SHORTER HASH-AND-SIGN LATTICE-BASED SIGNATURES

* * *

THOMAS ESPITAU[*], MEHDI TIBOUCHI[*], ALEXANDRE WALLET[⋆], AND YANG YU[◇]

Abstract. Lattice-based digital signature schemes following the hash-and-sign design paradigm of Gentry, Peikert and Vaikuntanathan (GPV) tend to offer an attractive level of efficiency, particularly when instantiated with structured compact trapdoors. In particular, NIST postquantum finalist Falcon is both quite fast for signing and verification and quite compact: NIST notes that it has the smallest bandwidth (as measured in combined size of public key and signature) of all round 2 digital signature candidates. Nevertheless, while Falcon–512, for instance, compares favorably to ECDSA–384 in terms of speed, its signatures are well over 10 times larger. For applications that store large number of signatures, or that require signatures to fit in prescribed packet sizes, this can be a critical limitation.

In this paper, we explore several approaches to further improve the size of hash-and-sign lattice-based signatures, particularly instantiated over NTRU lattices like Falcon and its recent variant Mitaka. In particular, while GPV signatures are usually obtained by sampling lattice points according to some *spherical* discrete Gaussian distribution, we show that it can be beneficial to sample instead according to a suitably chosen *ellipsoidal* discrete Gaussian: this is because only half of the sampled Gaussian vector is actually output as the signature, while the other half is recovered during verification. Making the half that actually occurs in signatures shorter reduces signature size at essentially no security loss (in a suitable range of parameters). Similarly, we show that reducing the modulus q with respect to which signatures are computed can improve signature size as well as verification key size almost "for free"; this is particularly true for constructions like Falcon and Mitaka that do not make substantial use of NTT-based multiplication (and rely instead on transcendental FFT). Finally, we show that the Gaussian vectors in signatures can be represented in a more compact way with appropriate coding-theoretic techniques, improving signature size by an additional 7 to 14%. All in all, we manage to reduce the size of, e.g., Falcon signatures by 30–40% at the cost of only 4–6 bits of Core-SVP security.

1. Introduction

Currently deployed public-key cryptography is, to a large extent, vulnerable to general-purpose quantum computers. As the likelihood increases that such computers may be

built in the coming decades, it appears important to prepare the transition to quantum-secure primitives instead. Doing so, however, requires postquantum schemes that are not far below currently deployed ones in terms of efficiency. Selecting and recommending such schemes is the main goal of the ongoing NIST standardization effort for postquantum cryptography. As part of that effort, primitives based on algebraic lattices have generally been strong contenders, with good performance and conservative security analyses: many of the finalists are in that category.

For digital signatures in particular, two of the three NIST round 3 finalists are lattice-based: Dilithium [11, 28] and Falcon [36]. The third finalist, Rainbow [9], is a multivariate scheme that boasts very short signatures and fast signing and verification, but suffers from very large keys and has seen its security substantially reduced by recent attacks [3]; as a result, NIST has leaned towards the lattice candidates. Indeed, the lattice-based signatures are the only "TLS-ready" candidates, in the sense that they are reasonably efficient and have a relatively small bandwidth requirement (the sum of public key size and signature size, which is the relevant size metric for TLS and other protocols relying on public key certificates). Recently, isogeny-based signatures [8] have also emerged as possible options with even better bandwidth requirements (although they were developed too late for the current NIST process), but they are considerably slower than lattice-based schemes, and thus limited in terms of possible applications.

Dilithium and Falcon represent each of the two main paradigms for the construction of lattice-based signatures: Dilithium follows Lyubashevsky's Fiat–Shamir with aborts framework [26, 27], while Falcon uses the hash-and-sign framework of Gentry, Peikert and Vaikuntanathan [20]. Due to this and several other design choices (such as the deliberate avoidance of Gaussian sampling), Dilithium is substantially simpler and easier to implement. Falcon on the other hand, is the strongest contender in terms of performance: it has signing times on par with Dilithium or better, faster verification times, and its public key and signature sizes are significantly smaller (by a factor of ≈ 1.5 for public keys and ≈ 3.5 for signatures at equivalent security levels). In fact, NIST mentions that Falcon had the best bandwidth requirements of all nine round 2 candidates for signatures.

In terms of speed, at least on larger CPU architectures, both Dilithium and Falcon could replace currenly deployed schemes without much trouble: for example, Falcon–512 outperforms OpenSSL's implementation of ECDSA (as of version `1.1.1l`) for all supported curve parameters in terms of verification time (by far), and all parameters

except `nistp224` and `nistp256` for signing. Key and signature sizes, however, are a different story. While ECDSA over a 256-bit curve has 32-byte verification keys and 64-byte signatures those numbers are 897 and 666 respectively for Falcon–512, and 1312 and 2420 respectively for the smallest round 3 parameters of Dilithium: bandwidth requirements are thus ≈ 16 times larger with Falcon and ≈ 39 times larger with Dilithium.

These larger key and signature sizes can be a serious impediment for numerous applications. For example, the DNSSEC protocols transmits verification keys as well as signatures on DNS records for signed DNS zones, and this information has to fit within a single TCP DNS packet. ICANN has pointed out [37] how this could cause difficulties for the transition to postquantum signatures. Similarly, TLS handshakes involve the transmission of multiple signatures and verification keys, and larger keys and signatures lead to more data transmission at handshake stage. CloudFlare observed [40] that this caused the handshake to exceed the initial TCP congestion window of most network infrastructure, leading to substantial slowdown. DNSSEC and TLS (or routers worldwide) could in principle be updated to mitigate those issues, but the massive coordination needed to do so makes that unlikely even in the medium term. Finally, some protocols like blockchains also require storing considerable amounts of digital signatures, and are therefore directly affected by signature size in terms of storage requirements and communication cost.

In view of these challenges, exploring ways of making lattice-based signatures and keys shorter is of clear importance.

1.1. **Hash-and-sign signatures over lattices.** In this paper, we propose several approaches to reduce the size of lattice-based signatures, with particular emphasis on hash-and-sign signatures over NTRU lattices: we mainly have Falcon in mind, but our techniques also apply to its recent variant Mitaka [17], as well as to the earlier scheme of Ducas, Lyubashevsky and Prest [12]. In order to describe these approaches, it is useful to briefly recall the structure of these schemes.

First, following the framework of Gentry, Peikert and Vaikuntanathan, hash-and-sign signatures over lattices are constructed as follows: they are defined with respect to a certain lattice $\mathscr{L}$ (a subgroup of $\mathbb{Z}^d$, say), which is usually chosen q-ary (i.e., such that $q\mathbb{Z}^d \subset \mathscr{L}$ for some integer modulus q). The signing key is a good basis, or *trapdoor*, for the lattice $\mathscr{L}$, the knowledge of which makes it possible to solve the approximate closest vector problem for $\mathscr{L}$ within a relatively small factor. In other words, given an arbitrary vector $\mathbf{c} \in \mathbb{Z}^d$, the trapdoor makes it possible to find $\mathbf{x} \in \mathscr{L}$ such that the distance $\|\mathbf{x}-\mathbf{c}\|$ is relatively small. By carefully randomizing this operation, it also becomes possible to

do *discrete Gaussian sampling*: sample a lattice point $\mathbf{x} \in \mathscr{L}$ according to a distribution statistically close to the discrete Gaussian $D_{\mathscr{L},\sigma,\mathbf{c}}$ over $\mathscr{L}$ centered at $\mathbf{c}$ with relatively small standard deviation σ. On the other hand, the verification key is a "bad" basis of $\mathscr{L}$, with which one can decide membership to the lattice, but that is not good enough to enable finding close vectors or sample discrete Gaussians with small standard deviation.

Then, the signing algorithm proceeds as follows. The message to be signed is hashed to a certain point $\mathbf{c} \in \mathbb{Z}_q^d$, and the signer uses its discrete Gaussian sampling algorithm to sample a vector $\mathbf{x} \in \mathscr{L}$ according to $D_{\mathscr{L},\sigma,\mathbf{c}}$. The signature is then the vector $\mathbf{s} = \mathbf{x} - \mathbf{c}$, which is relatively short: $\|\mathbf{s}\| \approx \sigma\sqrt{d}$ (it can also be seen as a sample from the Gaussian distribution $D_{\mathscr{L}-\mathbf{c},\sigma}$ supported over the lattice coset $\mathscr{L} - \mathbf{c}$). To verify the signature, one recomputes $\mathbf{c}$ by hashing the message, checks that $\mathbf{x} = \mathbf{s} + \mathbf{c}$ belongs to the lattice $\mathscr{L}$ and that $\mathbf{s}$ is indeed short. Security relies in a crucial way on the discrete Gaussian sampling, which ensures that signatures follow a distribution that depend only on the lattice $\mathscr{L}$ and the message, and *not* on the specific trapdoor used by the signer (contrary to what happened in early insecure hash-and-sign constructions like NTRUSign and GGH, in which signatures would leak information on the trapdoor, and therefore ultimately allow key recovery [33, 13, 41]).[1]

A standard optimization is the following. Since the lattice $\mathscr{L}$ is q-ary, it can be described by a parity-check matrix $\mathbf{A} \in \mathbb{Z}_q^{k \times d}$ such that $\mathbf{x} \in \mathscr{L}$ if and only if $\mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q$. One can assume without loss of generality (at least for prime q) that $\mathbf{A} = [\mathbf{A}_0 | \mathbf{I}_k]$ for some $\mathbf{A}_0 \in \mathbb{Z}_q^{k \times (d-k)}$. Thus, for any $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}^{d-k} \times \mathbb{Z}^k$, we have $\mathbf{x} \in \mathscr{L}$ if and only if $\mathbf{A}_0\mathbf{x}_0 + \mathbf{x}_1 \equiv \mathbf{0} \bmod q$. In that setting, if the signature $\mathbf{s}$ computed above is written as $(\mathbf{s}_0, \mathbf{s}_1) \in \mathbb{Z}^{d-k} \times \mathbb{Z}^k$, one can simply transmit the compressed signature $\mathbf{s}_0$. Indeed, the verifier can then recover $\mathbf{s}_1$ using the relation:

$$\mathbf{0} \equiv \mathbf{A}\mathbf{x} \equiv \mathbf{A}(\mathbf{s} + \mathbf{c}) \equiv \mathbf{A}_0\mathbf{s}_0 + \mathbf{s}_1 + \mathbf{A}\mathbf{c} \bmod q$$

and hence $\mathbf{s}_1 \equiv -\mathbf{A}_0\mathbf{s}_0 - \mathbf{A}\mathbf{c} \bmod q$. Signature verification then consists in recovering the $\mathbf{s}_1$ component and checking that $\mathbf{s}$ is small as expected (if it is, it is a valid signature with respect to the uncompressed verification algorithm by construction, so compression does not weaken security).

A further optimization used in practical schemes is as regards to the representation of the signature vector $\mathbf{s}_0$. By construction, it follows a discrete Gaussian distribution:

[1] This independence of the distribution on the trapdoor could in principle be achieved by distributions other than Gaussians, and it was recently shown to be feasible [29], albeit with much worse parameters than can be achieved with Gaussian sampling.

therefore, its coefficients are far from uniform. They lie in an interval $[-B, B]$ with $B = \Theta(\sigma\sqrt{\log(d-k)})$, but concentrate around 0. Therefore, simply representing them as numbers in $[-B, B]$ is suboptimal: while the vector $\mathbf{s}_0$ has $\Theta\big((d-k)\log\sigma\big)$ bits of entropy, this naive representation would use $\Theta\big((d-k)\log\log(d-k)\big)$ more bits. This can be addressed by coding-theoretic compression techniques: for example, Falcon (following the Gaussian-sampling based Fiat–Shamir signature scheme BLISS [10]) uses Huffman coding to reduce the representation size.

1.2. **Our contributions.** Using the two techniques described above, it may seem that all the available information on $\mathbf{s}$ is used to make the signature smaller: by transmitting only $\mathbf{s}_0$, we fully use the fact that $\mathbf{s}$ is in a known lattice coset, and by carrying out Huffman coding, we also take advantage of its Gaussian distribution. Since those two properties basically describe the distribution, it does not seem easy to do better.

In this paper, we explore and analyze three further strategies to reduce signature size (and, in one case, verification key size as well): one on the coding-theoretic side, and two on the lattice side.

Improved coding of Gaussian vectors. Our first observation (although it is presented last in the paper) is that the Huffman coding technique used in Falcon is fairly suboptimal: in particular, it represents the sign bit of Gaussian samples separately, and carries out a unary encoding of the absolute value, which follows a folded Gaussian distribution. Instead, we show that we can represent the whole Gaussian sample more compactly using batch Golomb-Rice coding with ANS (Asymetric Numeral System), and achieve a representation size for the full vector very close to the entropy bound without any computationally expensive technique like arithmetic coding. This allows us to reduce the size of signatures for Falcon by 7–14% essentially for free, and applies to all Gaussian sampling techniques in a black-box way.

Ellipsoidal Gaussians. Our second idea is based on the observation that the hardness of the approximate closest vector problem that underlies the security of a hash-and-sign based signature is, roughly speaking, determined by the volume of the decoding domain (the domain around the hashed point $\mathbf{c}$ that contains lattice vectors $\mathbf{x}$ corresponding to valid signatures $\mathbf{s} = \mathbf{x} - \mathbf{c}$). When transmitting the entire vector $\mathbf{s}$, it is thus optimal to choose the decoding radius as a ball around $\mathbf{c}$, and hence sample $\mathbf{x}$ according to a spherical Gaussian around $\mathbf{c}$, so as to minimize the length of $\mathbf{s}$ for a fixed decoding volume. However, as we have seen, we actually only transmit $\mathbf{s}_0$. Therefore, one can try to make the actually transmitted signature shorter by choosing a different decoding domain making

the transmitted part $\mathbf{s}_0$ shorter, and the recovered part $\mathbf{s}_1$ longer, while maintaining the overall decoding volume constant.

This intuition can be realized by sampling $\mathbf{x}$ according to an *ellipsoidal* discrete Gaussian distribution instead of a spherical one. Indeed, existing lattice Gaussian samplers either support ellipsoidal Gaussians out-of-the-box (as is the case for the Klein–GPV [20] and Peikert [34] samplers) or can be fairly easily adapted to do so for our ellipsoids of interest (as is the case for Prest's hybrid sampler [35, 17] and the fast Fourier orthogonalization-based sampler [14] used in Falcon).

There are of course substantial technical difficulties to address in order to fully make this idea work. To begin with, one needs to verify that it is possible to construct trapdoors for these ellipsoidal Gaussian samplers that achieve the same decoding volume as the one we started from; this is experimentally validated in our case. Moreover, while preserving the decoding volume is a rule of thumb to maintain security, extensive analysis is needed to evaluate the actual security level of the resulting scheme, at least for practical constructions like DLP, Falcon and Mitaka whose security is heuristic in nature (it provably reduces to, e.g., approximate CVP in a certain family of lattices, but the concrete parameters are too aggressive to support worst-case to average-case reductions in the style of [38, 39]). As a matter of fact, we find that this approach does cause a mild security loss of a 3–4 bits for typical parameters, when reducing signature size by 20–30%. Given the comfortable security margin of lattice-based constructions, this is likely to be an acceptable trade-off in many contexts.

Using a smaller modulus q. A simpler idea of the same flavor as the previous one is to simply reduce the modulus q with respect to which the q-ary lattice $\mathscr{L}$ is defined. We focus on NTRU lattices in what follows. The security analysis already carried out for the NTRU-based schemes DLP, Falcon and Mitaka shows that, at the proposed parameters for those schemes (and unlike other schemes like ModFalcon [6]), the best attacks are actually independent of q. As a result, it is possible to increase or decrease q in a certain range at no security loss, up to the point where other q-dependent attacks start to kick in.

For those lattices, the trapdoor makes it possible to sample signatures $\mathbf{s} \sim D_{\mathscr{L}-\mathbf{c},\sigma}$ with parameter $\sigma = \Theta(\sqrt{q})$: the transmitted vector $\mathbf{s}_0 \in \mathbb{Z}^{d/2}$ then has coefficients of magnitude $\approx \sqrt{q}$ (and can be represented by $\Theta(\log(d\sqrt{q}))$ bits after encoding). Moreover, the module structure reduces the parity-check matrix (i.e., the verification key of the signature scheme) to a single ring element $\mathbf{h}$ which can be seen as a uniform-looking element of $\mathbb{Z}_q^{d/2}$. As a result, a very simple way to reduce both signature size and verification key size is to

choose a smaller q: reducing q by a factor of γ should reduce signature size by roughly $\frac{d}{4}\log_2\gamma$ bits and verification key by $\frac{d}{2}\log_2\gamma$ bits.

FALCON parameters (like BLISS, DLP, and MITAKA in the power-of-two setting) are chosen for the modulus $q = 12289$, which is the smallest prime with the property that $q \equiv 1 \bmod 2^{12}$, making it number theoretic transform-friendly for power-of-two cyclotomics up to dimension 2048 (and in particular also 512 and 1024). Reducing q loses this property, and therefore can be seen as a trade-off. Practically speaking, however, this is a fairly minor trade-off as far as larger CPU architectures are concerned, because FALCON mostly relies on transcendental FFT instead of NTT for multiplication. NTT is only used for simplicity in signature verification and a small part of key generation, but it is easy to replace it by FFT followed by reduction modulo q everywhere, at little performance cost. And the same holds for variants like MITAKA.

An obvious question, however, is how far we can go. Certainly, arbitrarily small values of q should be impossible, if only for the fact that signatures would not "fit" anymore (in the sense that $\|\mathbf{s}\|_\infty$ would exceed $q/2$). But even before that, one encounters q-dependent attacks that slightly reduce security with respect to forgeries, as well as an issue with the generation of trapdoors. Normally, the NTRU trapdoor consists of a pair of ring elements $(\mathbf{f}, \mathbf{g})$ such that $\mathbf{h} = \mathbf{g}/\mathbf{f} \mod q$ over the ring. Moreover, $\mathbf{f}$ and $\mathbf{g}$ have to be sampled such that $\|(\mathbf{f}, \mathbf{g})\| \approx \sqrt{q}$. As a result, $\mathbf{f}$ and $\mathbf{g}$ are normally sampled as discrete Gaussians with parameter $\approx \sqrt{q/d}$. However, when q becomes small, $\sqrt{q/d}$ can go below 1 (or more precisely, below the so-called "smoothing parameter" of $\mathbb{Z}^d$), at which point the discrete Gaussian vector $(\mathbf{f}, \mathbf{g})$ stops "behaving like" a continuous Gaussian. It becomes ternary and sparse, with abnormally high probability of very low Hamming weight, giving rise to weak keys with non-negligible probability.

The correct approach is then to generate $(\mathbf{f}, \mathbf{g})$ directly as sparse ternary vectors of prescribed Hamming weight in order to reach to target length $\approx \sqrt{q}$ (and this observation also applies to the ellipsoidal case for very skewed ellipsoids). This eliminates the abnormal behavior of sub-smoothing discrete Gaussians, but still opens up the possibility of additional attacks exploiting the small, sparse secret keys. We therefore carefully analyze those attacks, and find that they allow us to reduce q down to values like $q = 257$ at little security loss, and for very substantial gains in terms of key and signature size!

Security analysis. As was apparent from the previous discussion, the security analysis of our new compression techniques relies on extensive cryptanalytic work. Since there is no simple way of relating the security of a scheme like FALCON between different values of q, or between different choices of Gaussian covariance matrices, one has to estimate the

TABLE 1. Parameters and classical bit security estimates for FALCON and MITAKA with $q = 257$ and ellipsoidal Gaussians with factor $\gamma = 8$ compared to the original schemes, in dimension 512 and 1024.

	FALCON–512			MITAKA–512		
	Security	Sig Size	Key Size	Security	Sig Size	Key Size
Original	123	666	896	102	710	896
Small $q = 257$	118	425	576	94	475	576
Ellipsoidal $\gamma = 8$	116	410	896	92	460	896
	FALCON–1024			MITAKA–1024		
	Security	Sig Size	Key Size	Security	Sig Size	Key Size
Original	272	1280	1792	233	1405	1792
Small $q = 257$	264	805	1152	209	935	1152
Ellipsoidal $\gamma = 8$	261	780	1792	204	905	1792

best attacks in each setting and parameter range. As usual, this is done separately for forgery (which follows a fairly standard methodology, but with appropriate twists for the ellipsoidal setting) and key recovery (where more subtle attacks come into play).

We in particular identify several parameter regimes relevant to the key recovery analysis, and carefully evaluate possible attacks in each of them. For ellipsoidal sampling, we distinguish between a range where both components $(\mathbf{f}, \mathbf{g})$ of the trapdoor are Gaussian, and a range where the smaller component becomes ternary and sparse (and is therefore chosen with fixed Hamming weight). Similarly, for the small q case, while the security analysis of FALCON and MITAKA applies directly for Gaussian $(\mathbf{f}, \mathbf{g})$, other attacks become relevant in the sparse ternary regime.

As part of this analysis, we propose several new lattice-based attacks that may be of independent interest.

Resulting parameters. Example parameters achievable with our approaches, including signature size, verification key size and classical bit security, are presented in Table 1. More complete numbers can be found in Table 2. As we can see, our techniques lead to a gain of 30–40% in signature size for FALCON, for example, at the cost of only a few bits of Core-SVP security. Using small q also leads to a considerable improvement in key size, of around 35%.

1.3. **Related works.** Chuengsatiansup et al. extended the FALCON design to NTRU lattices of larger (module) ranks and proposed MODFALCON [6]. This relaxation of constraints lead to additional parameters sets for intermediate security level. We note that our techniques can apply to MODFALCON as well.

In [5], Chen, Genise and Mukherjee introduce the notion of approximate trapdoors to construct smaller hash-and-sign signatures based on LWE and SIS. The size of such signatures is then further reduced using elliptic Gaussian sampling in [22]. However, we stress that these constructions rely on Micciancio-Peikert "gadget trapdoors" [30], and that adapting their techniques to the NTRU setting that is the focus of our paper seems far from being straightforward. On the other hand, some of our analysis and techniques could be used in Micciancio-Peikert schemes. efficient than NTRU-trapdoor based signatures.

Asymmetric variants of LWE and SIS were studied in [42] and used to design lattice-based cryptosystems. The assymetry allows to reduce the bandwidth at no cost on the security level, and the flavour reminds of the elliptic Gaussian sampling of our work. We note that [42] focuses on lattice-based KEM and Fiat-Shamir signatures, which have constraints and challenges quite different from our setting.

Lastly, some efforts have been made to design lattice schemes with a small modulus. By using error correcting codes, the modulus in LWE-based KEMs can be reduced to byte-level [25]. Fouque et al. designed BAT–KEM, also based on optimal NTRU trapdoors combined with a new decryption approach to work with small moduli [19]. While the underlying objects in there schemes and ours are similar, the cryptanalysis of KEM and signatures are significantly different problems. modulus size affects the security of the signature scheme.

2. BACKGROUND

When f is a real-valued function over a countable set S, we note $f(S) = \sum_{s\in S} f(s)$ assuming that this sum is absolutely convergent. Write $\mathbf{A}^t$ for the transpose of any matrix $\mathbf{A}$. Let $Q \in \mathbb{R}^{n\times n}$ be a symmetric matrix. We write $Q \succ 0$ when Q is *positive definite*, i.e. $\mathbf{x}^t Q\mathbf{x} > 0$ for all non-zero $\mathbf{x} \in \mathbb{R}^n$. We also write $Q_1 \succ Q_2$ when $Q_1 - Q_2 \succ 0$. It holds that $Q \succ 0$ if and only if $Q^{-1} \succ 0$ and that $Q_1 \succ Q_2 \succ 0$ if and only if $Q_2^{-1} \succ Q_1^{-1} \succ 0$. A positive definite matrix Q defines a norm as $\|\mathbf{x}\|_Q = \sqrt{\mathbf{x}^t Q\mathbf{x}}$, and corresponds uniquely to a bilinear form $\langle \mathbf{x}, \mathbf{y}\rangle_Q = \mathbf{x}^t Q\mathbf{y}$. Let $s_{1,Q}(\mathbf{A}) = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_Q}{\|\mathbf{x}\|_Q}$.

A lattice $\mathscr{L}$ is a discrete additive subgroup in a Euclidean space. When the space is $\mathbb{R}^m$, and if it is generated by (the columns of) $\mathbf{B} \in \mathbb{R}^{m\times d}$, we also write $\mathscr{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^d\}$.

If $\mathbf{B}$ has full column rank, then we call $\mathbf{B}$ a basis and d the rank of $\mathscr{L}$. When the ambient space is equipped with a norm $\|\cdot\|_Q$, the volume of $\mathscr{L}$ is $\mathrm{Vol}_Q(\mathscr{L}) = \det(\mathbf{B}^t Q \mathbf{B})^{\frac{1}{2}} = |\det(\mathbf{B})|\sqrt{\det(Q)}$ for any basis $\mathbf{B}$.

Power-of-two cyclotomic fields. 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 its ring of algebraic integers is $\mathscr{R} := \mathbb{Z}[\zeta_d]$. The field automorphism $\zeta_d \mapsto \zeta_d^{-1} = \overline{\zeta_d}$ corresponds to the complex conjugation, and we write the image f^* of f under this automorphism. We have $\mathscr{K} \simeq \mathbb{Q}[x]/(x^d+1)$ and $\mathscr{R} \simeq \mathbb{Z}[x]/(x^d+1)$, and both are contained in $\mathscr{K}_\mathbb{R} := \mathscr{K} \otimes \mathbb{R} \simeq \mathbb{R}[x]/(x^d+1)$. Each $f = \sum_{i=0}^{d-1} f_i \zeta_d^i \in \mathscr{K}_\mathbb{R}$ can be identified[2] with its coefficient vector $(f_0, \ldots, f_{d-1}) \in \mathbb{R}^d$. The adjoint operation extends naturally to $\mathscr{K}_\mathbb{R}$, and $\mathscr{K}_\mathbb{R}^+$ is the subspace of elements satisfying $f^* = f$.

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 defines the so-called *canonical embedding* $\varphi(f) := (\varphi_1(f), \ldots, \varphi_d(f))$. It extends straightforwardly to $\mathscr{K}_\mathbb{R}$ and identifies it to the space $\mathcal{H} = \{\mathbf{v} \in \mathbb{C}^d \,:\, v_i = \overline{v_{d/2+i}}, 1 \leqslant i \leqslant d/2\}$. For notational simplicity, we sometimes identify an element $x \in \mathscr{K}_\mathbb{R}$ as $\varphi(x) \in \mathcal{H}$ and denote by $\varphi_i(x)$ its i-th coordinate. Note that $\varphi(fg) = (\varphi_i(f)\varphi_i(g))_{i \leqslant d}$. When needed, this embedding extends entry-wise to vectors or matrices over $\mathscr{K}_\mathbb{R}$. We let $\mathscr{K}_\mathbb{R}^{++}$ be the subset of $\mathscr{K}_\mathbb{R}^+$ which have all positive coordinates in the canonical embedding. We have a partial ordering over $\mathscr{K}_\mathbb{R}^+$ by $f \succ g$ if and only if $f - g \in \mathscr{K}_\mathbb{R}^{++}$. The algebra $\mathscr{K}_\mathbb{R}$ is also equipped with a norm $\mathrm{N}(x) = \prod_i \varphi_i(x)$, which extends the standard field norm.

$\mathscr{K}_\mathbb{R}$-valued matrices. For $Q \in \mathscr{K}_\mathbb{R}^{2\times 2}$, we write Q^* its conjugate-transpose, where $*$ is the conjugation in $\mathscr{K}_\mathbb{R}$. Positive definiteness extends to such matrices: we say Q is *totally* positive definite when $Q = Q^*$and all the d matrices $\varphi_i(Q)$ induced by the field embeddings are hermitian positive definite. We then write $Q \succ 0$. For example, $\mathbf{B}^*\mathbf{B} \succ 0$ for all $\mathbf{B} \in \mathscr{K}_\mathbb{R}^{2\times 2}$. A positive definite form over $\mathscr{K}_\mathbb{R}$ corresponds uniquely to a $\mathscr{K}_\mathbb{R}$-bilinear form $\langle \mathbf{x}, \mathbf{y} \rangle_Q = \mathbf{x}^* Q \mathbf{y}$. [3] Under the canonical embedding, it induces a euclidean norm on $\mathcal{H}$ as $\|\varphi(\mathbf{x})\|_Q^2 = \sum_i \varphi_i(\langle \mathbf{x}, \mathbf{x} \rangle_Q)$. Such forms come with a corresponding notion of orthogonality. In particular, the well-known Gram-Schmidt orthogonalization procedure for a pair of linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathscr{K}^2$ is defined as

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

[2] This is the so-called coefficient embedding.

[3] We keep the same notation as in the common real case, since in the context of our work it will causes no confusion.

One readily checks that $\langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2 \rangle_Q = 0$. 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 a given form Q, we let $|\mathbf{B}|_{\mathscr{K},Q} = \max(\|\varphi(\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle_Q)\|_\infty, \|\varphi(\langle \widetilde{\mathbf{b}_2}, \widetilde{\mathbf{b}_2} \rangle_Q)\|_\infty)^{1/2}$.

NTRU lattices. This work deals with free $\mathscr{R}$-modules of rank 2 in $\mathscr{K}^2$, or in other words, groups of the form $\mathscr{M} = \mathscr{R}\mathbf{x} + \mathscr{R}\mathbf{y}$ where $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$ span $\mathscr{K}^2$. A mild change compared to previous works on the subject is that we equip the ambient space $\mathscr{K}_\mathbb{R}^2$ with a totally positive definite form Q and its corresponding inner product. If we write $\mathbf{B}$ the basis matrix for $\mathscr{M}$, the volume of the associated lattice is $\mathrm{Vol}_Q(\mathscr{M}) = \mathrm{N}(\det(\mathbf{B}^* Q \mathbf{B}))^{1/2}$. If $\widetilde{\mathbf{B}}$ is the Gram-Schmidt orthogonalization of $\mathbf{B}$ with respect to Q, then we also have $\mathrm{Vol}_Q(\mathscr{M})^2 = \prod_i \mathrm{N}(\langle \widetilde{\mathbf{b}_i}, \widetilde{\mathbf{b}_i} \rangle_Q)$.

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:

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

where $F, G \in \mathscr{R}$ are such that $fG - gF = q$, and $\|(F, G)\|$ should be relatively small. This module is usually seen as a lattice of volume $q^d \,\mathrm{N}(\det Q)^{1/2}$ in $(\mathbb{R}^{2d}, Q)$ in the coefficient embedding.

Lemma 2.1 ([35, 17], adapted)**.** *Let $\mathbf{B}_{f,g}$ be a basis of an NTRU module and $\mathbf{b}_1 = (f, g)$. We have $\sqrt{q}\,\mathrm{N}(\det Q)^{1/(4d)} \leqslant |\mathbf{B}_{f,g}|_{\mathscr{K},Q}$ and*

$$|\mathbf{B}_{f,g}|_{\mathscr{K},Q}^2 = \max\left(\|\varphi(\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_Q)\|_\infty, \left\| \frac{q^2 \cdot \det Q}{\varphi(\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_Q)} \right\|_\infty \right).$$

Gaussians measures and module lattices. For a positive definite matrix $Q \in \mathbb{R}^{d \times d}$, the Gaussian function with standard deviation σ is $\rho_{Q,\sigma}(\mathbf{x}) = \exp(-\frac{1}{2}\|\mathbf{x}\|_Q^2/\sigma^2)$. The standard (spherical) Gaussian function corresponds to $Q = \mathbf{I}$. Then, for a full rank lattice $\mathscr{L}$ in $\mathbb{R}^d$ and a given $\mathbf{t} \in \mathbb{R}^d$, the discrete Gaussian probability with parameters $\mathbf{t}$ and σ with respect to the form Q is defined as

$$D_{\mathscr{L},Q,\sigma,\mathbf{t}}(\mathbf{x}) = \frac{\rho_{Q,\sigma}(\mathbf{x} - \mathbf{t})}{\rho_{Q,\sigma}(\mathscr{L} - \mathbf{t})},$$

where $\mathbf{x}$ ranges in $\mathscr{L}$. When $\mathbf{t} = 0$, we omit it. When given a totally positive definite Q over $\mathscr{K}_\mathbb{R}$ and representing $\mathscr{R}$-modules with any embedding of $\mathscr{K}$, we keep the same notation, that is, we omit writing the embedding in formulas, as the context will always be clear.

Algorithm 1: Ring sampler

Input: A target center $t \in \mathscr{K}_{\mathbb{R}}$, parameters $\sigma \in \mathscr{K}_{\mathbb{R}}^{\times}$ and a real $r > 0$.

Result: y with distribution close to $D_{\mathscr{R},(\sigma\sigma^* + r^2)^{-1}\mathbf{I},t}$

$x \leftarrow \sigma \cdot \mathcal{N}_{\mathscr{K}_{\mathbb{R}}}$

return $\lfloor t - x \rceil_r$

For any positive definite form Q, there are always matrices $\mathbf{T} \in \mathbb{R}^{d \times d}$ such that $Q = \mathbf{T}^t\mathbf{T}$ (one example is given by the Cholesky decomposition). One checks that $\rho_{Q,\sigma}(\mathbf{x}) = \rho_{\mathbf{I},\sigma}(\mathbf{Tx})$ for any such $\mathbf{T}$, and well-known results about lattice Gaussian measures then extend to any form Q. The smoothing parameter of a lattice $\mathscr{L}$ for a given $\varepsilon > 0$ is $\eta_{Q,\varepsilon}(\mathscr{L}) = \min\{s > 0 \ : \ \rho_{Q^{-1},1/s}(\mathscr{L}^\vee) \leqslant 1 + \varepsilon\}$. Here, $\mathscr{L}^\vee$ refers to the dual lattice, and its exact definition is not needed: in this work, it is enough to know that for a full rank lattice $\mathscr{L}(\mathbf{B}) \subset \mathbb{R}^d$, it is encoded by $\mathbf{B}^{-t}$. The next lemma says that above the smoothing parameter, a discrete Gaussian measure does not "see" cosets of a lattice (hence the name).

Lemma 2.2 (Adapted from [31]). *Let Q be a positive definite form over $\mathbb{R}^d$, $\mathbf{t} \in \mathbb{R}^d$ and $\varepsilon > 0$. Let $\mathscr{L} \subset \mathbb{R}^d$ be a full rank lattice. If $\sigma \geqslant \eta_{Q,\varepsilon}(\mathscr{L})$, we have $\rho_{Q,\sigma}(\mathscr{L} - \mathbf{t}) \in [\frac{1-\varepsilon}{1+\varepsilon}, 1] \cdot \rho_{Q,\sigma}(\mathscr{L})$.*

We will also use standard tail bounds for elliptic discrete Gaussians.

Lemma 2.3 (Adapted from [27]). *Let Q be a positive definite form over $\mathbb{R}^d$, $\mathbf{t} \in \mathbb{R}^d$ and $\varepsilon > 0$. Let $\mathscr{L} \subset \mathbb{R}^d$ be a full rank lattice and $\mathbf{x} \leftarrow D_{\mathscr{L},Q,\sigma,\mathbf{t}}$, where $\sigma > \eta_{Q,\varepsilon}(\mathscr{L})$. For any $\tau > 1$, we have $\mathbb{P}[\|\mathbf{x} - \mathbf{t}\|_Q > \tau \cdot \sigma\sqrt{d}] \leqslant 2 \cdot \frac{1+\varepsilon}{1-\varepsilon} \cdot \tau^d \exp((1 - \tau^2)d/2)$.*

Lastly, we give the following upper bound on the smoothing parameter.

Lemma 2.4 (Adapted from [20, 17]). *Let $\mathbf{B}\mathscr{R}^2$ be a free $\mathscr{R}$-module, and $[\mathbf{b}_1, \ldots, \mathbf{b}_{2d}]$ the basis of the associated lattice $\mathscr{L}$ in $\mathbb{R}^{2d}$. Let $\varepsilon > 0$. For all totally positive definite $\mathscr{Q} \in \mathscr{K}_{\mathbb{R}}^{2\times 2}$, we have $\eta_{\mathscr{Q},\varepsilon}(\mathscr{L}) \leqslant |\mathbf{B}|_{\mathscr{Q},\mathscr{K}} \cdot \eta_\varepsilon(\mathbb{Z}^d)$.*
For any positive definite form $Q \in \mathbb{R}^{2d\times 2d}$, we have $\eta_{Q,\varepsilon}(\mathscr{L}) \leqslant \max \|\widetilde{\mathbf{b}}_i\|_Q \cdot \eta_\varepsilon(\mathbb{Z})$, where $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{2d}$ is the Gram-Schmidt orthogonalization of the $\mathbf{b}_i$'s with respect to Q. For any integer $n > 0$, we have $\eta_\varepsilon(\mathbb{Z}^n) \leqslant \frac{1}{\pi}\sqrt{\frac{\log(2n(1+1/\varepsilon))}{2}}$.

Some Gaussian samplers. Algorithm 1 is a subcase of [34] and inspired of [35, 17]. It allows to sample spherical discrete Gaussians in $\mathscr{R}$ for adequate parameters, as long as a discrete Gaussian sampler *over the integer* is given.

Algorithm 2: Module Elliptic Gaussian sampler

Input: A target center $\mathbf{t} \in \mathcal{K}_{\mathbb{R}}^2$, a totally positive matrix $Q \in \mathcal{K}_{\mathbb{R}}^{2\times 2}$, a basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ of a free $\mathscr{R}$-module $\mathscr{M}$ and its GSO $[\widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_2}]$ with respect to Q, and a parameter $\sigma \in \mathcal{K}_{\mathbb{R}}$.

Result: $\mathbf{z}$ with distribution negligibly far from $D_{\mathscr{M},(\sigma\sigma^*)^{-1}Q,1,\mathbf{c}}$.

Precomputed: $\tau_i := \sqrt{\frac{\sigma\sigma^*}{\langle \widetilde{\mathbf{b}_i}, \widetilde{\mathbf{b}_i}\rangle_Q} - r^2} \in \mathcal{K}_{\mathbb{R}}^{++}$.

$\mathbf{s} \leftarrow 0$

$\widetilde{t_2} \leftarrow \frac{\langle \widetilde{\mathbf{b}_2}, \mathbf{t}\rangle_Q}{\langle \widetilde{\mathbf{b}_2}, \widetilde{\mathbf{b}_2}\rangle_Q}$

$x_2 \leftarrow$ Algorithm 1$(\widetilde{t_2}, \tau_2, r)$

$\mathbf{t}' \leftarrow \mathbf{t} - x_2\mathbf{b}_2, \mathbf{s} \leftarrow x_2\mathbf{b}_2$

$\widetilde{t_1} \leftarrow \frac{\langle \widetilde{\mathbf{b}_1}, \mathbf{t}'\rangle_Q}{\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1}\rangle_Q}$

$x_1 \leftarrow$ Algorithm 1$(\widetilde{t_1}, \tau_1, r)$

$\mathbf{s} \leftarrow \mathbf{s} + x_1\mathbf{b}_1$

return $\mathbf{s}$

Proposition 2.1 (Adapted from [34, 17])**.** *Let $\mathscr{D}$ be the output distribution of Algorithm 1. If $\varepsilon < \frac{1}{2}$ and $r \geqslant \eta_\varepsilon(\mathscr{R})$, then the statistical distance between $\mathscr{D}$ and $D_{\mathscr{R},1,\sigma\sigma^*+r^2,t}$ is bounded by 2ε and we have*

$$\sup_{y\in\mathscr{R}} \left| \frac{\mathscr{D}(y)}{D_{\mathscr{R},(\sigma\sigma^*+r^2)^{-1}\mathbf{I},1,t}(y)} - 1 \right| \leqslant 4\varepsilon.$$

We observe that equivalently, Algorithm 1 can reach any covariance parameter $\tau \in \mathcal{K}_{\mathbb{R}}^{++}$ as long as $\tau - r^2 \in \mathcal{K}_{\mathbb{R}}^{++}$. Algorithm 2 is a generalization of the so-called *hybrid sampler* of [35, 17] to obtain Gaussian ring elements with elliptic covariances. It relies on the fact that elliptic Gaussians are merely spherical Gaussians in a different metric and is proved in Appendix A.

Proposition 2.2. *Let $\mathscr{D}$ be the output distribution of Algorithm 2. If $\varepsilon < \frac{1}{2}$ and $\sigma\sigma^* \succ (|\mathbf{B}|_{\mathscr{K},Q} \cdot \eta_\varepsilon(\mathscr{R}))^2$, then the statistical distance between $\mathscr{D}$ and $D_{\mathscr{M},(\sigma\sigma^*)^{-1}Q,1,\mathbf{t}}$ is bounded by 7ε and we have*

$$\sup_{\mathbf{y}\in\mathscr{M}} \left| \frac{\mathscr{D}(\mathbf{y})}{D_{\mathscr{M},(\sigma\sigma^*)^{-1}Q,1,\mathbf{t}}(\mathbf{y})} - 1 \right| \leqslant 14\varepsilon.$$

As already observed, sampling elliptically amounts to sampling spherically but changing the form defining the metric. It is thus no surprise that the well-known Klein sampler [20] can be extended identically by simply computing the initial Gram-Schmidt orthogonalization with respect to the adequate form: this change is purely syntactic. In particular,

there is no obstruction either to extending FALCON's *fast Fourier sampler* [14]: its core mechanic relies on the underlying tower of cyclotomic field and an adequate representation of the Cholesky factor for the lattice basis. The proof and description would be tedious and uneventful, yet for the sake of modularity, we restrict ourselves to a statement in this article.

Proposition 2.3 (Adapted from [20, 35])**.** *The fast Fourier sampler of* [36] *can be extended to a Gaussian sampler over a module lattice* $\mathscr{L}(\mathbf{B}) \in (\mathbb{R}^{2d}, Q)$*. Let* $\mathscr{D}$ *be its output distribution. Let also* $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{2d}$ *be the Gram-Schmidt of* $\mathbf{B}$ *with respect to* Q*,* $\varepsilon < 1/2$ *and* $\mathbf{t} \in \mathbb{R}^{2d}$*. When* $\sigma > \eta_\varepsilon(\mathbb{Z}) \cdot \max_{i \leqslant 2d} \|\widetilde{\mathbf{b}}_i\|_Q$*, the statistical distance between* $\mathscr{D}$ *and* $D_{\mathscr{L}(\mathbf{B}),Q,\sigma,\mathbf{t}}$ *is bounded by* $(2d+1)\varepsilon$ *and we have*

$$\sup_{\mathbf{y} \in \mathscr{L}(\mathbf{B})} \left| \frac{\mathscr{D}(\mathbf{y})}{D_{\mathscr{L}(\mathbf{B}),Q,\sigma,\mathbf{t}}(\mathbf{y})} - 1 \right| \leqslant (4d+1)\varepsilon.$$

3. New hash-and-sign tradeoffs

3.1. **Shorter signatures by elliptic sampling.** In hash-and-sign over NTRU lattices, it is well-known that only one of the components of a signature $(s_1, s_2) \in \mathscr{L}_{\text{NTRU}}$ is needed as input to the verification algorithm. This comes from the algebraic definition of such lattices, as we always have $s_1 h = s_2 \bmod q$ when h is the corresponding public key. To compress signatures, it, therefore, makes sense to try to minimize the magnitude of the coefficients in the component that is sent. To this end, we let $\gamma \succ 1$ (in $\mathscr{K}_{\mathbb{R}}$) and consider the totally positive form

$$Q = \begin{pmatrix} \gamma^2 & 0 \\ 0 & \gamma^{-2} \end{pmatrix},$$

and keep the same notation for its version in the coefficient embedding. Note that the resulting lattice volumes are preserved, as $\det Q = 1$. Following Algorithm 3, a signature is an elliptic Gaussian in $\mathscr{L}_{\text{NTRU}}$ centered at $\mathbf{c} = (0, c)$, where c is the (hash of the) message. Such random vectors can be sampled with Algorithm 2 or implicit in Proposition 2.3, for an input basis $\mathbf{B}_{f,g}$ reaching a good quality (as conditioned by Lemma 2.1). the smallest $|\mathbf{B}_{f,g}|_{Q,\mathscr{K}}$ is, the shortest the signatures are.

Now, since Q "favors" vectors with smaller first components, we send s_1 as the signature. Indeed, we can show that the first component of elliptic signatures has an expected length shorter by a factor γ compared to "regular" spherical ones. We however keep our discussion at an informal level for the sake of clarity, as the arguments are standard. Note first that saying $\mathbf{s} \leftarrow D_{\mathscr{L},Q,\sigma,\mathbf{c}}$ is equivalent to saying $\mathbf{Ts} \leftarrow D_{\mathbf{T}\mathscr{L},\mathbf{I},\sigma,\mathbf{Tc}}$ for any $\mathbf{T}$ such that $\mathbf{T}^t\mathbf{T} = Q$. Taking $\mathbf{T} = \text{diag}(\gamma, \gamma^{-1})$, the first coordinates of $\mathbf{Tc}$ in the signing algorithm

Algorithm 3: Hash-and-sign

Input: an NTRU trapdoor $\mathbf{B}_{f,g}$ and message m; a parameter $\sigma > 0$, a quadratic form Q and an acceptance bound $B > 0$

Result: a signature $s \in \mathscr{R}$.

$c := \mathsf{hash}(m) \in \mathscr{R}$, $\mathbf{c} := \begin{pmatrix} 0 \\ c \end{pmatrix}$

Sample $\mathbf{s} = (s_1, s_2)$ from $D_{\mathscr{L}(\mathbf{B}_{f,g}),Q,\sigma,\mathbf{c}}$ with Algorithm 2

if $\|\mathbf{s} - \mathbf{c}\|_Q > B$ **then**

  Restart

end if

return s_1

Algorithm 4: Verification

Input: an NTRU public key h and a signature s for a message m; a quadratic form Q and and acceptance bound $B > 0$

Result: Accept or Reject.

$c := \mathsf{hash}(m) \in \mathscr{R}$

$s' = hs - c \bmod q$

if $\|(s, s')\|_Q > B$ **then**

  Reject

end if

Accept

are 0. Therefore, the first component of $\mathbf{Ts}$, i.e γs_1, closely follows a Gaussian of covariance $\sigma^2 \mathbf{I}_d$, which shows that the signature s_1 has an expected length of essentially $\frac{\sigma\sqrt{d}}{\gamma}$.

3.2. **Parameters selection.** The resilience of hash-and-sign over lattices against forgery requires signatures to be short. Getting short signatures is achieved thanks to a *trapdoor* for $\mathscr{L}_{\mathrm{NTRU}}$, that is, a basis composed of short vectors with good properties with respect to a selected sampling algorithm. We consider two instantiations of the framework, namely, Falcon [36] and the recent Mitaka [17]. Each of these schemes find good trapdoors with the following method. First, candidates f and g are sampled according to a fixed distribution. Because the resulting lattice is morally a 2 dimensional object with prescribed volume q, it is possible to deduce the quality $\mathcal{Q}(\mathbf{B}_{f,g}) = \alpha\sqrt{q}$ of the basis before computing it, so that if the expected quality is good, the basis is completed, else another pair f, g

is sampled. We first deal with the value of α depending on the scheme, then discuss the distribution of f, g.

In our work, we sometimes consider different norms on the ambient space. This could have an impact on the quality that the key generation algorithm achieves; our experiments however suggested that it has no impact on the trapdoors we could find. Additionally, FALCON relies on the algebraic structure of power-of-two cyclotomic fields, while MITAKA allow for more number field choices. However, here, we will restrict to the power-two-cyclotomic case.

3.2.1. *Quality of Gaussian samplers.* FALCON uses the so-called fast Fourier sampler [14], while MITAKA relies on the so-called hybrid sampler [35] to sample signatures. If $\mathbf{b}_1, \mathbf{b}_2$ is the module basis of $\mathscr{L}_{\mathrm{NTRU}}$ and $\mathbf{b}_1^{\mathbb{Z}}, \ldots, \mathbf{b}_{2d}^{\mathbb{Z}}$ its corresponding lattice basis, the requirements are:

- $\mathcal{Q}_{\mathrm{FALCON}} = \max_{i \leqslant 2d} \|\widetilde{\mathbf{b}}_i^{\mathbb{Z}}\|_Q = \alpha_{\mathrm{FALCON}} \cdot \sqrt{q}$ where $\alpha_{\mathrm{FALCON}} = 1.17$;
- $\mathcal{Q}_{d,\mathrm{MITAKA}} = |\mathbf{B}_{f,g}|_{Q,\mathscr{K}} = \alpha_{d,\mathrm{MITAKA}} \cdot \sqrt{q}$, where $\alpha_{512,\mathrm{MITAKA}} = 2.04$ and $\alpha_{1024,\mathrm{MITAKA}} = 2.33$.

The standard deviation parameters for our signatures are set with Lemma 2.4 as $\sigma = r\mathcal{Q}_*$ where:

- we want $\sigma_{\mathrm{FALCON}} \geqslant \eta_\varepsilon(\mathbb{Z}) \cdot \alpha_{\mathrm{FALCON}} \sqrt{q}$, so we take $r = \frac{1}{\pi}\sqrt{\frac{\log(2(1+1/\varepsilon))}{2}}$;
- we want $\sigma_{d,\mathrm{MITAKA}} \geqslant \eta_\varepsilon(\mathbb{Z}^d) \cdot \alpha_{\mathrm{MITAKA}} \sqrt{q}$, so we take $r = \frac{1}{\pi}\sqrt{\frac{\log(2d(1+1/\varepsilon))}{2}}$.

These parameters combined give us the tailcut rate of the used sampler. We set the rejection bound as

$$\rho = \tau \cdot \sigma\sqrt{2d}, \tag{1}$$

where $\tau = 1.04$ is enough to guarantee that 90% of samples might be too long, thanks to Lemma 2.3. Lastly, the analyses in [17] states that $\varepsilon_{\mathrm{MITAKA}} = 2^{-41}$, while FALCON claims $\varepsilon_{\mathrm{FALCON}} = 2^{-36}$.

3.2.2. *On the distribution of secret keys.* The standard choice [12, 36] for FALCON is to sample f, g as independent discrete Gaussians in $\mathscr{R}$ to satisfy

$$\mathbb{E}[\|(f, g)\|_Q^2] = \alpha_{\mathrm{FALCON}}^2 q, \tag{2}$$

which means the standard deviation parameter is $\sigma_{\mathrm{FALCON}} = r_{\mathrm{FALCON}} \alpha_{\mathrm{FALCON}} \sqrt{q}$. On top of several tricks to speed up the key-generation algorithm, MITAKA uses a different strategy. The approach is to look for good trapdoors among those which could already be used by

FALCON. This also means that the expected behavior of f, g for the Euclidean norm is the same. We now distinguish the regime where the norm is changed and an elliptic signature is sampled, from the regime where q is reduced and the signature are regular spherical samples.

Selection in twisted norm: To simplify the exposition, we take $\gamma \in \mathbb{R}$. Condition (2) becomes

$$\gamma^2 \mathbb{E}[\|f\|^2] + \frac{1}{\gamma^2}\mathbb{E}[\|g\|^2] = \alpha^2 q. \tag{3}$$

If we want to keep f, g as discrete Gaussians, Equation (2) shows that we can select $\sigma_f = \frac{\alpha}{\gamma}\sqrt{q/2d}$ and $\sigma_g = \gamma\alpha\sqrt{q/2d}$, where γ remains *a priori* arbitrary. This choice[4] has expectedly a large impact on the security, and γ should not be too large either.

In any case, when γ grows, there is a parameter window where f looks essentially sparse and ternary, and the target standard deviation may be below the smoothing parameter of $\mathbb{Z}$. Since we can no longer predict the behavior of Gaussians in that regime, it is then natural to sample it directly as a uniform ternary vector of small (fixed) Hamming weight κ, and we now have $\mathbb{E}[\|\|f\|^2] = \kappa$. This change also enables different attacks exploiting the sparseness of f, see also Section 4. The next step is simplified by balancing the terms in Equation (3), asking

$$\frac{1}{\gamma^2}\mathbb{E}[\|g\|^2] = \gamma^2\kappa = \frac{\alpha^2 q}{2}. \tag{4}$$

The distortion factor can then be as large as $\gamma = \alpha\sqrt{\frac{q}{2\kappa}}$, and we can keep g sampled as a spherical discrete Gaussian with $\sigma_g^2 = (\alpha\gamma)^2\frac{q}{2d}$.

Selection for small q's: The ambient norm corresponds here to $Q = \mathbf{I}$, and the situation is simplified by taking the same distribution for f, g. As q is now close to d, the standard deviation of secret keys in the usual setting makes them again behave essentially like ternary and sparse vectors. This prompts us to sample directly f, g uniform in the set of ternary vectors of hamming weight κ, which translates in the following constraint:

$$\kappa = \frac{\alpha^2 q}{2}. \tag{5}$$

This implies in particular that q should be slightly smaller than $2d$, and may open the road for combinatoric and hybrid attacks against the secret keys.

[4]In particular, we could have selected a variable Hamming weight; our analyses suggest that it is a suboptimal choice for security.

4. Security analysis

To assess the concrete security of our methods, we proceed using the usual cryptanalytic methodology of estimating the complexity of the best attacks against *key recovery attacks* on the one hand, and *signature forgery* on the other. For the rest of this section, we consider that the ambient norm over our lattices is given by

$$\|\mathbf{x}\|_\gamma^2 = \mathbf{x}^t Q_\gamma \mathbf{x} \quad \text{with} \quad Q_\gamma = \mathbf{T}_\gamma^t \mathbf{T}_\gamma \quad \text{and} \quad \mathbf{T}_\gamma := \begin{bmatrix} \gamma \mathbf{I}_d & \\ & \gamma^{-1}\mathbf{I}_d \end{bmatrix}, \tag{6}$$

for some *real* $\gamma \geqslant 1$. To better reflect the impact of this distortion factor, we propose a parameterized security analysis, and instantiate it depending on our use case (either elliptic sampling, or "small q" regime with $\gamma = 1$).

Lattice reduction setting. 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 [32]: 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 the following relation on the length of its Gram-Schmidt vectors:

$$\|\widetilde{\mathbf{b}}_i\| = \delta_\beta^{n-2(i-1)} \operatorname{Vol}_Q(\mathscr{L})^{\frac{1}{n}}, \quad \text{where} \quad \delta_\beta = \left(\frac{(\pi\beta)^{\frac{1}{\beta}} \cdot \beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}. \tag{7}$$

4.1. Forging signatures.

In the hash-and-sign paradigm signature, forging a signature boils down to finding a point $\mathbf{v} \in \mathscr{L}_{\mathrm{NTRU}}$ at distance at most ρ from a random space point $\mathbf{x}$. Since we are quite above $\lambda_1(\mathscr{L}_{\mathrm{NTRU}})/2$, this is an instance of the Approximate Closest Vector Problem (ApproxCVP). This problem can be solved using the so-called *Nearest-Cospace* framework developed by Espitau and Kirchner in [18]. Under the Geometric Series assumption, Theorem 3.3 of [18] states that the decoding can be done in time $\mathrm{Poly}(d)$ calls to a CVP oracle in dimension β under the condition

$$\|\mathbf{x} - \mathbf{v}\|_\gamma \leqslant \delta_\beta^{2d} \operatorname{Vol}_{Q_\gamma}(\mathscr{L}_{\mathrm{NTRU}})^{\frac{1}{2d}}.$$

Equivalently,an adversary can consider the lattice spanned by $\mathbf{T}_\gamma \mathbf{B}_h$ and decode $\mathbf{T}_\gamma \mathbf{x}$ in the usual $\ell_2$norm $\|\cdot\|$, where

$$\mathbf{B}_h = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{H} & q\mathbf{I}_d \end{bmatrix},$$

and $\mathbf{H}$ is the matrix of multiplication by h in the power basis of $\mathscr{R}$.

While this change with regards to the classical situation of Falcon and Mitaka([36, 17]) seems purely syntactic, it can have an impact on the best approach to decoding,

and some care must be taken in the details. Indeed, as mentioned in [6], a standard optimization of this attack consists in only considering the lattice spanned by a subset of the vectors of the public basis and performing the decoding within this sublattice. The only interesting subset seems to remove only the $k \leqslant d$ first vectors. The dimension is of course reduced by k, at the cost of working with a lattice with of relatively bigger *normalized* covolume.

Let $S \subset [d]$ a set of k indices. Write $\mathbf{H}_S$ as the submatrix with column indices outside of S, and $\mathbf{I}_S$ the analogous submatrix of $\mathbf{I}_d$. Let also $\mathbf{B}_S$ be the corresponding submatrix of $\mathbf{B}_h$ while keeping all the "q vectors". We then have

$$(\mathbf{T}_\gamma \mathbf{B}_S)^t (\mathbf{T}_\gamma \mathbf{B}_S) = \begin{pmatrix} \gamma^2 \mathbf{I}_{d-k} + \frac{1}{\gamma^2}\mathbf{H}_S^t \mathbf{H}_S & \frac{q}{\gamma^2}\mathbf{H}_S^t \\ \frac{q}{\gamma^2}\mathbf{H}_S & \frac{q^2}{\gamma^2}\mathbf{I}_d \end{pmatrix}.$$

By Shur's complement formula, we find

$$\mathrm{Vol}_{Q_\gamma}(\mathscr{L}(\mathbf{B}_S))^2 = (q/\gamma)^{2d} \cdot \det(\gamma^2 \mathbf{I}_{d-k}) = q^{2d}\gamma^{-2k}.$$

As such, we need to enforce the following condition on the blocksize β with respect to the rejection bound:

$$\rho \geqslant \min_{k \leqslant d} \left(\delta_\beta^{2d-k} q^{\frac{d}{2d-k}} \gamma^{-\frac{k}{2d-k}}\right). \tag{8}$$

From Section 3.2.1 and Equation (1), we know that ρ is proportional to $\sqrt{q}$ once other parameters are fixed. Then Eq. (8) is equivalent to

$$\tau \cdot \alpha \cdot \eta_\varepsilon \cdot \sqrt{2d} \geqslant \min_{k \leqslant d} \left(\delta_\beta^{2d-k} \left(\frac{\sqrt{q}}{\gamma}\right)^{\frac{k}{2d-k}}\right), \tag{9}$$

where α and η_ε depends on whether Falcon or Mitaka parameters are considered.

There are three noteworthy observations about Condition (9). In the previous security analysis for Falcon and Mitaka, saturating the bound showed that $k = 0$ was the best case[5] from an attacker's point of view. A first and immediate observation is that the distortion of the norm directly impacts the hardness of the forgery. For fixed q, larger distortion factors γ, we observed that $\gamma \geqslant 2.3$ for $d = 512$ and $\gamma \geqslant 1.7$ for $d = 1024$ made forgetting vectors interesting for the attacker. The second one is more subtle. Note that the regime of schemes such Falcon or Mitaka always assumes that q is fixed in advance. In our work, we tolerate smaller q, and it turns out that when q gets smaller, an attacker finds it advantageous to forget some of the vectors. Experimentally[6] we found

[5] But this does not hold for ModFalcon, as observed in [6].

[6] It is of course possible to *calculate* the local maximum of the function, but an experiment confirmation seems enough for the purpose of this work.

that the phase transition happens when $q \leqslant 2434$ for $d = 512$ and $q \leqslant 4820$ when $d = 1024$. Lastly, Condition (9) reveals that tolerating smaller q's with the standard norm, or keeping usual (larger) choices but twisting the norm by γ has essentially an identical effect on the forgery. One can indeed think of q/γ^2 as a "reduced modulus", or in other words, designing a signature scheme with $q' = \lfloor q/\gamma^2 \rceil$. Hence from the point of view of forgery, our compression techniques can be seen as equivalent. However, they differ notably when we enter the domain of key recovery.

4.2. **Key-recovery attacks.** As advertised in Section 3.2, there are three distinct regimes to consider:

- in the **twisted Gaussian** regime, we twist the norm by $\gamma > 1$ and have imbalanced Gaussian secret keys;
- the **twisted-mixed** regime, the norm is also twisted by a larger γ, so the first half f of the secret key is now sparse ternary with Hamming weight κ;
- and in the **small q** regime, we keep the standard norm, but $q \leqslant 2d/\alpha^2$ so that both f and g are sparse, ternary with Hamming weight $\kappa = \alpha^2 q/2$.

A direct approach to key recovery is to do lattice reduction on the public basis, aiming at finding a relatively short vector in the spanned lattice: such attacks are addressed in Section 4.2.1. Whenever (a part of) the key becomes sparse ternary, combinatorics and more importantly *hybrid* attacks (combining lattice reduction and meet-in-the-middle approach) can be considered as a potential threat. In particular, in our mixed setting, we propose in Section 4.2.2 a new hybrid approach, of a slightly different flavor than the well-known Howgrave-Graham approach [21].

We also identify a new attack exploiting the sparsity of the secret keys in Section 4.2.3 The core idea is that when at least f is sparse, the number of "modulo turns" $k := (fh - g)/q$ is expected to be small too. This leads to another lattice reduction attack in a suitable orthogonal lattice (of rank $2d$ in a $3d$ dimensional space), that can also be improved by the "hybridization" approach. We also consider different metric choices for the ambient space of the lattice. Finally, we deal with algebraic, combinatoric, and classic hybrid attacks in Section 4.2.4.

4.2.1. *Projection onto the tail of the reduced basis.* 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 $\mathbf{s} = (f, g)$ among all possible lattice vectors of norm bounded by $\|\mathbf{s}\|_\gamma = \sigma\sqrt{2d}$ (or a functionally equivalent vector, for instance $(\mu g, \mu f)$ for any unit μ of the number field). We make use of the so-called *projection trick* to avoid enumerating over all the sphere of radius $\sigma\sqrt{2d}$ (which contains around $\left(\frac{2d\sigma^2}{q}\right)^d$ vectors under the Gaussian heuristic).

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 $\mathbf{b}_1, \ldots, \mathbf{b}_{2d}$ the resulting vectors. Then, if we can recover the *projection* of the secret key onto the orthogonal space $\mathcal{P}$ to $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{2d-\beta-1})$, 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 is enough to find the projection of the secret key among the shortest vectors 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 $\sqrt{4/3} \cdot \ell$, where ℓ is the norm of the $2d - \beta$-th Gram-Schmidt vector $\widetilde{\mathbf{b}}_{2d-\beta}$ of the reduced basis. Under the GSA (7), we therefore have:

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

Moreover, considering that $\mathbf{s}$ behaves as a random vector of norm $\sigma\sqrt{2d}$, and using the GSA again, the expected norm of its projection over $\mathcal{P}$ is $\sqrt{\beta/(2d)}\|\mathbf{s}\|_\gamma = \beta^{\frac{1}{2}}\sigma$. Hence, we will retrieve the projection among the sieved vectors if $\beta^{\frac{1}{2}}\sigma \leqslant \sqrt{4/3}\ell$, that is if the following condition is fulfilled:

$$\sigma^2 \leqslant \frac{4q}{3\beta} \cdot \delta_\beta^{4(\beta+1-d)} \tag{10}$$

Remark. *This approach is similar to the one used in the security evaluation of* [1]*, but we use all the vectors given by the last step of sieving, resulting in a slightly stronger attack and as such more conservative parameters choices.*

Finding short vectors in tweaked-norm setting: As our scheme suggests the use of different (Euclidean) norms, when it comes to the analysis of key recovery, it is also legitimate to wonder which norm is indeed the best to mount lattice attacks. Let us assume that we take an inner product matrix G and split in blocks of size $d \times d$ as

$$G = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

with $A, C \in \mathrm{Sym}^+(\mathbb{R}, d)$. By homogeneity, we can restrict the study to the case where the determinant of G is 1. Hence, the squared norm of (f, g) (viewed as a vector over $\mathbb{Z}^{2d}$)

for this norm is $\langle Af, f\rangle + \langle Cg, g\rangle + 2\langle f, Bg\rangle$. Observe that since f, g (and thereof f, Bg) are *centered* independent vectors, the expected value of the inner product $\langle f, Bg\rangle$ is zero. Thus, we have by (bi)linearity:

$$\begin{aligned}\mathcal{E} := \mathbb{E}\left[\|(f,g)\|_G^2\right] &= \mathbb{E}\left[\langle Af, f\rangle\right] + \mathbb{E}\left[\langle Cg, g\rangle\right] + 2\mathbb{E}\left[\langle f, Bg\rangle\right] \\ &= \mathrm{Tr}(A\mathrm{Cov}(f)) + \mathrm{Tr}(C\mathrm{Cov}(g)). \end{aligned} \tag{11}$$

Following Section 3.2.2, we have:

- in the twisted Gaussian regime, $\sigma_f^2 = (\frac{\alpha}{\gamma})^2 \cdot \frac{q}{2d}$ and $\sigma_g^2 = (\alpha\gamma)^2 \cdot \frac{q}{2d}$;
- in twisted-mixed f has scalar covariance[7] with parameter $\frac{\kappa}{d} = \sigma_f^2$;
- in the small q regime, we have $\gamma = 1$ and $\mathrm{Cov}(f) = \mathrm{Cov}(g) = \frac{\alpha^2 q}{2d}$.

In all cases, Equation (11) becomes

$$\mathcal{E} = \frac{\alpha^2 q}{2d} \cdot \left(\frac{1}{\gamma^2}\mathrm{Tr}(A) + \gamma^2 \mathrm{Tr}(C)\right).$$

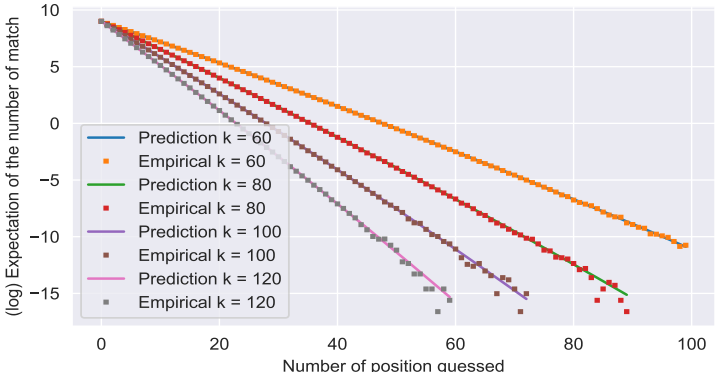
To favor lattice attacks, the used norm defined by G should minimize $\mathcal{E}$. By the arithmetic-geometric inequality and Fischer's inequality, we have

$$\frac{1}{\gamma^2}\mathrm{Tr}(A) + \gamma^2 \mathrm{Tr}(C) \geqslant \frac{d}{\gamma^2}\det(A)^{\frac{1}{d}} + d\gamma^2 \det(C)^{\frac{1}{d}} \geqslant 2d \cdot (\det(A)\det(C))^{\frac{1}{2d}} \geqslant 2d.$$

Hence $\mathcal{E}$ achieves a minimum $\alpha^2 q$ at $\mathrm{Tr}(A) = \gamma^2 d, \mathrm{Tr}(C) = \frac{d}{\gamma^2}$, which proves the optimality of Q_γ (Eq. (6)) whatever the regime.

4.2.2. *An hybrid attack on half-sparse vectors.* We now show that we can improve this attack by exploiting the sparsity of the f part of the secret key. Indeed, if its sparsity level is low, then with a reasonable probability we can guess the positions of some zeros of the vector. If such a guess of positions, say $\mathcal{I} \subseteq \{1, \cdots, 2d\}$ appears to be correct, we can intersect the NTRU lattice with $\mathbb{Z}^{\overline{\mathcal{I}}}$. (where $\overline{\mathcal{I}}$ refers to the complement of the set $\mathcal{I}$ in the overset $\{1, \cdots, 2d\}$) In this lattice, we can apply readily the methodology of Section 4.2.1 to retrieve the intersected secret and as such the secret itself. This new lattice has dimension $2d - |\mathcal{I}|$ and its covolume is likely to be q^d (see infra for a discussion of this phenomena). As a result, the normalized covolume of the intersection lattice is bigger than previously, and its dimension of course smaller. As such, this final lattice reduction part is now easier and thus faster. Hence, there exists a trade-off between the probability of right guessing (the more zeroes to guess, the harder it becomes to correctly guess their

[7] Indeed, $\mathbb{E}[f_i f_j] = 0$ for $i \neq j$ and by invariance of the distribution by permutation, all the diagonal elements are equal.



positions) and the time required by the lattice reduction.

Estimation of the cost of the attack:

Good guess probability estimation [Fixed Hamming weight]. We now derive an estimation of the probability of making a successful guess of the zero coefficients. Suppose that the sparsity of f is $0 < \kappa < d$ and that $|\mathcal{I}| = k$. Then, over the randomness of f, the probability of getting a correct guess is equal to the probability of f having its non-zero coefficient outside the k positions of $\mathcal{I}$, i.e. is $\binom{d-k}{\kappa}\binom{d}{\kappa}$. Remark now that we can enhance this probability by remarking that it suffices that a *conjugate* of f has is zero on $\mathcal{I}$. It seems however difficult to estimate such a probability explicitly as it depends on the pattern of $\mathcal{I}$. An underestimate of this probability consists in assuming that the event of right guessing for each of the conjugates are independent. Using this heuristic, the average number of conjugates with zeroes on $\mathcal{I}$ is $d\binom{d-k}{\kappa}\binom{d}{\kappa}^{-1}$. This heuristic is in practice precise enough for the simulation: we simulated the behavior, in cryptographically relevant parameters, of this expectation by repeating the counting on 2^{19} trials and report a relative error of at most 0.5% for results greater than 2^{-12}.

Volume of intersection. Suppose now that a good guess was performed. We intersect the whole NTRU $\mathcal{L}$ lattice with $\mathbb{Z}^{\overline{\mathcal{I}}}$ and claim that with high probability this lattice has volume in ℓ_2 norm equal to $q^{d/2}$. First, remark that it necessarily q-ary and as such that it is sufficient to study the rank of $\mathcal{L}$ with $\mathbb{Z}^{\mathcal{I}}$, which will be full rank with overwhelming probability, according to [7]. As such the volume of the intersection is expected to be $q^{\frac{1}{2}}$. We can now compute the volume in the twisted norm $\|.\|_\gamma$. Remember that we obtained the intersection by removing d rows coordinates over f, which are all scaled by the parameter γ in $\|.\|_\gamma$. Hence, the volume is now scaled by the determinant of the intersected Gram

matrix $G = \mathrm{Diag}(\overline{\mathcal{I}})Q_\gamma \mathrm{Diag}(\overline{\mathcal{I}})$, which is exactly γ^k. All in all the (normalized) volume of the intersected lattice for the twisted norm is $q^{\frac{d}{2d-k}}\gamma^{\frac{k}{2d-k}}$.

Remark. *The normalized covolume of the intersection is now bigger than the original normalized covolume (which is $\sqrt{q}$), making the lattice reduction attack slightly easier. However this normalized covolume is not large enough to enter the* overstretched NTRU *regime (see* [15] *for recent developments on this matter).*

Remark (On non-fixed Hamming weight secrets)**.** *It could seem natural to let the small secret vectors be sampled with ternary distribution and no restriction on the Hamming weight (as a limit case of a tail cut Gaussian for instance). However this choice is sub-optimal security-wise. Indeed in this case, a simple estimation reveals that the probability of getting abnormally short vectors (i.e. with shorter weight than expected) is sufficiently high to reduce the whole security of the scheme: in other words, the fraction of generable weak keys is too high. Fixing in advance the Hamming weight avoids this phenomenon and has zero drawback on the key generation nor the scheme itself.*

4.2.3. *A new attack using the small number of modulus rounds.* First of all, we stress that this attack only concerns the **twisted-mixed** and **small q** regimes, as it exploits the sparseness of f. In the twisted Gaussian regime, little can be said about the Hamming weight, and as the standard deviation parameter is still above the smoothing of $\mathbb{Z}$, it is also likely that the vector is not "so ternary", that is, it has enough coefficients of magnitude at least 2 so that the enumerating part of the attack becomes too costly anyway. As f is small, we can give a closer look at the size of the polynomial $hf - g$ which vanishes mod q by the construction of the NTRU basis (Section 2). It appears that $k := \frac{1}{q}(hf - g)$ has a norm closely related to the Hamming weight of f as it grows proportionally to $\sqrt{\kappa}$ (see infra. for an analysis of this fact). For small κ, this quantity is sufficiently small to be exploited in the lattice reduction. Indeed, instead of working modulo q as in the previous attack to recover directly f and g, we can aim at recovering directly the vector (f, g, k) in a rank two module, embedded in a $\mathscr{K}$-vector space of dimension 3. Since $fh - g = kq$, this module is nothing else than the orthogonal module to the vector $(h, -1, q)$. A public basis of this module (in rows) is

$$\mathbf{B} = \begin{pmatrix} 1 & h & 0 \\ 0 & q & 1 \end{pmatrix}.$$

On the space containing f, g the metric is given by Q_γ as defined in (6), and the "q part" is rescaled to take into account the expected length of k. Equivalently, this metric is

described over $\mathscr{K}$ by the matrix $D = \mathrm{diag}(\gamma^2, \gamma^{-2}, \delta^2)$, for a parameter δ to be discussed later. The corresponding Gram matrix is then

$$G = \mathbf{B}D\mathbf{B}^* = \begin{pmatrix} \gamma^2 + \frac{hh^*}{\gamma^2} & \frac{qh}{\gamma^2} \\ \frac{qh^*}{\gamma^2} & \delta^2 + \frac{q^2}{\gamma^2} \end{pmatrix},$$

and it follows that

$$\mathrm{N}_{\mathscr{K}/\mathbb{Q}}(\det(G)) = \mathrm{N}_{\mathscr{K}/\mathbb{Q}}\left(q^2 + (\gamma\delta)^2 + \frac{\delta^2}{\gamma^2}hh^*\right).$$

We now estimate the expected normalized volume $\mathcal{V} = \mathrm{N}_{\mathscr{K}/\mathbb{Q}}(\det G)^{1/4d}$ of $\mathscr{L}(\mathbf{B})$. The intuition guiding the calculation is that q^2 will be the dominating term in the expansion and that we want δ to be a "balancing parameter" for the expected norm of k. Ultimately, its choice will make $\gamma\delta$ to be a small fraction of q, and δ/γ to be constant, see Appendix B for details about the calculation:

$$\mathbb{E}[\mathcal{V}] \leqslant \sqrt{q} \cdot \left(1 + \frac{(\gamma\delta)^2}{q^2} + \frac{\delta^2}{\gamma^2} \cdot \frac{d}{12}\right)^{1/4}. \tag{12}$$

On the expected size of k: We now give a model of the distribution of the euclidean norm k in our setting, in the sense that its approximations match accurately our experimental results. Let f be uniform among the set of ternary polynomials of degree d with weight κ. Recall that we already assumed that h is uniform in $\mathscr{R}/q\mathscr{R}$. Then, we model the coefficients of fh as a sum of κ independent discrete uniform random variables in $[-q/2, q/2] \cap \mathbb{Z}$. Such a sum as expected value 0 and variance $\frac{\kappa(q^2-1)}{12}$. Assuming that the coefficients of hf behaves independently, then the (squared) expected norm of the vector hf/q is then $\frac{d\kappa(q^2-1)}{12q^2} \approx \frac{d\kappa}{12}$. In the twisted–mixed regime, as seen in Section 3.2.2, the vector g is a discrete Gaussian distributed with standard deviation $\sigma_g = \gamma\alpha\sqrt{q/2d}$, and thus the expected squared norm of g/q is $\frac{\alpha^4 q^2}{\kappa q^2} = \frac{\alpha^4}{\kappa}$.When q is small, Section 3.2.2 says $\|g\|^2 = \kappa$ so that $(fh - g)/q$ has a squared norm of $\kappa(\frac{d}{12} + \frac{1}{q})$, which is still reasonably close to $\frac{d\kappa}{12}$. Therefore, a reasonable approximation for the expected squared norm of k is

$$\mathbb{E}[\|k\|^2] = \frac{d\kappa}{12}.$$

The experiments reported in Appendix D, Figure D confirms the quality of the approximation.

With that additional estimation in our arsenal, we can now see concrete values for the parameters. We chose δ so that the vector (f, g, k) has balanced coordinates in the given norm. Since we have $\mathbb{E}[\|(f, g, k)\|_D^2] = \alpha^2 q + \delta^2\mathbb{E}[\|k\|^2]$, we set $\delta^2 = \frac{\alpha^2 q}{2\mathbb{E}[\|k\|^2]}$, and our attack has to find a short vector of expected length $\alpha(\frac{3q}{2})^{1/2}$. Now, in the twisted–mixed

regime, we have $(\gamma\delta)^2 = \frac{3\alpha^4}{2\kappa^2 d}q^2$ and $\frac{\delta^2}{\gamma^2} = \frac{12}{d}$, and in the small q regime, $\gamma = 1$ and $\delta^2 = \frac{\kappa}{\mathbb{E}[\|k\|^2]} = \frac{12}{d}$. In any case, Inequality (12) becomes:

$$\mathbb{E}[\mathcal{V}] \leqslant \begin{cases} \alpha\sqrt{q} \cdot \left(\frac{2}{\alpha^4} + \frac{3}{2\kappa^2 d}\right)^{1/4} & \text{in the twisted-mixed regime,} \\ \sqrt{q} \cdot \left(2 + \frac{12}{dq^2}\right)^{1/4} & \text{in the small } q \text{ regime.} \end{cases}$$

For our smallest considered Hamming weight, we observe that $\mathbb{E}[\mathcal{V}] \leqslant 1.19 \cdot \sqrt{q}$ in the twisted–mixed regime. This was in turn confirmed by our experiments: we computed the average of these normalized volumes for several classes of parameters and found that the ratio $\mathcal{V}/\sqrt{q}$ never exceeded $\alpha = 1.17$. In the small q regime, the experiments showed that $\mathcal{V}/\sqrt{q} \leqslant 1.19$ on average too. As in the previous attack, the vector f is sparse so we hybridize the lattice reduction attack with the guessing technique. The whole attack is algorithmically depicted in Appendix C.

4.2.4. *Combinatorial and hybrid attacks.* In this section, we list the other possible type of attacks on signatures, which are nonetheless not the most effective for the parameters we consider.

Exploiting the algebraic structure. The schemes we consider are defined over algebraic lattices, which have a rich structure that could in principle lead to improved attacks. However, there is no known way to improve all the algorithms previously mentioned for their general lattice equivalent by more than polynomial factors in an asymptotic sense (see for instance the speedup on lattice reduction of [23]), and they do not affect our concrete security levels.

Overstretched NTRU.. As observed in [24] and reanalysed in [15], 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 $q > n^{2.484}$ for binary secrets, implying that, as it is the case for Falcon and other NTRU-based NIST candidates, that even *very* significant improvements to this attack would still be irrelevant to the security of our proposed parameters: in fact, we are even further away from the fatigue point when reducing q!

Combinatorial and hybrid attacks. Odlyzko's meet-in-the-middle attack, and its recent improvements by May and Kirshanova–May, are a priori very relevant to our ternary sparse settings, particularly in the small-q case (and although non-ternary errors has not been analyzed in the literature, the Kirshanova–May improvements do in principle extend

to that setting as well, and hence could affect our ternary regime even for distortion). However, running bit security estimator for the state-of-the-art attack of this type shows that it is very far from competing with the lattice attacks considered earlier in this section. At best, they yield time complexities over 2^{180} in dimension 512, for example.

The hybrid attack of Howgrave-Graham [21], and its improved analysis by Wunderer, appears to be more of a threat in principle. However, again using the available estimator (adapted to use the Core-SVP metric for BKZ cost) reveals that attacks reach at best 2^{138} complexity in dimension 512, again not competing with tailor-made lattice approaches.

4.3. **Concrete security estimates.** Under the heuristics we explicited, we can estimate the concrete bit security of our techniques on the Falcon and Mitaka NTRU based hash and sign signaures schemes. The analysis translates into concrete bit-security estimates following the methodology of NewHope [1], sometimes called "core-SVP methodology". In this model [2], the bit complexity of lattice sieving (which is asymptotically the best SVP oracle) is taken as $\lfloor 0.292\beta \rfloor$ in the classical setting and $\lfloor 0.2570\beta \rfloor$ in the quantum setting in dimension β (using the recent progress of [4]).

4.3.1. *Example parameters.* We now present the concrete data obtained for our new trade-offs. In Table 2 we gathered several options for $d \in \{512, 1024\}$, and choices for moduli q and distortion factor γ, for both Falcon and Mitaka. The bit-security was obtained by taking into account our new attacks (impacting the more extremal ranges of parameters) and the last quantum sieving exponent for the core-SVP hardness, using updated versions of the scripts from the Falcon and Mitaka team. In Figures 1 and 2, we also provide curves representing the security level in function of the main compression parameter.

5. Batch compressing Gaussian vectors

In this section, we deal with the problem of efficient and lossless compression of a batch of random discrete Gaussian variables. Our goal is to further compress the s_1 part of the signature before outputting it. Of course, arithmetic coding would reach almost perfect entropic coding at the cost of requiring arithmetic computations of high precision floating-point numbers. We thus want to exploit the specificities of Gaussian variables to design a near entropic compression while retaining maximal efficiency.

5.1. **Preliminary information-theoretical analysis.** Let n be a positive integer and $X = (X_i)_{1 \leqslant i \leqslant n}$ be a sequence of independent variables drawn under the discrete Gaussian distribution of standard deviation σ (assumed to be larger than the smoothing parameter

TABLE 2. Bit security estimates for FALCON and MITAKA with small q and ellipsoidal Gaussians (compared to the original schemes), in dimension 512 and 1024. Security levels are given in pairs Classical/Quantum.

	FALCON-512				MITAKA-512			
	KeyRec	Forgery	Sig Size	Key Size	KeyRec	Forgery	Sig Size	Key Size
Original	133/117	123/108	666	896	133/117	102/89	710	896
Small $q = 1031$	132/116	122/108	490	704	132/116	99/87	540	704
Small $q = 521$	132/116	121/106	455	640	132/116	97/85	505	640
Small $q = 257$	130/114	118/104	425	576	130/114	94/82	475	576
Distortion $\gamma = 2$	132/116	123/108	540	896	132/116	101/89	590	896
Distortion $\gamma = 4$	132/116	122/107	475	896	132/116	98/87	525	896
Distortion $\gamma = 6$	131/115	119/105	440	896	131/115	95/84	490	896
Distortion $\gamma = 8$	128/113	116/102	410	896	128/113	92/81	460	896
Distortion $\gamma = 10$	125/110	113/99	390	896	125/110	88/78	441	896
	FALCON-1024				MITAKA-1024			
Original	272/239	284/250	1280	1792	272/239	233/205	1405	1792
Small $q = 1031$	272/239	280/246	932	1408	272/239	224/197	1160	1408
Small $q = 521$	269/237	275/242	870	1280	269/237	218/191	1000	1280
Small $q = 257$	264/233	268/235	805	1152	264/233	209/184	935	1152
Distortion $\gamma = 2$	271/239	284/250	1033	1792	271/239	230/202	1160	1792
Distortion $\gamma = 4$	270/237	278/245	905	1792	270/237	221/195	1035	1792
Distortion $\gamma = 6$	267/235	271/239	830	1792	267/235	213/187	960	1792
Distortion $\gamma = 8$	261/229	263/232	780	1792	261/229	204/180	905	1792

of $\mathbb{Z}$). The entropy of this random vector is (close to) $\mathscr{H} = \frac{n}{2}\left(1 + \log_2(2\pi\sigma^2)\right)$ Therefore for a given sample $\mathbf{x}$, an entropic code for this distribution should have a codeword of length:

$$L(\mathbf{x}) = \mathscr{H} - n\log_2\left(e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}\right) = \underbrace{\frac{n}{2}\left(1 + \log_2(2\pi) + \frac{\|\mathbf{x}\|^2}{\sigma^2\log(2)}\right)}_{:=H} + \underbrace{n\log_2(\sigma)}_{:=T}$$

The decomposition of this expression in the two main terms H, T (which we will refer to by *Head* and *Tail*) indicates two contributions, of different geometrical interpretations. The H part can be thought of as the σ-quantile where x landed, whereas the T part demonstrates that the $\log(\sigma)$ least significant bits of each coefficient behave as uniform variables in $[0, 2^\sigma]$, giving the position of x inside the quantile. This rough analysis invites

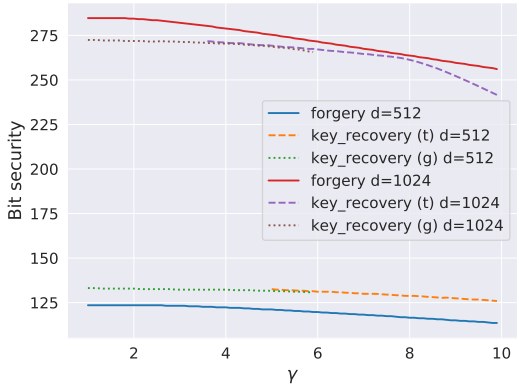


FIGURE 1. Bit-security in function of the Hamming weight, $q = 12289$

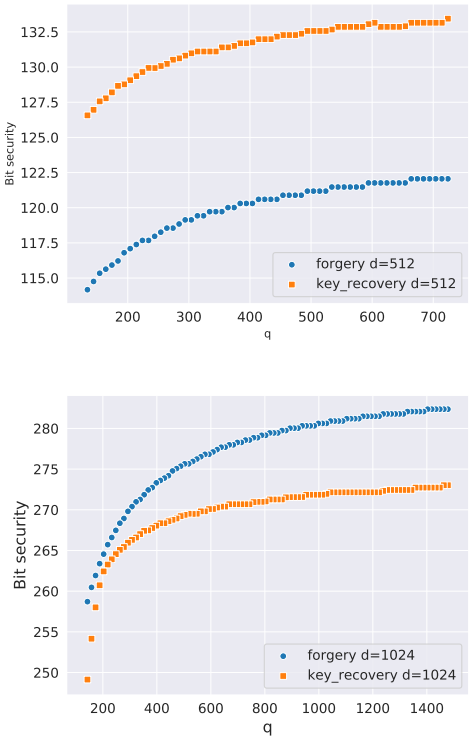


FIGURE 2. Bit-security in function of q, for $d = 512$ (left) and $d = 1024$ (right).

us to work modulo σ: we can not compress the $\log_2(\sigma)$-lower-order bits, but we can work on the most significant bits.

5.2. **Golomb-Rice style coding of a single variable.** This preliminary observation leads to a first natural algorithm, working coefficient-wise: we can not compress the reminder modulo σ, which remains in binary form. If the quotient behaves roughly like a discrete normal distribution of unit standard deviation, then we can encode its values using a Huffman encoding. We then stack the coded heads (as the Huffman code is prefix, we can decode the heads on the fly) and the tails together which are chunks of equal lengths. Remark that as the modulus part can not be compressed it will take up to $\lceil \log_2(\sigma) \rceil$-bits to be represented. As such, we will perform the euclidean division by $k = 2^{\lceil \log_2(\sigma) \rceil}$ instead of σ in order to maximize the information put in the tail and retain the least possible information in the head part. The following diagram presents an example of such encoding for on a sample (x_1, x_2, x_3), where $x_i = t_i + h_i k$:

This is very close to the so-called *Golomb-Rice* encoding of each coefficient with split at k.

5.3. **Batch-coding and full signature compression.** Since the signature can be (somehow) interpreted as d independent samples, we can compress them not individually but as a whole. We then want to encode in the most efficient way the message consisting of the d quotients. We propose here to rely on adaptive arithmetic encoding (or finite state entropy method), usually refered as Asymetric Numeral system (or ANS for short) of Duda[16]. The following diagram presents an example of such encoding for on a sample (x_1, x_2, x_3), where $x_i = t_i + h_i k$:

5.3.1. *Ranged arithmetic encoding.* Adapting the ranged version to our contexts, works as follow. Suppose that the distribution of the quotient part is a discrete Gaussian of standard deviation σ_q, denoted by $\mathcal{G}$ and of probability density function (pdf for short) ρ. As the size of signature is itself bounded by construction, we can truncate this distribution as well by a certain threshold T[8].

We also choose an *integral quantization factor* $p > 0$, and denote by f the quantized quotient distribution of $\mathcal{G}_{\leqslant T}$, that is to say its approximation at p bits of precision. More

[8]By truncating a discrete distribution $\mathcal{D}$ over $\mathbb{Z}$ of pdf p, we mean constructing the distribution $\mathcal{D}_{\leqslant T}$ of support $\{-T, \dots, T\} \cap \mathrm{Supp}(\mathcal{D})$ and of pdf $p_{\leqslant T}(x) = p(x) \left(\sum_{u=-T}^{T} p(u) \right)^{-1}$.

formally, we construct the distribution $\mathcal{G}^{(p)}_{\leqslant T}$ as the distribution of support $\{-T, \dots, T\}$ and of pdf $\tilde{\rho}$ proportional to $x \mapsto \lceil 2^p \rho_{\leqslant T}(x) \rceil 2^{-p}$.

Then denoting by $R(x) = \sum_0^x \tilde{\rho}(x)$ its cumulative distribution function, we define the symbol encoding function to be

$$s : \left| \begin{array}{ccc} [0; 2^{p-1}] & \longrightarrow & [0, 2T] \\ x & \longmapsto & \mathrm{argmin}_s\{F(s) \leqslant x < F(s+1)\}. \end{array} \right.$$

The coding function is now:

$C(x, s) = \left\lfloor \frac{x}{f(s)2^p} \ll n \right\rfloor + (x \mod f[s]) + F(s)$, The encoding of the (head) of the signature is then perfomed iteratively by (left)-folding the function C: for a sequence of integers $[s_1, \dots, s_n]$, define inductively $x_0 := 0$ and $x_{i+1} = C(x_i, s_i)$. The encoded sequence is then the integer x_n.

With this construction, the decoding function is now, denoting by & the bitwise and operator, $D(x) = (f(s)(x \gg n) + (x \,\&\, (2^n - 1)) - F(s), s)$, used again by left folding: given a compressed sequence represented as the integer x, we stream out the sequence $(s_i)_i$ defined inductively by $x_0 = x$, $(x_{i+1}, s_{i+1}) = D(x_i)$.

5.3.2. *ANS on the raw input.* As the distribution of the signature coefficient is public, we could use ANS encoding directly on the coefficients. This is of course possible and naturally would offer the best compression rates, but it would require to multiply larger numbers. Indeed, using the aforementioned separation only requires handling the head, which is encoded on a small integer, whereas a direct ANS would require to handle arithmetic with numbers of around $n/2 \log(2\pi\sigma^2)$ bits.

In addition, as the standard deviation of the quotients is small, the alphabet will be very limited and we also can use a *tabulated variant* to completely avoid arithmetic computations (or so-called finite-state-entropy methods).

5.4. **Nearly optimal encoding for hash-and-sign signatures.**

5.4.1. *Encoding of Falcon signatures.* For completeness, we recall the compression process used in the Falcon. The outline of the compression is quite similar to the one of Section 5.2, but the sign is taken out of the coefficient and encoded as a separated bit. As such, the quotient by σ is now following a folded-normal distribution. A careful study of this distribution reveals that the corresponding Huffman coding corresponds to the unary encoding of the variable. The following diagram presents an example of such encoding for on a sample (x_1, x_2), where $|x_i| = t_i + h_i k$ and $s_i = \mathrm{sgn}(x_i)$

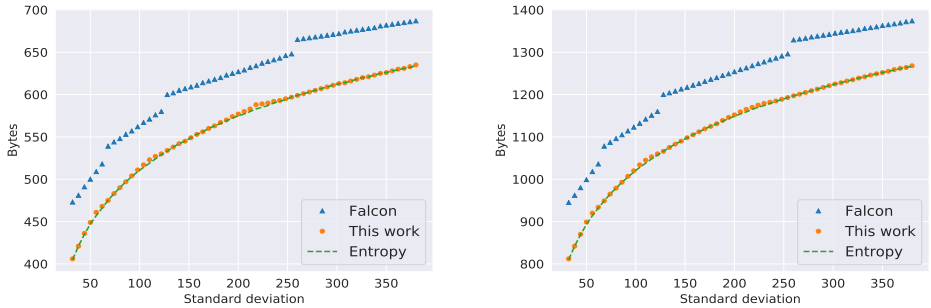


FIGURE 3. Comparison between FALCON and our encoding. The left figure is for $d = 512$, the right one is for $d = 1024$. They are computed using a quantization factor of $f = 16$ (i.e. 16-bits approximation of the density function of the distribution).

0^{h_1}	1	s_1	t_1	0^{h_2}	1	s_2	t_2

5.4.2. *Practical comparison with our method.* We exhibit a practical comparison of the compression performances between our encoding and Falcon's, together with the entropy lower bound. The experiments reveal that our technique is nearly optimal (standing at *at most* 3 bytes to the entropic limit). For dimension 512, we can save between 45 and 65 bytes compared to FALCON's Huffman-based coefficient-wise compression. In dimension 1024, the gaps now lie in between 80 and 130 bytes, which represents a total gain of 7%-14% on the signature size.

Acknowledgements. We thank the anonymous reviewers for their helpful comments. Yang Yu is supported by the National Natural Science Foundation of China (No. 62102216), the National Key Research and Development Program of China (Grant No. 2018YFA0704701), the Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008) and Major Scientific and Technological Innovation Project of Shandong Province, China (Grant No. 2019JZZY010133).

REFERENCES

[1] 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)

[2] 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).

[3] Beullens, W.: Improved cryptanalysis of UOV and rainbow. In: EUROCRYPT 2021. pp. 348–373 (2021)

[4] Chailloux, A., Loyer, J.: Lattice sieving via quantum random walks. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 63–91. Springer, Heidelberg (Dec 2021).

[5] Chen, Y., Genise, N., Mukherjee, P.: Approximate trapdoors for lattices and smaller hash-and-sign signatures. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 3–32. Springer, Heidelberg (Dec 2019).

[6] Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., Xagawa, K.: ModFalcon: Compact signatures based on module-NTRU lattices. In: Sun, H.M., Shieh, S.P., Gu, G., Ateniese, G. (eds.) ASIACCS 20. pp. 853–866. ACM Press (Oct 2020).

[7] Coja-Oghlan, A., Ergür, A.A., Gao, P., Hetterich, S., Rolvien, M.: The rank of sparse random matrices. In: Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms. p. 579–591. SODA '20, Society for Industrial and Applied Mathematics, USA (2020)

[8] De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Heidelberg (Dec 2020).

[9] Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y., Kannwischer, M., Patarin, J.: Rainbow: Submission to the nist's post-quantum cryptography standardization process, https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

[10] 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).

[11] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)

[12] Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: ASIACRYPT 2014. pp. 22–41 (2014)

[13] 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).

[14] Ducas, L., Prest, T.: Fast Fourier Orthogonalization. In: ISSAC 2016. pp. 191–198 (2016)

[15] Ducas, L., van Woerden, W.P.J.: NTRU fatigue: How stretched is overstretched? In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 3–32. Springer, Heidelberg (Dec 2021).

[16] Duda, J., Tahboub, K., Gadgil, N.J., Delp, E.J.: The use of asymmetric numeral systems as an accurate replacement for huffman coding. In: 2015 Picture Coding Symposium (PCS) (2015)

[17] Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., Yu, Y.: Mitaka: A simpler, parallelizable, maskable variant of falcon. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 222–253. Springer, Heidelberg (May / Jun 2022).

[18] Espitau, T., Kirchner, P.: The nearest-colattice algorithm: Time-approximation tradeoff for approx-cvp. ANTS XIV p. 251 (2020)

[19] Fouque, P.A., Kirchner, P., Pornin, T., Yu, Y.: Bat: Small and fast kem over ntru lattices. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022, Issue 2**, 240–265 (2022)

[20] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206 (2008)

[21] 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).

[22] Jia, H., Hu, Y., Tang, C.: Lattice-based hash-and-sign signatures using approximate trapdoor, revisited. IET Information Security **16**(1), 41–50 (2022)

[23] 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).

[24] 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).

[25] Lu, X., Liu, Y., Zhang, Z., Jia, D., Xue, H., He, J., Li, B., Wang, K.: Lac: Practical ring-lwe based public-key encryption with byte-level modulus. Cryptology ePrint Archive, Paper 2018/1009 (2018), https://eprint.iacr.org/2018/1009, https://eprint.iacr.org/2018/1009

[26] Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: ASIACRYPT 2009. pp. 598–616 (2009)

[27] Lyubashevsky, V.: Lattice signatures without trapdoors. In: EUROCRYPT 2012. pp. 738–755 (2012)

[28] Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D., Bai, S.: Dilithium: Submission to the NIST's post-quantum cryptography standardization process, https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

[29] 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).

[30] Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (Apr 2012).

[31] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM Journal on Computing **37**(1), 267–302 (2007)

[32] Micciancio, D., Walter, M.: Practical, predictable lattice basis reduction. In: EUROCRYPT 2016. pp. 820–849 (2016)

[33] Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 271–288. Springer, Heidelberg (May / Jun 2006).

[34] Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: CRYPTO 2010. pp. 80–97 (2010)

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

[36] Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Submission to the NIST's post-quantum cryptography standardization process, https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

[37] Rasmussen, R.: ICANN SSAC comment to NIST on quantum cryptography algorithms. Tech. rep. (Dec 2019), available at https://www.icann.org/en/system/files/files/sac-107-en.pdf

[38] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005).

[39] Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (May 2011).

[40] Westerbaan, B.: Sizing up post-quantum signatures (Sep 2021), available at https://blog.cloudflare.com/sizing-up-post-quantum-signatures/

[41] 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).

[42] Zhang, J., Yu, Y., Fan, S., Zhang, Z., Yang, K.: Tweaking the asymmetry of asymmetric-key cryptography on lattices: KEMs and signatures of smaller sizes. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part II. LNCS, vol. 12111, pp. 37–65. Springer, Heidelberg (May 2020).

Appendix A. Proof of Proposition 2.2

Let $\mathscr{D}(\mathbf{s})$ the probability that the algorithm outputs $\mathbf{s} = x_1\mathbf{b}_1 + x_2\mathbf{b}_2$, and $P(x_i)$ be the probability that Algorithm 1 outputs $x_i \in \mathscr{R}$. Let also $\Sigma_i = \tau_i^2 + r^2$, so that thanks to Proposition 2.1, we can write

$$\mathscr{D}(\mathbf{s}) = P(x_1)P(x_2) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2\right] \cdot D_{\mathscr{R},\Sigma_1^{-1},\widetilde{t_1}}(x_1) D_{\mathscr{R},\Sigma_2^{-1},\widetilde{t_2}}(x_2). \tag{13}$$

Let $\mathbf{U}$ be the transformation such that $\mathbf{B} = \widetilde{\mathbf{B}}\mathbf{U}$ and μ_{12} its top right coefficient. By construction, we have $\mathbf{t} = \widetilde{\mathbf{B}}(\widetilde{t_1} + x_2\mu_{12}, \widetilde{t_2})$ and $\mathbf{s} = \widetilde{\mathbf{B}}\mathbf{U}(x_1, x_2)$. We can then write over $\mathscr{K}_{\mathbb{R}}$

$$(\mathbf{s}-\mathbf{t})^* Q (\mathbf{s}-\mathbf{t}) = (x_1 - \widetilde{t_1}, x_2 - \widetilde{t_2})^* \widetilde{\mathbf{B}}^* Q \widetilde{\mathbf{B}} (x_1 - \widetilde{t_1}, x_2 - \widetilde{t_2}).$$

Because we orthogonalized $\mathbf{B}$ with respect to Q, we have $\widetilde{\mathbf{B}}^* Q \widetilde{\mathbf{B}} = \mathrm{diag}(\langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i \rangle_Q)$. With the equality above, this leads us to

$$\rho_{(\sigma\sigma^*)^{-1}Q,1}(\mathbf{s}-\mathbf{t}) = \rho_{\Sigma_1^{-1}}(x_1 - \widetilde{t_1}) \rho_{\Sigma_2^{-1}}(x_2 - \widetilde{t_2}).$$

By assumption on σ, the covariances Σ_1 and Σ_2 "are above" $\eta_\varepsilon(\mathscr{R})$, so using Lemma 2.2, Identity (13) becomes

$$\mathscr{D}(\mathbf{s}) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^4\right] \cdot \delta \cdot D_{\mathscr{M},Q,\sigma\sigma^*,\mathbf{t}}(\mathbf{s}),$$

where we let $\delta = \frac{\rho_{(\sigma\sigma^*)^{-1}Q,1}(\mathscr{M}-\mathbf{t})}{\rho_{\Sigma_1^{-1}}(\mathscr{R})\rho_{\Sigma_2^{-1}}(\mathscr{R})}$. The claims follow from routine computations. □

Algorithm 5: Key recovery in distorted setting

Input: an NTRU public key $h \in \mathscr{K}$, public parameters q, γ of the scheme
Result: (functionally equivalent) secret key (f, g)

$\mathbf{B} \leftarrow \begin{pmatrix} \mathbf{I}_d & \mathbf{H} & 0 \\ 0 & q\mathbf{I}_d & \mathbf{I}_d \end{pmatrix}$
while 1 **do**
  Sample a subset $\mathcal{I}$ of $\{1, \dots, d\}$ of cardinality ℓ
  $\mathbf{B}' \leftarrow \mathbf{B}[\bar{\mathcal{I}}]$
  $\mathbf{B}' \leftarrow \mathsf{BKZ}(\mathbf{B}, \beta, \|.\|_\gamma)$
  $\mathbf{P} \leftarrow \pi_{2d-\ell-\beta}(\mathbf{B}')$
  for $\tilde{x} \in \mathbf{P}\mathbb{Z}^{2d-\ell-\beta} \cap \mathbf{B}\left(0, \sqrt{4/3}\|\pi_{2d-\ell-\beta}(\mathbf{B}'[2d-\ell-\beta])\|_\gamma\right)$ **do**
    // Using Babai's Nearest plane
    Lift $\tilde{x}$ into $x \in \mathbf{B}'\mathbb{Z}^{2d-\ell}$
    $(f, g, k) \leftarrow x$
    if $fh - g = k = 0 \ mod \ q$ **then return** (f, g)
  end for
end while

Appendix B. Estimation of volume for Section 4.2.3

Let $\mu^2 = 1 + (\frac{\gamma\delta}{q})^2$. By definition of the algebraic norm, we have

$$\mathrm{N}_{\mathscr{K}/\mathbb{Q}}(\det(G)) = (\mu q)^{2d} \prod_{\varphi} \left(1 + \frac{\delta^2}{\gamma^2} \cdot \frac{|\varphi(h)|^2}{(\mu q)^2}\right),$$

where the product ranges over the complex field embeddings. Using the arithmetic-geometric inequality and the fact that $\sum_\varphi |\varphi(h)|^2 = d\|h\|^2$, we have

$$\mathrm{N}_{\mathscr{K}/\mathbb{Q}}(\det(G)) \leqslant (\mu q)^{2d} \left(1 + \frac{\delta^2}{\gamma^2} \cdot \frac{\|h\|^2}{(\mu q)^2}\right)^d.$$

We now assume that the coefficients of h behave like discrete uniform and independent variables in $[-\frac{q}{2}, \frac{q}{2}] \cap \mathbb{Z}$, so that $\mathbb{E}[\|h\|^2] = \frac{d(q^2-1)}{12}$. Next we use the convexity of $x \mapsto x^{\frac{1}{4d}}$ and Jensen's inequality to obtain

$$\mathbb{E}[\mathcal{V}] \leqslant \sqrt{q} \cdot \left(1 + \frac{(\gamma\delta)^2}{q^2} + \frac{\delta^2}{\gamma^2} \cdot \frac{d}{12}\right)^{1/4}. \tag{14}$$

Appendix C. Algorithmic description of the attack against small-modulus turns

Algorithm 5 sums up our new attack for key recovery.

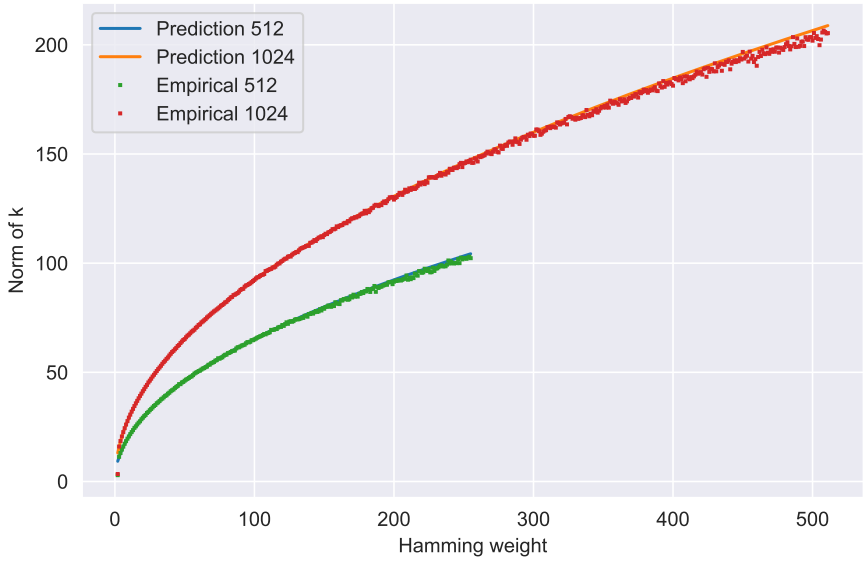


FIGURE 4. Comparison between the estimated size of k and the empirical data, where the mean is computed by generating 5000 independant secret keys and compute the correspinding k.

APPENDIX D. ADDITIONAL EXPERIMENTAL RESULTS

* NTT CORPORATION, TOKYŌ, JAPAN t.espitau@gmail.com, mehdi.tibouchi.br@hco.ntt.co.jp,
⋆ IRISA, UNIV RENNES 1, INRIA, BRETAGNE-ATLANTIQUE CENTER, FRANCE alexandre.wallet@inria.fr,
◇BNRIST, TSINGHUA UNIVERSITY, BEIJING, CHINA AND NATIONAL FINANCIAL CRYPTOGRAPHY RESEARCH CENTER, BEIJING, CHINA yu-yang@mail.tsinghua.edu.cn