i,j}$
    $r_{j,i} \leftarrow (a_i b_j - r_{i,j}) + a_j b_i$
    $c_j \leftarrow c_j + r_{j,i}$
  end for
end for
return $(c_1, \ldots, c_{t+1})$