Boosting the Hybrid Attack on NTRU: Torus LSH, Permuted HNF and Boxed Sphere

Phong Q. Nguyen[1,2⋆]

[1] Inria Paris, France. pnguyen@inria.fr
[2] Département d'informatique de l'ENS, ENS, CNRS, PSL University, Paris, France.

Abstract. We revisit collision attacks on NTRU, namely Odlyzko's meet-in-the-middle attack and Howgrave-Graham's hybrid attack. We show how to simplify and improve these attacks with respect to efficiency, analysis and ease of implementation. Our main ingredients are randomization and geometry: we randomize the attacks by introducing torus variants of locality sensitive hashing (LSH) and new HNF-like bases of the NTRU lattice; and we establish a connection between the success probability of the hybrid attack and the intersection of an n-dimensional sphere with a random box. We provide mathematical and algorithmic bounds for such intersections, which is of independent interest. Our new analyses remain partially heuristic, but are arguably much more sound than previous analyses, and are backed by experiments. Our results show that the security estimates of the NTRU finalist in NIST's post-quantum standardization need to be revised.

1 Introduction

Due to the on-going NIST standardization [24] of post-quantum cryptography and the development of fully-homomorphic encryption, it is crucial to analyze and design the best lattice algorithms. Out of the seven third-round candidates selected by NIST [24], five [4, 7, 9, 11, 12] are based on the hardness of lattice problems, such as finding short lattice vectors (SVP) and close lattice vectors (CVP). For efficiency reasons, many lattice-based schemes (including the five lattice-based finalists [24]) use secret vectors with very small coordinates, and possibly sparse. This kind of property can be exploited by collision attacks introduced against the NTRU public-key encryption scheme [16]: Odlyzko's meet-in-the-middle attack (described in [16, 18]) and Howgrave-Graham's hybrid attack [19], the latter combining the former with lattice reduction. It is therefore very important to analyze the efficiency of such collision attacks: for many parameters, the hybrid attack is considered the most powerful attack against the NTRU cryptosystem, by NTRU itself [15] and in the latest NIST submission [7, 8].

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 885394).

Unfortunately, these collision attacks, especially the hybrid attack, are still poorly understood (as first pointed out by Schanck [26] and later Wunderer [30]), for several reasons. First, the attacks are technical and not easy to implement: they are memory intensive, and the original description of [19] uses functions whose output is a list whose size varies depending on the input. The hybrid attack was surprisingly not used in any of the solved NTRU numerical challenges [27]. As a result, the literature [30, 6, 15, 19] on the hybrid attack has focused on theoretical analyses involving quite a few heuristics, but for which very limited experiments have been carried out to support them, if any. Second, all the analyses so far arguably fail to give a clear full picture. As an example, the recent analysis of the hybrid attack in NTRU's NIST submission [7, 8, Sect. 6.4.3] does not follow the analyses [30, 19].

Our results. We revisit Odlyzko's meet-in-the-middle attack (described in [16, 18]) and Howgrave-Graham's hybrid attack [19], which both exploit clever collisions to recover the NTRU secret key. We simplify and improve these attacks both in terms of efficiency and ease of implementation. Our main ingredients are randomization and geometry. We randomize several steps: this clarifies what depends on random coins used by the attacker and what depends on the secret key, and has the benefit of significantly increasing the success of the attack, as well as analyzing more rigorously the frequency of these collisions. In all previous versions of the hybrid attacks, there were keys for which the attack would always fail or require much more time than usual: such keys no longer exist here.

Our analysis removes the most annoying issues in previous analyses of the hybrid attack: informal arguments on "flipping bits" are replaced by a rigorous torus variant of locality sensitive hashing, "simplified" distributions are replaced by provable bounds on a natural modelling asking what proportion of the n-dimensional unit sphere is covered by a given box, *i.e.* a product of intervals.

Our work impacts at least the NTRU finalist in the NIST competition: the security analysis of [7] needs to be revised. We show that the concrete complexities given by the scripts of [7] for the hybrid attack are not reliable, mixing both overestimates and underestimates. For instance, for the `ntruhps2048677` parameter set, the success probability which was not estimated but assumed to be non-negligible is actually less than 2^{-46}, and at the same time, the meet-in-the-middle stage can be sped up by a factor roughly $160,000$.

Our most important results deal with the hybrid attack, which is much more powerful than Odlyzko's attack. However, because Odlyzko's attack is much simpler, it is very helpful as a case study, in order to introduce and explain more easily some of our key ideas.

Technical overview. At a high-level, a hybrid attack [19] can be explained as follows. Assume that one is looking for a very short secret vector $\boldsymbol{t} \in \mathbb{Z}^N$ whose coordinates are all very small integers. This target $\boldsymbol{t}$ belongs to a lattice, which has a basis of a special shape. Accordingly, the secret vector $\boldsymbol{t}$ is split into three parts $\boldsymbol{t}_1|\boldsymbol{t}_2|\boldsymbol{t}_3$ where the number of coordinates of $\boldsymbol{t_1}$, $\boldsymbol{t_2}$ and $\boldsymbol{t}_3$ are respectively N_1, N_2 and N_3 such that $N = N_1 + N_2 + N_3$:

1. The tail part $\boldsymbol{t}_3$ is guessed by meet-in-the-middle techniques. Depending on the exact distribution of $\boldsymbol{t}$, it is hoped that if $\boldsymbol{t}_3$ runs over a set L of s elements, one can build two lists L_1 and L_2 of roughly $\sqrt{s}$ elements, such that $\boldsymbol{t}_3 = \boldsymbol{u}_1 - \boldsymbol{u}_2$ for some $(\boldsymbol{u}_1, \boldsymbol{u}_2) \in L_1 \times L_2$. Instead of doing a full exhaustive search on $\boldsymbol{t}_3$ by enumerating over $L_1 \times L_2$, one wants to enumerate only over $\boldsymbol{u}_1 \in L_1$ and be able to detect the correct guess by recording collisions in a later stage, but the success of the detection depends on $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$.
2. The middle part $\boldsymbol{t}_2$ is taken care of by lattice reduction techniques, which are made possible by the special shape of the starting basis. Roughly speaking, the shorter $\boldsymbol{t}_2$ is and/or the stronger the lattice reduction algorithm is, the higher the success probability p. This p dictates our ability to detect the correct guess $(\boldsymbol{u}_1, \boldsymbol{u}_2) \in L_1 \times L_2$ by recording collisions, when enumerating over $\boldsymbol{u}_1 \in L_1$.
3. The head part $\boldsymbol{t}_1$ is tackled by error-correction techniques and also involves a success probability, but we will show that this success probability can be made significantly high, and is therefore much less problematic.

What makes the analysis cumbersome is that we need to analyze all possible choices of N_1, N_2 and N_3, and for each choice, we need to estimate the probability p as well as the running time and output quality of lattice reduction. Otherwise, it is not possible to determine the optimal choice of parameters. All descriptions of the hybrid attack adopted this framework, because they all relied on the Hermite normal form of the NTRU lattice: the only freedom was over the choice of N_1, N_2 and N_3 and the lists $L_1 \times L_2$.

Our key contributions are the following. First, we observe that this splitting $\boldsymbol{t}_1|\boldsymbol{t}_2|\boldsymbol{t}_3$ is sub-optimal because the target vector $\boldsymbol{t}$ in NTRU is unbalanced. More precisely, it is well-known that $\boldsymbol{t} = g|f$ can be split in two halves, where g and f are both binary or ternary polynomials but whose number of non-zero coefficients can vary significantly. In the historical version [16] as well as the latest versions [7, 8] of NTRU, g is significantly sparser than f: for instance, in the `ntruhps2048677` parameter set of NIST's NTRU finalist [7], f has nearly 1.8 times more non-zero coefficients than g. However, the initial splitting $\boldsymbol{t}_1|\boldsymbol{t}_2|\boldsymbol{t}_3$ forces $\boldsymbol{t}_3$ to be the tail of the f-part, and $\boldsymbol{t}_1$ to be the head of the g-part. Intuitively, the g-part is the easiest part of the target: it has smaller entropy than f. Thus, it is unclear why the meet-in-the-middle guessing should target f (*i.e.* the part of $\boldsymbol{t}$ with the highest entropy), rather than g. We stress that this was not the case when the hybrid attack was invented [19], because at that time, recommended NTRU parameters actually used an f which was sparser than g!

Furthermore, the probability p is strongly influenced by the norm of $\boldsymbol{t}_2$, which mixes both f and g. If we want to maximize p, it would be better if the middle part involved more coordinates of g. Finally, because it turns out that error-correcting techniques are the least problematic part of the attack, one wishes $\boldsymbol{t}_1$ to include the largest coordinates of $\boldsymbol{t}$, *i.e.* to involve as many coordinates of f as possible. In other words, current hybrid attacks (as described in [30, 7]) do not fully exploit the unbalanced structure of the NTRU secret key.

To improve the attack, we introduce randomness and exploit more bases of the NTRU lattice. Instead of decomposing deterministically $g|f$ as $\boldsymbol{t}_1|\boldsymbol{t}_2|\boldsymbol{t}_3$, we apply the three-part decomposition over well-chosen random permutations of the target $g|f$, in order to minimize the norm of the middle part $\boldsymbol{t}_2$, and to absorb the densest part of $\boldsymbol{t}$ into $\boldsymbol{t}_1$. But such a modification is not trivial, because it requires to find bases of the NTRU lattice with the right shape, and whose existence may not even be guaranteed: we can no longer use the original Hermite normal form. Fortunately, we observe that experimentally, for the overwhelming majority of permutations Π of $\mathbb{Z}^N$, the (row) Hermite normal form of the modified NTRU row lattice whose columns have been permuted by Π has the following crucial properties: the top-left block is q times the identity matrix, except possibly its very last bottom vectors, and the bottom-right block is the identity matrix, except possibly its very first front vectors. Permuting coordinates of the target allows us to mount an optimized hybrid attack which exploits the known structure of the distribution of the secret key. It also has the added benefit of making $\boldsymbol{t}_3$ as sparse as possible, and taking better advantage of the well-known cyclic rotations of NTRU, thereby decreasing the cost of the meet-in-the-middle stage. For instance, in the `ntruhps2048677` parameter set [7], the best non-local hybrid attack found in [7] used a meet-in-the-middle stage costing 2^{144}. With our randomization, this cost decreases to $2^{126.7}$, giving a $160,000\times$ speed-up.

Our second key contribution is to introduce randomness at another step of the hybrid attack, the "error-correction" step where we detect collisions by "absorbing" the front part $\boldsymbol{t}_1$. In all previous versions of the hybrid attack, the detection of collisions was done deterministically, which made implementations cumbersome (due to varying output length caused by flipped bits), and made it impossible to analyze rigorously the probability of collisions happening, because one had to argue that some fixed elements behaved randomly. To do so, we introduce a torus variant of locality-sensitive hashing (LSH) which we can analyze rigorously. The main idea behind our torus LSH is exactly the one underlying Odlyzko's collision attack except that we add a crucial randomization: if we split a torus at random into two halves, then any two close torus elements are very likely to belong to the same half. This allows us to check efficiently if an input vector is very close modulo q to a given list of vectors, in time logarithmic in the number of elements of the list.

Our third most important contribution deals with the middle part $\boldsymbol{t}_2$. Perhaps the biggest issue with all previous analyses of the hybrid attack is related to the probability p (known in the NTRU literature as the admissibility probability). Its value is critical to evaluate the efficiency of the attack, yet it was unclear in the literature how to efficiently approximate this probability: as a result, the NTRU submission [7, Sect. 6.4.3] even ignored p. Experimental methods and heuristic formulas were proposed, such as in [19, 6, 30], but very limited experimental evidence nor completely rigorous arguments were presented. We show that all previous efficient methods to assess p were somewhat questionable, potentially leading to exponential errors, and we present new ways to approximate

p. This is based on a geometric model of p as the probability that a random unit vector belongs to a certain box, *i.e.* a product of intervals. A related but distinct problem was studied by Aono and Nguyen [3]: approximating the volume of the intersection between a ball and a box. We provide rigorous and efficient methods to bound p, and use extreme value theory to better understand the asymptotical behaviour. It is well-known that the n-dimensional unit-sphere is contained in the box $[-1,1]^n$, and includes the box $[-1/\sqrt{n}, 1/\sqrt{n}]^n$. Our work addresses the problem of "framing" a sphere into a box from a probabilistic point of view: which boxes are likely to contain a significant proportion of the sphere? We show that for any fixed $\alpha > 0$, a random unit vector is very unlikely to belong to $[-\alpha/\sqrt{n}, \alpha/\sqrt{n}]^n$, but belongs to $[\sqrt{\frac{2\ln n}{n}}, \sqrt{\frac{2\ln n}{n}}]^n$ with at least constant probability. This is of independent interest: interestingly, it allows to bound the success probability of Babai's widely used nearest plane algorithm [5]. Given as input a basis B of an n-rank lattice L and a target vector $\boldsymbol{t}$ of the form $\boldsymbol{t} = \boldsymbol{v} + \boldsymbol{e}$ where $\boldsymbol{v} \in L$, Babai's algorithm returns $\boldsymbol{v}$ in polynomial time if $\|\boldsymbol{e}\| \leq r/2$ where is r is the minimal norm of the Gram-Schmidt orthogonalization of B. This bound is tight in the worst case but is known to be pessimistic in practice. Our work shows that if $\boldsymbol{e}$ is a random vector in the sphere or in the ball of radius m, then one can guarantee a constant success probability with a radius m essentially as large as $r\sqrt{\frac{n}{2\ln n}}$.

Related work. Wunderer [30] pointed out several issues and mistakes in previous descriptions or uses of the hybrid attack, showing that several security estimates were not reliable. However, his analysis did not significantly differ from the original analysis of Howgrave-Graham [19], leaving several issues unsolved. Hoffstein*et al.* [17] briefly adapted the hybrid attack to target g (rather than f), but only in the case where g is invertible in the ring mod q, which is not the case for NTRU's NIST submission [7], nor for the original NTRU [16].

Roadmap. Sect. 2 provides background. Sect. 3 revisits Odlyzko's attack [18] on NTRU using LSH. Sec. 4 introduces HNF-like bases arising from permutations, which are useful for our randomization of the hybrid attack. Sect. 5 presents our randomized hybrid attack incorporating both LSH and permuted HNF, with the new analysis based on boxed spheres. Sect. 6 deals with the probability that a random unit vector belongs to a box: can we frame most random unit vectors into a small box? In Sect. 7, we report experimental results and discuss the security analysis of the NTRU finalist [7].

2 Preliminaries

General. $\mathbb{N}$ is the set of integers ≥ 0. For any finite set U, its number of elements is $\#U$. For any measurable subset $S \subseteq \mathbb{R}^n$, its volume is $\mathrm{vol}(S)$. We use row representations of matrices. The Euclidean norm of a vector $\boldsymbol{v} \in \mathbb{R}^n$ is $\|\boldsymbol{v}\|$. We denote by $\mathrm{Ball}_n(\boldsymbol{c}, R)$ the n-dim Euclidean ball of radius R and center $\boldsymbol{c}$, whose volume is $\mathrm{vol}(\mathrm{Ball}_n(R)) = R^n \frac{\pi^{n/2}}{\Gamma(n/2+1)}$. If $\boldsymbol{c}$ is omitted, we mean $\boldsymbol{c} = 0$.

Lattices. A *lattice* L is a discrete subgroup of $\mathbb{R}^m$, or equivalently the set $L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{\sum_{i=1}^n x_i \boldsymbol{b}_i \ : \ x_i \in \mathbb{Z}\}$ of all integer combinations of n linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^m$. Such $\boldsymbol{b}_i$'s form a *basis* of L. All the bases have the same number n of elements, called the dimension or rank of L, and the same n-dimensional volume of the parallelepiped $\{\sum_{i=1}^n a_i \boldsymbol{b}_i \ : \ a_i \in [0,1)\}$ they generate. We call this volume the co-volume of L, denoted by $\mathrm{covol}(L)$. The lattice L is said to be *full-rank* if $n = m$. The *shortest vector problem* (SVP) asks to find a non-zero lattice vector of minimal Euclidean norm. The *closest vector problem* (CVP) asks to find a lattice vector closest to a target vector.

Orthogonalization. For a basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of a lattice L and $i \in \{1, \ldots, n\}$, we denote by π_i the orthogonal projection on $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1})^\perp$. The *Gram-Schmidt orthogonalization* of the basis B is defined as the sequence of orthogonal vectors $B^\star = (\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_n^\star)$, where $\boldsymbol{b}_i^\star := \pi_i(\boldsymbol{b}_i)$. $\pi_i(L)$ is a lattice of rank $n+1-i$ generated by $\pi_i(\boldsymbol{b}_i), \ldots, \pi_i(\boldsymbol{b}_n)$, with $\mathrm{covol}(\pi_i(L)) = \prod_{j=i}^n \left\|\boldsymbol{b}_j^\star\right\|$.

Sublattices and Quotient Lattices. Let $L \subseteq \mathbb{R}^m$ be an n-rank lattice. A *sublattice* of L is any subgroup M of L. A sublattice M is said to be *pure* or *primitive* if L/M is torsion-free, which is equivalent to the existence of a subspace E of $\mathbb{R}^m$ such that $M = L \cap E$, which is also equivalent to the existence of a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of L such that $M = L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-1})$ for some integer $k \in \{1, \ldots, n\}$. Then the quotient group L/M can be viewed as an $(n-k+1)$-rank lattice, isomorphic to $\pi_k(L)$. So the projected lattices $\pi_k(L)$ can be viewed as an implementation of quotient lattices. One advantage of viewing a projected lattice as a quotient lattice is that if two lattice vectors $\boldsymbol{u}, \boldsymbol{v} \in L$ have the same projection $\pi_k(\boldsymbol{u}) = \pi_k(\boldsymbol{v})$, then we immediately see that $\boldsymbol{u} - \boldsymbol{v} \in M$, *i.e.*$\boldsymbol{u} \equiv \boldsymbol{v}$ modulo M.

Statistics. We denote by $\mathbb{E}()$ the expectation and $\mathbb{V}()$ the variance of a random variable. The CDF of the Gaussian distribution of expectation 0 and variance σ^2 is $\frac{1}{2}(1+\mathrm{erf}(\frac{x}{\sigma\sqrt{2}}))$ where the error function is $\mathrm{erf}(z) := \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$. The normal distribution $\mathcal{N}$ is the special case $\sigma = 1$. The multivariate Gaussian distribution over $\mathbb{R}^m$ of parameter σ selects each coordinate with Gaussian distribution.

Locality Sensitive Hashing (LSH). It was introduced in a breakthrough work by Indyk and Motwani [20] to find approximate neighbours: LSH is a family of hash functions such that close inputs are likely to have the same hash and far inputs are unlikely to have the same hash. We use the following definition. LSH is any samplable distribution $\mathcal{H}$ of hash functions over a set E. This induces a similarity measure: for any $(a, b) \in E^2$, we let $\sigma(a, b)$ be the probability that $H(a) = H(b)$ for a random H from $\mathcal{H}$. The LSH is useful if there are real $p_1 > p_2$ such that $\sigma(a, b) \geq p_1$ if a and b are close to each other, and $\sigma(a, b) \leq p_2$ otherwise. Once $p_1 > p_2$, it is possible to arbitrarily increase the gap by classical techniques [20], using for instance several independent hash functions.

The NTRU cryptosystem. The NTRU cryptosystem [16], proposed by Hoffstein, Pipher and Silverman, works in the ring $\mathcal{R} = \mathbb{Z}[X]/(X^n - 1)$. An element $F \in \mathcal{R}$ is seen as a polynomial or a row vector: $F = \sum_{i=0}^{n-1} F_i x^i = [F_0, F_1, \ldots, F_{n-1}]$. To select keys, one uses the set $\mathcal{L}(d_1, d_2)$ of polynomials $F \in \mathcal{R}$ such that d_1 coefficients are equal to 1, d_2 coeffients are equal to -1, and the rest are zero. Depending on the NTRU instantiation, d_2 might actually set to zero. There are two small coprime moduli $p < q$, such as $q = 128$ and $p = 3$.

Historically, the secret keys were $f \in \mathcal{L}(d_f, d_f - 1)$ and $g \in \mathcal{L}(d_g, d_g)$ for some integers d_f and d_g significantly smaller than n, but other NTRU instantiations [15, 17, 8, 7] use different parameters for $\mathcal{L}$, such as binary polynomials $\mathcal{L}(d, 0)$. Here, we focus on the NTRUHPS parameters of NTRU's NIST submission [7], one of the seven finalists: f is a random polynomial in $\{0, \pm 1\}^n$, and $g \in \mathcal{L}(d_g, d_g)$ where $2d_g = q/8 - 2$. With high probability, f is invertible mod q. The public key $h \in \mathcal{R}$ is defined as $h = g/f \mod q$. Thus, in the ring $\mathcal{R}/q\mathcal{R}$ which we represent by $\mathbb{Z}_q^n$, we have $f * h = g$. In this article, it is not required to know how NTRU encryption or signature works. The polynomial h defines the so-called NTRU lattice Λ_h, formed by all $(u, v) \in \mathcal{R}^2$ such that $v * h \equiv u \mod q$. Here, we follow the definition of [19], but other papers may use a variant of Λ_h, using a permutation of the coordinates. Λ_h is generated by the rows of the following lower-triangular matrix, which is its Hermite normal form:

$$\begin{bmatrix} q & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & q & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ h_0 & h_1 & \cdots & h_{n-1} & 1 & 0 & \cdots & 0 \\ h_{n-1} & h_0 & \cdots & h_{n-2} & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\ h_1 & \cdots & h_{n-1} & h_0 & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

The lattice Λ_h contains by definition the following set of n secret short vectors $\mathcal{S}_h = \{(x^i * g, x^i * f), 0 \leq i \leq n-1\}$ formed by the secret vector (g, f) and its $n - 1$ rotations . In this paper, we choose the same NTRU lattice as in [19].

3 Randomizing Odlyzko's NTRU Attack with LSH

3.1 Odlyzko's Attack

Odlyzko's attack [18, 19] is a clever time/memory trade-off, which was one of the first attacks on NTRU. Recall that the public key h of NTRU is of the form $h \equiv g/f$ in $\mathcal{R}/q\mathcal{R}$, where the secret keys f and g are small polynomials chosen uniformly at random from respectively two public subsets $\mathcal{F}$ and $\mathcal{G}$ of $\mathcal{R}/q\mathcal{R}$. The simplest attack is exhaustive search: for every $(f', g') \in \mathcal{F} \times \mathcal{G}$, output (f', g') if and only if $g' \equiv h * f'$ in $\mathcal{R}/q\mathcal{R}$. This naive attack can be sped up by restriction to $\mathcal{F}$, thanks to the special shape of $\mathcal{G}$: for every $f' \in \mathcal{F}$, output $(f', h * f')$ if and only if $h * f' \in \mathcal{G}$. This requires $\#\mathcal{F}$ polynomial-time operations and negligible

space. Instead, Odlyzko's attack requires roughly $\sqrt{\#\mathcal{F}}$ time heuristically and $\sqrt{\#\mathcal{F}}$ space rigorously.

Because of the special shape of $\mathcal{F}$, one actually knows two subsets $\mathcal{F}_1$ and $\mathcal{F}_2$ of $\mathcal{R}/q\mathcal{R}$, such that $\#\mathcal{F}_1$ and $\#\mathcal{F}_2$ are both approximately $O(\sqrt{\#\mathcal{F}})$ and f or one of its rotations is equal to $f_1 - f_2$ for some unknown $(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$. Then $h * f \equiv g$ implies that $h * f_1 - h * f_2 \equiv g \pmod q$ In other words, $h * f_1$ is close to $h * f_2$ in the ring $\mathcal{R}/q\mathcal{R}$, because $h * f_1 \equiv g + h * f_2$ where g is small. To detect whether a given $h * f_2$ is close to $\{h * f_1, f_1 \in \mathcal{F}_1\}$, Odlyzko introduced a special deterministic hash function H [18]. More precisely, let $H : \mathcal{R}/q\mathcal{R} \to \{0,1\}^n$ with $H(\sum_{k=0}^{n-1} x_i X^i) = (\sigma(x_1), \ldots, \sigma(x_n))$, where $\sigma : \mathbb{Z}/q\mathbb{Z} \to \{0,1\}$ is defined by $\sigma(x) = 0$ if and only if $(x \bmod q) \leq \lfloor q/2 \rfloor - 1$. Then one precomputes the list $\mathcal{L} = \{(f_1, H(h * f_1)), f_1 \in \mathcal{F}_1\}$ and sorts it so that any collision in the second coordinate $H(h*f_1)$ can be detected in logarithmic time. Finally, for each $f_2 \in \mathcal{F}_2$, compute $h * f_2$ and all the possibilities $H_1, \ldots, H_m$ for $H(g + h * f_2)$:

1. Represent all the coefficients of $h * f_2$ in $\{0, \ldots, q-1\}$.
2. The number m is equal to 2^k where k is the number of coefficients of $h * f_2$ in $\{\lfloor q/2 \rfloor - 1, q-1\}$. Let $i_1, \ldots, i_k$ be the indices of these coefficients.
3. Then $H_1, \ldots, H_m$ are computed by flipping the k bits of $H(h*f_2)$ at positions $i_1, \ldots, i_k$ in any possible manner.
4. Check collisions with $\mathcal{L}$: if $H_i = H(h*f_1)$ for some $(f_1, H(h*f_1)) \in \mathcal{L}$, check if $f_1 - f_2$ is a valid secret key.

Although the attack is simple, it is surprisingly not easy to analyze its running time rigorously: the analysis of [18] is informal, and makes several implicit assumptions. In 2016, van Vredendaal [29] tried to make the analysis more formal, but we observe that [29, Lemma 1]and its proof are actually incorrect: see the appendix.

Clearly, any correct decomposition $f = f_1 - f_2$ will be output by the algorithm. The problem is to make sure that the number of operations is not much bigger than $\sqrt{\#\mathcal{F}}$:

- the correct $H(h * f_1)$ must not collide with too many other elements of $\mathcal{L}$: in the worst case, all elements of $\mathcal{L}$ might actually have the same hash $H(h*f_1)$, which would imply that any collision found in Step 4 already requires $\sqrt{\#\mathcal{F}}$ operations, and that there might be many collisions.
- the number m must never be too big, so that we can bound the number of collisions in Step 4. This would require to analyze the distribution of $h * f_2 \in \mathcal{R}/q\mathcal{R}$, which is non-trivial, because h and f_2 are not independent. Indeed, h is determined by (f, g).

We thus need to make heuristic assumptions, because H is deterministic and few interesting results are known on the distribution of $h * f_1$ or $h * f_2$. And because of the flipping bits issue, implementing the attack is not straightforward: this is why the implementation of [29] restricted the attack to special no-flipping keys. If we are lucky, both m and the number of collisions in $\mathcal{L}$ are polynomial, then the total running time would be $(\log \#\mathcal{F})\sqrt{\#\mathcal{F}}$ polynomial-time operations as announced.

We modify the attack to make it less heuristic and easier to implement. The key idea is to randomize the algorithm by randomizing the hash function H: to do so, we introduce locality sensitive hashing (LSH) on the discrete torus $\mathbb{Z}_q^n$, which prevents all the complications related to flipping bits.

3.2 LSH over a discrete torus

We start with the one-dimensional torus: the set $\mathbb{Z}_q$ of integers modulo q. For any $u \in \mathbb{Z}_q$, let $\sigma_u : \mathbb{Z}_q \to \{0,1\}$ defined by $\sigma_u(x) = 0$ if and only if:

$$(x-u) \bmod q \leq \lfloor q/2 \rfloor - 1.$$

Thus, σ_u maps $\lfloor q/2 \rfloor$ "consecutive" elements (starting from u) of $\mathbb{Z}_q$ into 0, and the remaining $q - \lfloor q/2 \rfloor = \lceil q/2 \rceil$ elements are mapped into 1. If q is a power of two, $\sigma_u(x)$ is simply the most significant bit of $(x-u) \bmod q$.

Lemma 1. *Let $a, b \in \mathbb{Z}_q$ where $q \geq 2$. Let c be the smallest residue of $b-a$ in absolute value: $c = \min_{k\in\mathbb{Z}} |b-a-kq| \leq \lfloor q/2 \rfloor$. Let u be chosen uniformly at random from $\mathbb{Z}_q$. Then:*

$$\Pr_u(\sigma_u(a) \neq \sigma_u(b)) = \frac{2c}{q}.$$

Proof. Without loss of generality, we may assume that $c = b - a - kq$ for some $k \in \mathbb{Z}$, otherwise we swap a and b. Then we choose suitable residues: there exist $\alpha \in \{0, \ldots, q-1\}$ and an integer $\beta \geq \alpha$ such that $c = \beta - \alpha$ and modulo q, $\beta \equiv b$ and $\alpha \equiv a$.

If we represent u by a residue in $\{0, \ldots, q-1\}$ and look at the function σ_u over the positive integers, it is 1 over $\{0, \ldots, u-1\}$, then 0 over $\{u, u+1, \ldots, u+\lfloor q/2 \rfloor - 1\}$, then 1 again over $\{u+\lfloor q/2 \rfloor, \ldots, u+q-1\}$, and so on. This means that $\sigma_u(\alpha) \neq \sigma_u(\beta)$ if and only if one of the indexes where σ_u changes is $> \alpha$ and $\leq \beta$, that is, $\alpha < u \leq \beta$ or $\alpha < u + \lfloor q/2 \rfloor \leq \beta$. Note that the two events $\alpha < u \leq \beta$ or $\alpha < u + \lfloor q/2 \rfloor \leq \beta$ are incompatible because $\beta - \alpha = c \leq \lfloor q/2 \rfloor$. And each of these two events has probability c/q, because $\beta - \alpha = c$. □

Corollary 1 *Under the same assumptions, if $c \in \{0,1\}$, then $\Pr_u(\sigma_u(a) = \sigma_u(b)) \geq 1 - 2/q$. Otherwise, $c \geq 2$ and $\Pr_u(\sigma_u(a) = \sigma_u(b)) \leq 1 - 4/q$.*

Since $1 - 2/q > 1 - 4/q$, σ_u defines a useful LSH over $\mathbb{Z}_q$, if we say that a and b are close modulo q if and only if the smallest residue of $a - b$ is in $\{0, \pm 1\}$.

We deduce a family $\mathcal{H}(q,n)$ of hash functions over the discrete torus $\mathbb{Z}_q^n$. To sample from $\mathcal{H}(q,n)$, we choose $u_1, \ldots, u_n \in \mathbb{Z}_q$ independently and uniformly at random. Then we define $H : \mathbb{Z}_q^n \to \{0,1\}^n$ by $H(x_1, \ldots, x_n) = (\sigma_{u_1}(x_1), \ldots, \sigma_{u_n}(x_n))$. The following is a direct consequence of LSH over $\mathbb{Z}_q$:

Theorem 1. *Let $q \geq 2$. Let $\boldsymbol{a}$ and $\boldsymbol{b} \in \mathbb{Z}_q^n$. Let $e_i = \min_{k\in\mathbb{Z}} |b_i - a_i - kq|$ for $1 \leq i \leq n$. Let H be chosen uniformly at random from $\mathcal{H}(q,n)$. Then:*

$$\Pr_H(H(\boldsymbol{a}) = H(\boldsymbol{b})) = \prod_{i=1}^{n} (1 - \frac{2e_i}{q}).$$

Proof. $H(\boldsymbol{x})$ is of the form $(\sigma_{u_1}(x_1), \ldots, \sigma_{u_n}(x_n))$ where the u_i's are chosen independently and uniformly at random from $\mathbb{Z}_q$. Since the events $\sigma_{u_i}(a_i) = \sigma_{u_i}(b_i)$ are independent for $1 \leq i \leq n$, the result follows from Lemma 1. □

Thus, the family $\mathcal{H}(n, q)$ defines the similarity measure:

$$\sigma(\boldsymbol{a}, \boldsymbol{b}) = \prod_{i=1}^{n} (1 - \frac{2e_i}{q}).$$

If ω is the Hamming weight of $\boldsymbol{a} - \boldsymbol{b}$ in $\mathbb{Z}_q^n$, (*i.e.* the number of non-zero coefficients), then $\sigma(\boldsymbol{a}, \boldsymbol{b}) \leq (1-2/q)^\omega$. If we further assume that each of the non-zero coefficients of $\boldsymbol{a} - \boldsymbol{b}$ is $\equiv \pm 1$ modulo q, then $\sigma(\boldsymbol{a}, \boldsymbol{b}) = (1 - 2/q)^\omega$.

Th. 1 explains why [29, Lemma 1] and its proofs are incorrect: the "probability" studied by the proof of [29, Lemma 1] is claimed to be $(1-d/(nq))^n \approx e^{-d/q}$, where g has exactly d coefficients equal to 1, and all other coefficients equal to zero. But Th. 1 shows that this probability should actually be $(1 - 2/q)^d$: see the appendix for more details on [29, Lemma 1].

Note that the family $\mathcal{H}(q, n)$ has some uniformity property:

Lemma 2. *Let H be in $\mathcal{H}(q, n)$. Let $\boldsymbol{y} \in \{0, 1\}^n$ of Hamming weight k. The number of preimages $\boldsymbol{x} \in \mathbb{Z}_q^n$ such that $\boldsymbol{y} = H(\boldsymbol{x})$ is $(q/2)^n$ if q is even, and $\lfloor q/2 \rfloor^k \lceil q/2 \rceil^{n-k} = (q-1)^k (q+1)^{n-k} 2^{-n}$ if q is odd.*

Proof. Indeed, for any $u \in \mathbb{Z}_q$, the coordinate function σ_u has $\lfloor q/2 \rfloor$ preimages of 0, and $q - \lfloor q/2 \rfloor = \lceil q/2 \rceil$ preimages of 1. □

It follows from Lemma 2 that if $\boldsymbol{a} \in \mathbb{Z}_q^n$ and $H \in \mathcal{H}(q, n)$ are fixed, and $\boldsymbol{b} \in \mathbb{Z}_q^n$ is chosen uniformly at random, the probability that $H(\boldsymbol{a}) = H(\boldsymbol{b})$ is always extremely close to $(q/2)^n$.

3.3 Application to NTRU

Algorithm 1 Simplifying Odlyzko's attack with LSH

Input: An NTRU public key (h, q), two sets $\mathcal{F}_1$ and $\mathcal{F}_2$ such that the corresponding secret key f or one of its rotations belongs to $\{f_1 - f_2, (f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2\}$, and an efficient membership test for the set $\mathcal{G}$ containing the other secret key g.

Output: A list of pairs $(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$ such that $h * (f_1 - f_2) \in \mathcal{G}$.

1: Select a random hash function H from the LSH family $\mathcal{H}(q, n)$ and let $\mathcal{R} = \mathbb{Z}[X]/(X^n - 1)$ be the NTRU ring.
2: Compute and sort the list $\mathcal{L} = \{(f_1, H(h*f_1)), f_1 \in \mathcal{F}_1\}$ where $h*f_1 \in \mathcal{R}/q\mathcal{R}$ so that collisions over $H(h * f_1)$ can be detected in logarithmic time.
3: **for** $f_2 \in \mathcal{F}_2$ **do**
4: Compute $H(h * f_2)$ where $h * f_2 \in \mathcal{R}/q\mathcal{R}$.
5: **for** each collision $(f_1, H(h * f_1)) \in \mathcal{L}$ such that $H(h * f_1) = H(h * f_2)$ **do**
6: Return (f_1, f_2) if $h * (f_1 - f_2) \in \mathcal{G}$.
7: **end for**
8: **end for**

Theorem 1 implies the correctness of Alg. 1, which simplifies Odlyzko's algorithm using our discrete-torus LSH.

Theorem 2. *Let (h, f, g, q) be an NTRU key such that f or one of its rotations belongs to $\{f_1 - f_2, (f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2\}$. Let ω be the number of non-zero coefficients of g and assume that all these non-zero coefficients are ± 1. Then, over the choice of H, the probability that Alg. 1, given as input $(h, q, \mathcal{F}_1, \mathcal{F}_2)$, outputs at least one pair (f_1, f_2) such that $h * (f_1 - f_2) \in \mathcal{G}$ is $\geq (1 - 2/q)^\omega$.*

Proof. There exists $(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$ such that f or one of its rotations is equal to $f_1 - f_2$. For such a pair, the probability (over the H chosen by Alg. 1) that $H(h * f_1) = H(h * f_2)$ is $\geq (1 - 2/q)^\omega$. If this event holds, the pair (f_1, f_2) will be found in Step 5 of Alg. 1, and therefore be output. □

For instance, for the old NTRU parameter set called `ees251ep6` and studied in [19], the lower bound $(1 - 2/q)^\omega$ is $\approx 28\%$: it thus suffices to run Alg. 1 four times with different H to have a high probability of returning a solution. The fact that this probability is non-negligible is actually not surprising. Asymptotically, the NTRU parameters satisfy $q = \Theta(n)$ and $\omega \leq n$ so $(1 - 2/q)^\omega \geq \Omega(1)$. Thus, the probability in Th. 2 is lower-bounded by a constant, so a constant number of executions of Alg. 1 is sufficient to have an overwhelming success probability.

It remains to analyze the running time of Alg. 1. This will require heuristics, but significantly less problematic ones than in the original Odlyzko's attack. Consider the list $\mathcal{L}$ built by Alg. 1. We model this construction by the classic balls into bins problem, as if we were placing $\#\mathcal{F}_1$ balls into 2^n bins derived

from H. Then the complexity of Alg. 1 is $O((\#\mathcal{F}_1 + m\#\mathcal{F}_2)\log(\#\mathcal{F}_1))$ poly-time operations, where m is the maximal load, *i.e.* the maximal number of elements in a bin. If $\#\mathcal{F}_1 \leq 2^n$ (which typically holds for NTRU parameters) and all the $H(h * f_1)$'s were independent and uniformly distributed, then it is well-known that with high probability, we would have $m = O(\frac{n}{\log n})$. In order for Alg. 1 to be efficient, it is sufficient to assume that m is polynomial in n for a random H, which is much weaker than $m = O(\frac{n}{\log n})$. Specifically, we can make the following heuristic assumption: Let (f, g, h) be an NTRU key. If a torus-LSH H is chosen uniformly at random, then with overwhelming probability, the maximal number of elements in a bin is polynomial in n.

4 Permuted HNF

The Hermite normal form (HNF) of a full-rank integer lattice $L \subseteq \mathbb{Z}^m$ is the unique basis whose row matrix $H = (h_{i,j})_{1 \leq i,j \leq m}$ is lower-triangular such that all the diagonal coefficients are > 0 and all off-diagonal coefficients are ≥ 0 and strictly less than the diagonal coefficient of their column: in other words, $0 \leq h_{i,j} \leq h_{j,j}$. It is well-known that the HNF of the NTRU lattice Λ_h, formed by all $(u, v) \in \mathcal{R}^2$ such that $v * h \equiv u \mod q$ is the following matrix:

$$H = \begin{bmatrix} q & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ 0 & q & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & & \vdots \\ 0 & \cdots & 0 & q & 0 & \cdots & \cdots & 0 \\ h_0 & h_1 & \cdots & h_{n-1} & 1 & 0 & \cdots & 0 \\ h_{n-1} & h_0 & \cdots & h_{n-2} & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\ h_1 & \cdots & h_{n-1} & h_0 & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

In this section, we introduce permutation variants of the HNF which will help to improve the efficiency of the hybrid attack, through randomization.

Let σ be a permutation of $\{1, \ldots, m\}$. For any square $m \times m$ matrix M, we denote by $M_{[\sigma]}$ the $m \times m$ matrix obtained by permuting the n columns of M using σ: if $M = (m_{i,j})$ then the (i, j)-th entry of $M_{[\sigma]}$ is $m_{i,\sigma(j)}$. For any square matrix H, we denote by H_σ the following permuted HNF matrix:

1. Let H' be the HNF of $H_{[\sigma]}$.
2. Let $H_\sigma = H'_{[\sigma^{-1}]}$. It is a basis of the lattice spanned by the rows of H.

Notice that the matrix H has a very special structure: it can be split into four square blocks, where the diagonal blocks are diagonal matrices (respectively q times the identity matrix, and the identity matrix itself), and the non-zero off-diagonal block is circulant. This structure is very important for the hybrid attack.

However, we observe experimentally that for most permutations σ, the matrix H also has a very special structure, sufficient to mount a hybrid attack. To give more intuition on this phenomenon, we start with a particular case.

4.1 The Reverse HNF Basis

In typical instantiations of NTRU, the secret polynomial g is such that $g(1) \equiv 0 \pmod q$, which prevents g and h from being invertible in the polynomial ring mod q. However, it is known [22] that h is likely to be *pseudo-invertible*, which means that there would exist a polynomial $\tilde{h}$ such that $h * \tilde{h} * t \equiv t$ in the ring modulo q for any polynomial t such that $t(1) \equiv 0 \pmod q$. In that case, we can identify the permuted HNF corresponding to the reverse permutation:

Lemma 3. *Let $\mathcal{R} = \mathbb{Z}[X]/(X^n - 1)$ Let $(f, g) \in \mathcal{R}^2$ such that f is invertible mod q and $g(1) \equiv 0 \ (mod q)$. Let $h \in \mathcal{R}$ defined as $h = g/f \mod q$. Let $\sigma = (2n \ 2n-1 \ \cdots \ 1)$ be the reverse permutation over $\{1, \ldots, 2n\}$ and H be the HNF of the NTRU lattice Λ_h. If h is pseudo-invertible with pseudo-inverse $\tilde{h}$, then H_σ is the following lower anti-triangular matrix:*

$$H_\sigma = \begin{bmatrix}
0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & 0 & q \\
\vdots & & & \vdots & \vdots & \ddots & 0 & q & 0 \\
\vdots & & & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & 0 & 0 & q & 0 & \cdots & 0 \\
0 & \cdots & \cdots & 0 & 1 & 1 & \cdots & 1 & 1 \\
0 & \cdots & 0 & q & 0 & 0 & \cdots & \cdots & 0 \\
\vdots & \ddots & 1 & q-1 & a_{0,0} & a_{0,1} & \cdots & & a_{0,n-1} \\
0 & \ddots & 0 & q-1 & a_{1,0} & a_{1,1} & \cdots & & a_{1,n-1} \\
0 & 1 & \ddots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
1 & 0 & \cdots & 0 & q-1 & a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1}
\end{bmatrix},$$

where the polynomial $a_i(X) = \sum_{j=0}^{n-1} a_{i,j} X^j$ satisfies $a_i(X) \equiv \tilde{h}(X) * (X^i - X^{n-1})$ modulo q.

Proof. Since this matrix has determinant q^n, which is equal to the co-volume of Λ_h, it suffices to show that all row vectors belong to Λ_h. Clearly, the first $n-1$ row vectors belong to Λ_h because they are q-vectors. The n-th row also belongs to Λ_h because $h(1) \equiv 0 \bmod q$. For any $t(X) = X^i - X^{n-1}$, we have $t(1) \equiv 0 \bmod q$, so $h * \tilde{h} * (X^i - X^{n-1}) \equiv X^i - X^{n-1}$ and therefore $h * a_i \equiv X^i - X^{n-1}$, This proves that the last n row vectors belong to Λ_h. □

This reverse HNF is different from the original HNF: the bottom half vectors are no longer circular rotations of each other; and the n-th row vector is not a q-vector. However, it does have the properties required to run a hybrid attack, which we specify below.

4.2 The Shape of Permuted HNF

In order for a permuted HNF $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ to be used to launch a hybrid attack, the following properties are required:

- The first r vectors must be q-vectors, that is q-multiples of canonical vectors (all coordinates equal to zero, except one equal to 1), in no particular order.

– The last k Gram-Schmidt vectors $(\boldsymbol{b}^\star_{m-k+1}, \ldots, \boldsymbol{b}^\star_m)$ must be canonical vectors, in no particular order.

For the initial HNF H, one may take $r = n$ and $k = n$, and for the reverse HNF, one may take $r = n-1$ and $k = n-1$, but such large values are not necessary: a hybrid attack only requires r and k not to be too small, such as a fraction of n. Experimentally, we observe that for most permutations σ, the permuted HNF H_σ satisfies the required properties for r and k nearly equal to n:

– If σ is such that $\sigma(\{1, \ldots, n\}) = \{1, \ldots, n\}$, then we will have $r = n$ and $k = n$ like in the initial HNF.
– If σ is such that $\sigma(\{1, \ldots, n\}) = \{n+1, \ldots, 2n\}$, hen we will have $r = n-1$ and $k = n-1$ like in the reverse HNF
– For general σ, the smallest possible values of $n-r$ and $n-k$ are very small in practice. As an illustration, Table 1 reports the experimental distribution of $n-r$ and $n-k$ (for maximal r and k) for about hundred 1018-rank lattices corresponding to the `ntruhps2048509` parameter set, with σ chosen uniformly at random. We do not know any precise theoretical justification for this phenomenon: it would be interesting to have a model for the distribution of $n-r$ and $n-k$, in the spirit of [21].

Values	0	1	2	3	4	5	6	7	8	9	10	11	12
$n-k$	25%	31%	23%	9%	4%	3%	1%	1%	1%	0%	0%	0%	1%
$n-r$	25%	24%	25%	11%	4%	7 %	2%	1%	0%	0%	0%	0%	0%

Table 1. Distribution of $n-r$ and $n-k$ for `ntruhps2048509` lattices of rank 1018.

However, we have a combinatorial explanation why $n-r$ should be small, which we omit. It is based on the fact that the first r vectors of H_σ are q-vectors if and only if the sublattice M generated by the q-vectors $(q\mathbb{Z}e_{\sigma(i)})_{1 \le i \le r}$ is a primitive sublattice of the NTRU lattice Λ_h.

5 Randomizing the Hybrid Attack with LSH and Permuted HNF

In this section, we present our new hybrid attacks. We will not recall the original hybrid attack [19], because previous presentations of the hybrid attack such as [19, 15, 30] are fairly technical, relying heavily on matrices. Instead, we will adopt a more algebraic point of view: we first describe the main underlying ideas, then present the new hybrid attack.

5.1 Enumerating Cosets

We are interested in finding a target vector $\boldsymbol{t}$ in some lattice L. To make the problem meaningful, the target $\boldsymbol{t}$ must have a special property, such as being short, or being very close to some given point in the subpace spanned by L. Accordingly, we assume that it is possible to check efficiently if a given vector is equal to the target $\boldsymbol{t}$. In the hybrid attack, such a check is easy: the target is any element of $\mathcal{S}_h$, *i.e.*any rotation of the secret vector derived from the secret polynomials f and g.

Many lattice algorithms, including the hybrid attack, but also the NewHope Unique-SVP attack [2] analyzed in [1], as well as Ducas' projected sieve [10], find the target $\boldsymbol{t}$ by first searching its coset as follows. Let M be a pure sublattice of a lattice L: M might be thought of as a random variable, such as the lattice generated by the first d vectors of some "random" reduced basis of L. Then the quotient group L/M is torsion-free, and can be viewed as a lattice, namely the projection of L over $\mathrm{span}(M)^{\perp}$. This orthogonal projection implements the canonical surjection $\pi : L \to L/M$ defined by $\pi(\boldsymbol{x}) = \boldsymbol{x} + M$.

One benefit of viewing L/M as a quotient lattice rather than a projected lattice is that if two lattice vectors $\boldsymbol{u}, \boldsymbol{v} \in L$ have the same projection $\pi(\boldsymbol{u}) = \pi(\boldsymbol{v})$, then we immediately see that $\boldsymbol{u} - \boldsymbol{v} \in M$, that is: $\boldsymbol{u} \equiv \boldsymbol{v}$ modulo M. We call *lifting* any function $\ell : L/M \to L$ which can invert π, that is, $\pi(\ell(\boldsymbol{y})) = \boldsymbol{y}$ for all $\boldsymbol{y} \in L/M$. Thus, a lifting selects a residue in a coset $\boldsymbol{x} + M$.

Identifying the Target Coset. Assume that we are able to confine the target coset $\pi(\boldsymbol{t})$: more precisely, we are given a subset $\mathcal{T} \subseteq L/M$ such that $\pi(\boldsymbol{t}) \in \mathcal{T}$. We provide a few examples below:

- In the NewHope attack [2, 1] on Unique-SVP, the target is the unique shortest vector. If the basis is sufficiently reduced, it is argued that $\boldsymbol{t}$ and one of the last basis vectors are likely to have the same projection by π. Thus, the subset $\mathcal{T}$ is small, formed by the projection of a few basis vectors.
- In Ducas' projected sieve [10], $\mathcal{T}$ consists of all the short vectors of L/M below a certain radius, and it is built by running a sieve algorithm on the quotient lattice L/M. Here, the number of elements of $\mathcal{T}$ is exponential in the rank of L/M.
- In [13, Sect. 5.2], Gama and Nguyen reported experimental results on NTRU lattices in which $\mathcal{T}$ was formed by the shortest vectors of L/M, found by applying strong reduction algorithms on L/M.
- In the final stage of the hybrid attack, $\mathcal{T}$ is a list of candidates for the target coset, much smaller than the initial list.

Imagine that we could enumerate the set $\mathcal{T}$: for each $\boldsymbol{u} \in \mathcal{T}$, we would like to be able to decide if $\boldsymbol{u} = \pi(\boldsymbol{t})$, and in that case, recover $\boldsymbol{t}$.

Lifting the Target Coset. Given $\boldsymbol{u} \in \mathcal{T}$, the simplest strategy is to select a lifting ℓ and check if $\ell(\boldsymbol{u}) = \boldsymbol{t}$. By enumerating $\mathcal{T}$, we will recover the target $\boldsymbol{t}$

if and only if $\ell(\pi(\boldsymbol{t})) = \boldsymbol{t}$. The success of this method depends on the choice of lifting ℓ, as well as the exact properties of the target $\boldsymbol{t}$.

The simplest lifting is the following lifting introduced historically by Hermite [14], which corresponds to LLL's so-called size-reduction and Babai's nearest plane algorithm [5]:

Definition 1. *Let $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be a basis of a lattice M. For any lattice L such that M is a pure sublattice of L, we denote by ℓ_B^L the map: $L/M \to L$ defined as: for any $\boldsymbol{x} \in L/M$, $\ell_B^L(\boldsymbol{x} + M)$ is the unique point $\boldsymbol{y} \in L$ such that $\pi(\boldsymbol{y}) = \boldsymbol{x}$ and $\tau(\boldsymbol{y}) \in \mathcal{P}(B^\star)$, where $\pi : L \to L/M$ is the canonical surjection, τ is the orthogonal projection over* $\mathrm{span}(M)$ *and*

$$\mathcal{P}(B^\star) = \{\sum_{i=1}^{d} x_i \boldsymbol{b}_i^\star, -1/2 \leq x_i < 1/2\}.$$

Thus, this map lifts cosets using Babai's nearest plane algorithm: for any preimage $\boldsymbol{z} \in L$ of $\boldsymbol{x}$ by π, we apply Babai's nearest plane algorithm to $(-\boldsymbol{z}, B)$ to obtain $\boldsymbol{z}' \in M$ such that $\tau(\boldsymbol{z}' + \boldsymbol{z}) \in \mathcal{P}(B^\star)$, and we let $\boldsymbol{y} = \boldsymbol{z}' + \boldsymbol{z}$. This lifting can be computed in polynomial time, but there are of course more expensive liftings. In fact, any deterministic CVP approximation algorithm defines a lifting. An extreme case would be the Voronoi lifting in which one replaces Babai's nearest plane algorithm with a CVP algorithm: this replaces $\mathcal{P}(B^\star)$ in the definition by the Voronoi cell of M. This lifting is expensive, but it is still cheaper than solving CVP on the full lattice L, because M has lower rank.

Lifting Success Probability. Since many algorithms [10, 2, 1, 19] are based on Hermite's lifting ℓ_B^L, it is important yet non-trivial to analyze its success probability: we'd like to assess how likely is $\ell_B^L(\pi(\boldsymbol{t})) = \boldsymbol{t}$, which is equivalent to $\tau(\boldsymbol{t}) \in \mathcal{P}(B^\star)$.

For the final stage of the hybrid attack, this actually won't be a problem at all, because there, M is an orthogonal sublattice, so B is an orthogonal basis, and if $\boldsymbol{t}$ is a shortest lattice vector, we will necessarily have $\ell_B^L(\pi(\boldsymbol{t})) = \boldsymbol{t}$. However, for completeness, we discuss the general case, and it will be useful for other parts of the algorithm.

In the projected sieve [10], this issue is loosely analyzed: it uses the well-known sufficient condition for $\tau(\boldsymbol{t}) \in \mathcal{P}(B^\star)$ that $\|\tau(\boldsymbol{t})\| \leq \min_{1 \leq i \leq d} \|\boldsymbol{b}_i^\star\|/2$, and it argues that the latter condition is very likely to be satisfied for the setting of [10]. Since the sufficient condition is not believed to be tight, this is only a rough analysis.

The articles [19, 15, 1, 31, 6] all require at some point to estimate the probability that $\tau(\boldsymbol{t}) \in \mathcal{P}(B^\star)$. They all make the usual assumption that $\mathrm{span}(M)$ is a random subspace, so that $\tau(\boldsymbol{t})$ is a random projection of $\boldsymbol{t}$. However, this is not sufficient to estimate the probability. To make it possible, they further assume that the coordinates of $\tau(\boldsymbol{t})$ along the axes of the parallelepiped $\mathcal{P}(B^\star)$ are independent and have the same distribution: this distribution is assumed to be Gaussian in [15, Assumption 5], whereas [1, Sect. 4.3] and [6, Discussion of

Heuristic 4] (reused in [31]) use other distributions, such as the "square root" of some Beta distribution. This is a very strong assumption, which does not hold for a random unit vector. In particular, the independence assumption implies that the probability can be expressed as a product of elementary probabilities: these elementary probabilities can be derived from the CDF of the underlying distribution, related to the error function erf for [19, 15] or the Beta function for [1, 31, 6].

Unfortunately, we argue that such models are oversimplifying and may lead to questionable estimates. To see this, consider the following concrete example: imagine to simplify that $\mathcal{P}(B^\star)$ is the unit-volume box $H = [-1/2, 1/2]^n$ and that $\tau(\boldsymbol{t})$ is actually a point chosen uniformly at random in the n-dimensional ball $B_n = \mathrm{Ball}_n(\sqrt{n/12})$ of radius $\sqrt{n/12}$. The ball B_n contains the unit-volume ball whose radius is equivalent to $\sqrt{\frac{n}{2\pi e}}$. B has asymptotical volume $\frac{1}{\sqrt{n\pi}}\sqrt{\frac{\pi e}{6}}^n$ where $\sqrt{\frac{\pi e}{6}} \approx 1.193$. [3, Th. 3] proves that $\mathrm{vol}(H_n \cap B_n)$ converges to $1/2$ as n grows to infinity, which implies that the probability that a random point in H_n belongs to B_n converges to $1/2$, and that the probability that a random point in B_n belongs to H_n is equivalent to $\frac{\sqrt{n\pi}}{2}\sqrt{\frac{6}{\pi e}}^n$. The latter probability is also the probability that a random point in the unit ball belongs to $[-\sqrt{\frac{3}{n}}, \sqrt{\frac{3}{n}}]^n$ Yet, heuristic estimates like [19, 15, 31, 1] lead instead to a completely different approximation. For instance, the methodology of [19, 15] would heuristically model a random point of $\mathrm{Ball}_n(\sqrt{n/12})$ as a point whose coordinates are independent and normally distributed with expectation $\mu = 0$ and standard deviation $\sigma = \sqrt{1/12}$. This would estimate $\mathrm{vol}(H \cap \mathrm{Ball}_n(\sqrt{n/12}))$ as:

$$\mathrm{vol}\left(\mathrm{Ball}_n\left(\sqrt{\frac{n}{12}}\right)\right)\left(\mathrm{erf}\left(\frac{1}{2\sigma\sqrt{2}}\right)\right)^n \sim \frac{(e\pi/6)^n}{\sqrt{\pi n}}\left(\mathrm{erf}\left(\frac{1}{2\sigma\sqrt{2}}\right)\right)^n = \frac{\alpha^n}{\sqrt{\pi n}},$$

where $\alpha \approx 1.094 > 1$. So we would think that the intersection volume grows exponentially in n, though it actually converges to $1/2$.

Instead, we suggest the following method: first, identify or model the distribution of $\|\tau(\boldsymbol{t})\|$, then, for the mean and or the most likely values R of $\|\tau(\boldsymbol{t})\|$, estimate the probability that a vector chosen uniformly at random from the sphere of radius R belongs to $\mathcal{P}(B^\star)$. Because the sphere is invariant by rotation, the probability remains the same if $\mathcal{P}(B^\star)$ is replaced by the box $\prod_{i=1}^n [-\|\boldsymbol{b}_i^\star\|/2, \|\boldsymbol{b}_i^\star\|/2]$. We will address the technical problem of estimating this probability in Sect. 6.

We now discuss the distribution of $\|\tau(\boldsymbol{t})\|$. Let $\boldsymbol{u} = (u_1, \ldots, u_n)$ be a point chosen uniformly at random from the unit sphere S^{n-1}: we do so by letting $u_i = x_i/\sqrt{\sum_{j=1}^n x_j^2}$, where $x_1, \ldots, x_n$ are independent, normally distributed random variables. Denote the k-th truncation of $\boldsymbol{u}$ by $\tau_k(\boldsymbol{u}) = (u_1, \ldots, u_k)$ for $1 \le k \le n$. Note that

$$\|\tau_k(\boldsymbol{u})\|^2 = \frac{\sum_{i=1}^k x_i^2}{\sum_{i=1}^n x_i^2} = \frac{\sum_{i=1}^k x_i^2}{\sum_{i=1}^k x_i^2 + \sum_{i=k+1}^n x_i^2} = \frac{X}{X+Y},$$

where X has distribution Gamma$(k/2, 2)$ and Y has distribution Gamma$((n-k)/2, 2)$. Hence, $\|\tau_k(\boldsymbol{u})\|^2$ has distribution Beta$(k/2, (n-k)/2)$.

We deduce that if $\boldsymbol{t} \in \mathbb{R}^n$ has norm R and span(M) is a random subspace of dimension d, then $\|\tau(\boldsymbol{t})\|^2$ follows some Beta distribution of expectation $R^2 d/n$. We note that the general problem of computing $\Pr(X \in H)$ where $\|X\|^2$ follows some Beta distribution, and H is a n-dimensional box has not been studied in the literature. To the best of our knowledge, only the special case where the Beta distribution corresponds to the uniform distribution over a ball has been studied in [3].

However, we stress that the analysis must be adapted to each situation. For instance, in the case of the hybrid attack, we will see that $\boldsymbol{t}$ is a short integer vector such as a ternary or binary vector, and span(M) will be generated by some randomly chosen canonical vectors. Then each coordinate of τt is a a small integer, such as $0, \pm 1$. Then $\|\tau(\boldsymbol{t})\|^2$ only takes finitely many values, all integral. So a rigorous estimate would compute the probability distribution of the discrete random variable $\|\tau(\boldsymbol{t})\|^2$, and for any integer m, would also compute $\Pr(X \in H)$ where X is chosen uniformly at random from the sphere of radius $\sqrt{m}$.

5.2 Close Points in a Torus

We started by assuming that we could directly enumerate the target coset $\pi(\boldsymbol{t})$: this is done in the final stage of the hybrid attack, but not in the first stage. Instead, the first stage speeds up such an enumeration somewhat like in a sieve algorithm. Namely, imagine that the list $\mathcal{T}$ can be expressed as a product: assume that we know two subsets $U, V \subseteq L/M$ such that there exists $(\boldsymbol{u}, \boldsymbol{v}) \in U \times V$ such that $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$. This equality can be characterized as follows:

Lemma 4. *Let M be a pure sublattice of a lattice L, with $\pi : L \to L/M$ the canonical surjection, and $\ell : L/M \to L$ any lifting inverting π. Let $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}) \in L \times (L/M)^2$. Then $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$ if and only if:*

$$\boldsymbol{t} + \ell(\boldsymbol{v}) - \ell(\boldsymbol{u}) \in M.$$

Proof. If $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$, then $\pi(\boldsymbol{t} + \ell(\boldsymbol{v})) = \pi(\ell(\boldsymbol{u}))$ and therefore: $\boldsymbol{t} + \ell(\boldsymbol{v}) - \ell(\boldsymbol{u}) \in M$. Reciprocally, if $\boldsymbol{t} + \ell(\boldsymbol{v}) - \ell(\boldsymbol{u}) \in M$ then $\pi(\boldsymbol{t} + \ell(\boldsymbol{v}) - \ell(\boldsymbol{u})) = 0$ and the result follows by linearity of π and definition of the lifting. □

Thus, if $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$, then we have in the torus span$(M)/M$:

$$\tau(\boldsymbol{t}) + \tau(\ell(\boldsymbol{v})) \equiv \tau(\ell(\boldsymbol{u})). \tag{1}$$

If the target $\boldsymbol{t}$ is a short vector, then its projection $\tau(\boldsymbol{t})$ is also short, which means that $\tau(\ell(\boldsymbol{v}))$ and $\tau(\ell(\boldsymbol{u}))$ are close to each other in the torus span$(M)/M$.

Hence, the problem has been transformed into identifying close pairs: can we find all $(\boldsymbol{u}, \boldsymbol{v}) \in U \times V$ such that $\tau(\ell(\boldsymbol{u}))$ and $\tau(\ell(\boldsymbol{v}))$ are close modulo M? We just need to narrow down the initial list $U \times V$ enough to speed up the coset enumeration. We now discuss two ways to tackle this problem efficiently, depending on the shape of the torus, namely whether the sublattice M is orthogonal or not. The hybrid attack will combine both.

5.3 Orthogonal Sublattices: Torus LSH

We assume here that the pure sublattice M is orthogonal, and that we know an orthogonal basis $(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$ of M. In Sect. 3, we introduced a family of $\mathcal{H}(q, n)$ of hash functions over the discrete $\mathbb{Z}_q^n$: this construction can easily be adapted to the torus $\mathrm{span}(M)/M$. In fact, if the $\boldsymbol{q}_i$'s were all q-vectors, we could simply reuse it after a suitable change of coordinates. In the general case, we simply replace the discrete torus by the continuous torus as follows. Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. For any $u \in \mathbb{T}$, let $\sigma'_u : \mathbb{T} \to \{0, 1\}$ defined by $\sigma'_u(x) = 0$ if and only if:

$$(x - u) \bmod 1 \leq 1/2.$$

We have the continuous variant of Lemma 1:

Lemma 5. *Let $a, b \in \mathbb{T}$. Let c be the smallest residue of $b - a$ in absolute value: $c = \min_{k \in \mathbb{Z}} |b - a - k| \leq 1/2$. Let u be chosen uniformly at random from $\mathbb{T}$. Then:*

$$\Pr_u(\sigma'_u(a) \neq \sigma'_u(b)) = 2c.$$

Similarly, we deduce a family $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$ of hash functions over $\mathrm{span}(M)/M$ by: To sample from $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$, we choose $u_1, \ldots, u_r \in \mathbb{T}$ independently and uniformly at random. Then we define $H : \mathrm{span}(M)/M \to \{0, 1\}^n$ by $H(\sum_{i=1}^r x_i \boldsymbol{q}_i) = (\sigma_{u_1}(x_1), \ldots, \sigma_{u_r}(x_r))$ where $x_1, \ldots, x_r \in \mathbb{T}$. We obtain the following analogue of Th. 1:

Theorem 3. *Let M be a lattice having an orthogonal basis $(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$. Let $\boldsymbol{a}$ and $\boldsymbol{b} \in \mathrm{span}(M)/M$: $\boldsymbol{a} = \sum_{i=1}^r a_i \boldsymbol{q}_i$ and $\boldsymbol{b} = \sum_{i=1}^r b_i \boldsymbol{q}_i$ where $a_i, b_i \in \mathbb{R}/\mathbb{Z}$. Let $e_i = \min_{k \in \mathbb{Z}} |b_i - a_i - k|$ for $1 \leq i \leq r$. If H is chosen uniformly at random from $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$, then:*

$$\Pr_H(H(\boldsymbol{a}) = H(\boldsymbol{b})) = \prod_{i=1}^r (1 - 2e_i).$$

This suggests to solve (1) as follows: if $\tau(\boldsymbol{t})$ is short, then the probability (over H) that $H(\tau(\ell(\boldsymbol{u}))) = H(\tau(\ell(\boldsymbol{v})))$ will be high, so we select a random hash function H from $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$ and sort all $H(\tau(\ell(\boldsymbol{v})))$ for $\boldsymbol{v} \in V$. Then one can detect all collisions with $H(\tau(\ell(\boldsymbol{u})))$ for $\boldsymbol{u} \in U$.

To conclude this section, we give elementary results on the distribution of the hash function: we will not need these results, they are just given for intuition.

If $\boldsymbol{a}$ is fixed and $\boldsymbol{b}$ is chosen uniformly at random from $\boldsymbol{b} \in \mathrm{span}(M)/M$, then the e_i's are independent and have uniform distribution over $[0, 1/2]$. Then each $X_i = \ln(1 - 2e_i)$ satisfies the requirements of the central limit theorem:

Lemma 6. *Let e be a random variable with uniform distribution over $[0, 1/2)$. Then the random variable $X = \ln(1 - 2e)$ satisfies:*

$$E(X) = -1, V(X) = 2, \sigma(X) = 1.$$

Proof. X has range $(-\infty, 0]$ and pdf e^x. □

This allows us to prove the following:

Theorem 4. *Let M be a lattice having an orthogonal basis $(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$. Let $\boldsymbol{a} \in \mathrm{span}(M)/M$. Let $\boldsymbol{b} \in \mathrm{span}(M)/M$ be chosen uniformly at random. Denote by $\delta(\boldsymbol{a}, \boldsymbol{b}) = \Pr_H(H(\boldsymbol{a}) = H(\boldsymbol{b}))$ where H is chosen uniformly at random from $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$. Then for any $x \in \mathbb{R}$*

$$\lim_{r\to\infty} \Pr_{\boldsymbol{b}}(\delta(\boldsymbol{a}, \boldsymbol{b}) \leq e^{x\sqrt{r}-r}) = \frac{1 + \mathrm{erf}((x+1)/\sqrt{2})}{2}.$$

Lemma 7. *Let L be a lattice having an orthogonal basis $(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$. Let H be in $\mathcal{H}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r)$. Let $\boldsymbol{y} \in \{0,1\}^n$. The set of preimages $\boldsymbol{x} \in \mathbb{T}^n$ such that $\boldsymbol{y} = H(\boldsymbol{x})$ has measure $\mathrm{vol}(L)/2^n$.*

5.4 Arbitrary Sublattices: the Admissibility Trick

It does not seem easy to build an efficient LSH over an arbitrary torus if M is not orthogonal. However, if the vector $\boldsymbol{t} + \ell(\boldsymbol{v}) - \ell(\boldsymbol{u})$ of M is actually zero and $\boldsymbol{t}$ is short, it will imply that $\ell(\boldsymbol{v})$ and $\ell(\boldsymbol{u})$ are close to each other, which suggests the following definition:

Definition 2. *We say that $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}) \in L \times (L/M)^2$ is* ℓ-admissible *if and only if $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$ and the following equation holds in L:*

$$\boldsymbol{t} + \ell(\boldsymbol{v}) = \ell(\boldsymbol{u}).$$

This is a nice case: if $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ is *ℓ-admissible*, then we can retrieve the target $\boldsymbol{t}$ as $\boldsymbol{t} = \ell(\boldsymbol{u}) - \ell(\boldsymbol{u})$.

If $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$, the second condition is in general unlikely to happen, because L/M is infinite (as while as $\mathrm{rank}(M) < \mathrm{rank}(L)$). But if $\boldsymbol{t}$ is short and if we choose Hermite's lifting for ℓ, it turns out to be possible. For this lifting, we have the following criterion for admissibility:

Lemma 8. *Let B be a basis of a pure sublattice M of L, and τ be the orthogonal projection over $\mathrm{span}(M)$. Assume that $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}) \in L \times (L/M)^2$ is such that $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$. Then $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ is ℓ_B^L-admissible if and only if*

$$\tau(\boldsymbol{t} + \ell_B^L(\boldsymbol{v})) \in \mathcal{P}(B^\star).$$

Proof. Let $\ell = \ell_B^L$. If $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ is ℓ-admissible, then: $\boldsymbol{t} + \ell(\boldsymbol{v}) = \ell(\boldsymbol{u})$. Thus, $\tau(\boldsymbol{t} + \ell(\boldsymbol{v}) = \tau(\ell(\boldsymbol{u})) \in \mathcal{P}(B^\star)$ by definition of ℓ_B^L.

Reciprocally, if $\tau(\boldsymbol{t} + \ell(\boldsymbol{v}) \in \mathcal{P}(B^\star)$, the fact that $\pi(\boldsymbol{t} + \ell(\boldsymbol{v})) = \ell(\boldsymbol{u})$ implies that $\ell(\boldsymbol{u}) = \boldsymbol{t} + \ell(\boldsymbol{v})$ by definition of ℓ_B^L (unicity). □

The condition $\tau(\boldsymbol{t} + \ell_B^L(\boldsymbol{v})) \in \mathcal{P}(B^\star)$ of Lemma 8 can be rewritten as $\tau(\boldsymbol{t}) + \tau(\ell_B^L(\boldsymbol{v})) \in \mathcal{P}(B^\star)$. Notice that by definition of ℓ_B^L, we know that $\tau(\ell_B^L(\boldsymbol{v})) \in \mathcal{P}(B^\star)$. This suggests the following model:

Definition 3. *The* admissibility model *states that with respect to admissibility,* $\tau(\boldsymbol{t})$ *and* $\tau(\ell_B^L(\boldsymbol{v}))$ *behave like independent vectors chosen chosen uniformly at random from respectively the sphere of radius* R *and* $\mathcal{P}(B^\star)$*.*

As usual in cryptanalysis, we stress that this is only an idealized model which we do not expect to hold formally (since $\boldsymbol{t}$ and $\boldsymbol{v}$ are related to each other): experiments with the attack will check if the model is realistic. The second assumption that a random-looking vector in $\mathcal{P}(B^\star)$ has uniform distribution is usual in lattice cryptanalysis: it appeared in attacks [23] against lattice-based signatures. We already discussed the first assumption in the overview (Sect. 5.1): if $\tau(\boldsymbol{t})$ has varying norm, we may use the distribution of $\|\tau(\boldsymbol{t})\|$ to identify the most likely norms.

In the admissibility model, the probability that $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ is ℓ_B^L-admissible can be identified with the probability that $\boldsymbol{x} + \boldsymbol{y} \in \mathcal{P}(B^\star)$ when $\boldsymbol{x}$ and $\boldsymbol{y}$ are chosen uniformly at random from respectively the sphere of radius R and $\mathcal{P}(B^\star)$. This is close to [6, Heuristic 4]. Because the sphere is invariant by rotation, the latter probability is also the probability that a random vector in the sphere of radius R belongs to $H - \boldsymbol{e}$ where $H = \prod_{i=1}^{n}[-\|\boldsymbol{b}_i^\star\|/2, \|\boldsymbol{b}_i^\star\|/2]$ and $\boldsymbol{e}$ is chosen uniformly at random from the box H. By scaling, we are thus interested in the distribution of the probability that a random unit vector belongs to a random box H': previous analyses [30, 19] implicitly attempted to heuristically estimate the mean of this distribution, but we observe that this distribution has intuitively (and experimentally) high variance, and it is therefore more useful to study the distribution. Hence, the admissibility model allows us to reduce the problem of approximating the probability of admissibility to approximating the probability that a random unit vector belongs to a given box: we will address this question in details in Sect. 6.

5.5 The Full Hybrid Attack

We now have all the ingredients to describe the hybrid attack, which is summarized by Alg. 2. Let L be the NTRU lattice we wish to attack. We choose a random permutation σ of $\{1, \ldots, 2n\}$ tailored to the distribution of the secret key (f, g): we require that $\sigma(\{1, \ldots, n\}) = \{1, \ldots, n\}$ if f is sparser than g, or $\sigma(\{1, \ldots, n\}) = \{n+1, \ldots, 2n\}$ if g is sparser than f. We compute the permuted HNF $H_\sigma = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2n})$ of L, which we can assume to have the following special shape (see Sect. 4):

- The first r vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ are q-vectors: they generate an orthogonal pure sublattice $\Lambda = L(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r)$ isomorphic to $q\mathbb{Z}^r$.
- The last k Gram-Schmidt vectors $\boldsymbol{c}_{2n-k+1}^\star, \ldots, \boldsymbol{c}_{2n}^\star$ are canonical vectors: If we define the pure sublattice $M = L(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m)$ where $m = 2n - k$, this means that the lattice L/M is isomorphic to the integer lattice $\mathbb{Z}^k$.

The third isomorphism theorem is essential to the hybrid attack:

$$L/M \simeq \frac{L/\Lambda}{M/\Lambda} \tag{2}$$

Next, we prepare the meet-in-the-middle stage:

- We define two subsets U and V of L/M such that there exists $(\boldsymbol{u}, \boldsymbol{v}) \in U \times V$ and a rotation $\boldsymbol{t}$ of the secret key such that $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$, where π is the canonical surjection $L \to L/M$.

- Compute a reduced basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ of the lattice M/Λ, aiming at maximizing $\min_{1 \leq i \leq d} \|\boldsymbol{b}_i^\star\|$. It is well-known that this amounts to reduce the dual lattice of M/Λ.

By the isomorphism (2), we can view $\pi(\boldsymbol{t}), \boldsymbol{u}, \boldsymbol{v}$ as elements of $(L/\Lambda)/(M/\Lambda)$. The basis B defines a lifting $\ell_B^{L/\Lambda}$ from L/M to L/Λ. If the triplet $(\boldsymbol{t}+\Lambda, \boldsymbol{u}, \boldsymbol{v}) \in L/\Lambda \times (L/M)^2$ is $\ell_B^{L/\Lambda}$-admissible, then by definition the following equality holds in L/Λ:

$$\boldsymbol{t} + \Lambda = \ell_B^{L/\Lambda}(\boldsymbol{u}) - \ell_B^{L/\Lambda}(\boldsymbol{v}) \tag{3}$$

But (3) means that we can express the target coset $\boldsymbol{t}+\Lambda$ as a difference of candidate cosets in respectively $\ell_B^{L/\Lambda}(U)$ and $\ell_B^{L/\Lambda}(V)$. Because we know a basis of Λ consisting only of q-vectors and because the projection of $\boldsymbol{t}$ over Λ is very short, we can solve (3) by Torus LSH, as in Odlyzko's attack (Sect. 3), as explained in Sect. 5.3. More precisely, let ℓ be the lifting from L/Λ to L defined by the q-vectors corresponding to Λ, and τ be the orthogonal projection over $\mathrm{span}(\Lambda)$: τ simply takes the coordinates corresponding to the q-vectors defining Λ. We let $\psi : L/\Lambda \to \mathbb{Z}_q^r$ be the map defined by $\psi(w) = (x_1, \ldots, x_r) \bmod q$ where the integers x_i's are defined by the decomposition of $\tau(\ell(w)) = \sum_{i=1}^r x_i \boldsymbol{e}_{j_i}$ over the canonical vectors $\boldsymbol{e}_{j_i}$'s defining $\mathrm{span}(\Lambda)$.

Algorithm 2 The Randomized Hybrid Attack

Input: An NTRU public key (h, n, q), two positive integers r and k.
Output: A secret vector in $\mathcal{S}_h$.

1: Select a random permutation σ of $\{1, \ldots, 2n\}$ optimized for the distribution of the secret key (f, g).
2: Compute the permuted HNF $H_\sigma = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2n})$, which defines the sublattices Λ and M spanned by respectively its first r and $2n - k$ vectors.
3: Compute a random reduced basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ of the lattice M/Λ, aiming at maximizing $\min_{1 \le i \le d} \|\boldsymbol{b}_i^\star\|$ to increase the admissibility probability.
4: Define (possibly randomly) two subsets U and V of L/M such that there exists $(\boldsymbol{u}, \boldsymbol{v}) \in U \times V$ and a target $\boldsymbol{t} \in \mathcal{S}_h$ such that $\pi(\boldsymbol{t}) = \boldsymbol{u} - \boldsymbol{v}$, where π is the canonical surjection $L \to L/M$.
5: Define the map $\psi : L/\Lambda \to \mathbb{Z}_q^n$ as in Sect. 5.5.
6: Select a random hash function H from the LSH family $\mathcal{H}(q, r)$ defined in Sect 3.
7: Compute and sort the list $\mathcal{L} = \{(\boldsymbol{u}, H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{u}))), \boldsymbol{u} \in U\}$ so that collisions over $H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{u})))$ can be detected in logarithmic time.
8: **for** $\boldsymbol{v} \in V$ **do**
9: Compute $H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{v})))$.
10: **for** each collision $(\boldsymbol{u}, H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{u})))) \in \mathcal{L}$ such that $H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{u}))) = H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{v})))$ **do**
11: Lift $\ell_B^{L/\Lambda}(\boldsymbol{u}) - \ell_B^{L/\Lambda}(\boldsymbol{v}) \in L/\Lambda$ as $\boldsymbol{w} \in L$ using the orthogonal basis of Λ.
12: Return $\boldsymbol{w}$ if $\boldsymbol{w} \in S_h$.
13: **end for**
14: **end for**

The correctness of the hybrid attack relies entirely on the admissibility condition:

Theorem 5. *Let (h, f, g, q, n) be an NTRU key defining a lattice L and a set of short target lattice vectors $\mathcal{S}_h = \{(x^i * g, x^i * f), 0 \le i \le n - 1\} \subseteq L$. If U, V, Λ, B selected by Alg. 2 on input (h, q, n) are such that there exists an $\ell_B^{L/\Lambda}$-admissible triplet $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}) \in \mathcal{S}_h \times U \times V$, then, over the choice of H in Line 6, the probability that Alg. 2 returns $\boldsymbol{t}$ is $\ge (1 - 2/q)^\omega$ where ω is the number of non-zero coordinates of the orthogonal projection of $\boldsymbol{t}$ over* $\mathrm{span}(\Lambda)$.

Proof. If $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ is $\ell_B^{L/\Lambda}$-admissible, then the proof is identical to Th. 2. The statement follows from Th. 1. □

Asymptotically, the NTRU parameters satisfy $q = \Theta(n)$ and $\omega \le r$ so $(1 - 2/q)^\omega \ge \Omega(1)$. Thus, the probability in Th. 5 is lower-bounded by a constant: In the attack parameters of NTRUHPS [7], we have $q \ge 2048$ and $r \le 150$ so the success probability (assuming admissibility) of one trial is $\ge 85\%$.

If there is no admissible triplet, the attack won't work, but we can rerun it with a different choice of B, Λ and even U and V. Independently of the success probability, it remains to analyze the running time of one run in Alg. 2. This will require heuristics, but significantly less problematic than in the original hybrid attack, similarly to Odlyzko's attack. The crucial lines are:

- Line 3: lattice reduction. Several models have been proposed in the past few years to approximately estimate the running time required by lattice reduction for a given output quality: they are used by all the lattice-based NIST finalists.
- Line 4: how to define the meet-in-the-middle sets U and V, depending on the distribution of f and g. In [7], it is assumed that if η is the entropy of $\pi(\boldsymbol{t})$, then we can build U and V of size approximately $2^{\eta/2}$. We will discuss the construction of U and V in Sect. 5.6.
- Line 10: like in Alg. 1, we model the construction of the list $\mathcal{L}$ built by Alg. 2 by the balls into bins problem, and we discuss the maximal load N, *i.e.* the maximal number of elements in a bin. If $\#U \leq 2^r$ (which typically holds for NTRU parameters) and all the hashes $H(\psi(\ell_B^{L/\Lambda}(\boldsymbol{u})))$'s were independent and uniformly distributed, then it is well-known that with high probability, we would have $N = O(\frac{r}{\log r})$. In order for Alg. 2 to be efficient, it is sufficient to assume that N is polynomial in r for a random H, which is much weaker than $N = O(\frac{r}{\log r})$. Specifically, we can make the following heuristic assumption: Let (f, g, h) be an NTRU key. If a torus-LSH H is chosen uniformly at random, then with overwhelming probability, the maximal number of elements in a bin is polynomial in r.

5.6 Optimization

The hybrid attack relies on many parameters, which makes finding the best hybrid attack tricky. One key issue is to balance the cost of lattice reduction (Line 3) with that of the meet-in-the-middle stage (Line 4).

We now discuss the generation of U and V. The coordinates of $\pi(\boldsymbol{t})$ are usually of two types:

Uniform Coordinates. Here, they are uniformly random in $\{0, 1\}$ or ± 1. In that case, one can set the cardinal of both U and V to exactly the square root of the total number of possibilities, by splitting the coordinates in two halves: one half defining U, and the other half defining V.

Sparse coordinates. Here, they can be regrouped in such a way that the coordinates are chosen uniformly at random with a prescribed number of 1 and -1. (with either only 0 and 1, or with nearly the same number of 1 and -1). In that setting, it is also possible to select U and V of size nearly the square root of the total number of possibilities.. To do this, we apply Coppersmith's trick for meet-in-the-middle algorithms [28] to solve the low-hamming weight discrete

logarithm, by choosing U and V randomly. Assume that the prescribed number of non-zero coordinates is $2c$ to be chosen among $2b$ coordinates (we select even parameters for simplicity). Then we select at random b coordinates among the $2b$ coordinates, and define U by considering all choices of c non-zero coordinates among those b. The set V is obtained by doing the same thing over the remaining b coordinates. The probability that a fixed combination is covered by our choice of (U, V) is $\binom{2c}{c}\binom{2b-2c}{b-c}/\binom{2b}{b}$. Once the non-zero coordinates have been guessed, we assign all possible signs, and it can be checked that the number of elements of U and V is close to the square root of exhaustive search.

There is a further optimisation for the case of sparse coordinates, which our permuted HNF can boost. In the previous paragraph, we assumed we knew the number of non-zero coordinates $2c$, but that freedom over c can be exploited: the smaller the c, the faster the exhaustive search. Consider a target vector (g, f). In the initial hybrid attack [19], one was targetting the last k coordinates of f, because f was sparser than g, and because the attack only used the HNF, and not other bases. Howgrave-Graham [19]then computed experimentally for all possible choices of k and $2c$ the probability that there exists at least one target $\boldsymbol{t} \in \mathcal{S}_h$ such that the last k coordinates of $\boldsymbol{t}$ contained exactly $2c$ non-zero coordinates. However, when we take the last k coordinates, they are consecutive: we observe that the number of non-zero coordinates in the last k coordinates of the rotations are correlated. Intuitively and experimentally, to minimize the number of non-zero coordinates, it is better to select k coordinates at random, rather than k consecutive coordinates, to decrease the effects of correlations between rotations. If f is sparser than g, we would choose these k coordinates among the n coordinates of f, and if g is sparser than f (as in NTRU's NIST submission), we would pick them among the n coordinates of g.

6 Boxed Spheres

To analyze lattice enumeration with discrete pruning, Aono and Nguyen [3] studied the following mathematical problem: approximate $\mathrm{vol}(B \cap H)$, where B is the ball of center $\boldsymbol{c} \in \mathbb{R}^n$ and radius R, and H is an arbitrary box, *i.e.* a product of intervals

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \text{ s.t. } \alpha_i \leq x_i \leq \beta_i\},$$

where the α_i's and β_i's are given as input. They proposed several methods to approximate the volume, but none was guaranteed to be efficient and provable in high dimension. More precisely, they gave two Fourier-based series expansions for $\mathrm{vol}(B \cap H)$ by sligthly generalizing the works of respectively Constales and Tibken [25], and a heuristic practical method based on the Laplace transform which works well in practice until at least dimension 150: however, it is unclear how well would the method behave in much higher dimension, while some of the NTRU parameters require a dimension over 800.

In this section, motivated by our analysis of the hybrid attack, we study a slightly different problem, where the ball is replaced by a sphere: we are again

given an arbitrary box $H = \prod_{i=1}^{n}[\alpha_i, \beta_i]$, but this time, we want to approximate the probability that a random point in the sphere S of center $\boldsymbol{c} \in \mathbb{R}^n$ and radius R belongs to H. There is a connection with the previous problem, as $\Pr_{\boldsymbol{x}\in S}(\boldsymbol{x} \in H) \leq \Pr_{\boldsymbol{x}\in B}(\boldsymbol{x} \in H) = \mathrm{vol}(B \cap H)/\mathrm{vol}(B)$ where B is the ball with the same center and radius as S. As in [3], without loss of generality, we may assume that $\boldsymbol{c} = 0$ after suitable translation, and $R = 1$ after suitable scaling, which means that S is the unit sphere S_{n-1} in $\mathbb{R}^n$. In our setting, our boxes H all include zero, that is $\alpha_i \leq 0 \leq \beta_i$ for all i, so we will restrict to this setting to simplify notation, but we stress that our methods can be generalized with suitable modifications.

6.1 Elementary Bounds

Let $H = \prod_{i=1}^{n}[\alpha_i, \beta_i]$. We want to bound the probability $p(n, H) = \Pr_{\boldsymbol{x}\in S_{n-1}}(\boldsymbol{x} \in H)$. It is well-known that the uniform distribution over S_{n-1} can be obtained by picking a vector $\boldsymbol{y} \in \mathbb{R}^n$ with independent coordinates from the normal distribution $\mathcal{N}$, and returning the unit vector $\boldsymbol{x} = \boldsymbol{y}/\|\boldsymbol{y}\|$. Here, $\|\boldsymbol{y}\|^2$ follows a χ^2 distribution of parameter n. This gives rise to the following elementary bounds:

Lemma 9. *Let $H = \prod_{i=1}^{n}[\alpha_i, \beta_i]$ such that $\alpha_i\beta_i \leq 0$ for all i. Then for any $m \geq 0$:*

$$p(n, H) \geq \left(\prod_{i=1}^{n} \Pr_{y\leftarrow\mathcal{N}}(\alpha_i m \leq y \leq \beta_i m)\right) - \Pr_{\boldsymbol{y}\leftarrow\mathcal{N}^n}(\|\boldsymbol{y}\| < m),$$

and

$$p(n, H) \leq \left(\prod_{i=1}^{n} \Pr_{y\leftarrow\mathcal{N}}(\alpha_i m \leq y \leq \beta_i m)\right) + \Pr_{\boldsymbol{y}\leftarrow\mathcal{N}^n}(\|\boldsymbol{y}\| > m)$$

Proof. We pick at random $\boldsymbol{y} = (y_1, \ldots, y_n) \leftarrow \mathcal{N}^n$.

If $\|\boldsymbol{y}\| \geq m$ and $\alpha_i m \leq y_i \leq \beta_i m$ for all i, then $\alpha_i \leq \frac{y_i}{\|\boldsymbol{y}\|} \leq \beta_i$ because $\alpha_i\beta_i \leq 0$, which implies that $\boldsymbol{y}/\|\boldsymbol{y}\| \in H$. The lower bound on $p(n, H)$ follows from the independence of the n events $\alpha_i m \leq y_i \leq \beta_i m$.

Assume that $\boldsymbol{y}/\|\boldsymbol{y}\| \in H$ with $\|\boldsymbol{y}\| \leq m$. Then $\alpha_i \leq y_i/\|\boldsymbol{y}\| \leq \beta_i$ and $\alpha_i\beta_i \leq 0$ implies that $\alpha_i m \leq y_i \leq \beta_i m$. This gives the upper bound on $p(n, H)$. □

Despite the simplicity of Lemma 9, we can already derive non-trivial results:

Corollary 1. *Let $\alpha > 0$ and $\varepsilon > 0$ be fixed. Then:*

- *When n grows to ∞, $\Pr_{\boldsymbol{x}\in S_{n-1}}(\boldsymbol{x} \in [-\frac{\alpha}{\sqrt{n}}, \frac{\alpha}{\sqrt{n}}]^n)$ converges to 0 and $\Pr_{\boldsymbol{x}\in S_{n-1}}(\boldsymbol{x} \in [-\frac{\alpha \ln^{1/2+\varepsilon} n}{\sqrt{n}}, \frac{\alpha \ln^{1/2+\varepsilon} n}{\sqrt{n}}]^n)$ converges to 1.*
- *Let $0 < \beta < 1$, $a_n = \sqrt{2 \ln n}$ and $b_n = 1/a_n$. If $\beta + \varepsilon < 1$ then for all sufficiently large n: $\Pr_{\boldsymbol{x}\in S_{n-1}}(\boldsymbol{x} \in [\pm\frac{a_n - b_n \ln(-\ln(\beta+\varepsilon))}{\sqrt{n - \sqrt{n}\ln n}}]^n) \geq \beta$. If $\beta > \varepsilon$ then for all sufficiently large n: $\Pr_{\boldsymbol{x}\in S_{n-1}}(\boldsymbol{x} \in [\pm\frac{a_n - b_n \ln(-\ln(\beta-\varepsilon))}{\sqrt{n + \sqrt{n}\ln n}}]^n) \leq \beta$*

Proof. Take $m = \sqrt{n + \sqrt{n}\ln n}$. Classical tail bounds on the χ^2 distribution guarantee that $\Pr_{\boldsymbol{y}\leftarrow\mathcal{N}^n}(\|\boldsymbol{y}\| > m) \leq e^{-(\ln^2 n)/3}$ for all sufficiently large n. But $\Pr_{y\leftarrow\mathcal{N}}(|y| \leq \frac{\alpha}{\sqrt{n}}m) = \Pr_{y\leftarrow\mathcal{N}}(|y| \leq \alpha\sqrt{1+\frac{\ln n}{\sqrt{n}}}) \leq \Pr_{y\leftarrow\mathcal{N}}(|y| \leq 2\alpha)$ for all sufficiently large n. We conclude since $\Pr_{y\leftarrow\mathcal{N}}(|y| \leq 2\alpha)$ is a constant < 1.

For the second statement, take $m = \sqrt{n - \sqrt{n}\ln n}$. Similarly, it is known that $\Pr_{\boldsymbol{y}\leftarrow\mathcal{N}^n}(\|\boldsymbol{y}\| < m) \leq e^{-(\ln^2 n)/3}$. But $\Pr_{y\leftarrow\mathcal{N}}(|y| \leq \frac{\alpha\ln^{1/2+\varepsilon} n}{\sqrt{n}}m) \geq \Pr_{y\leftarrow\mathcal{N}}(|y| \leq \frac{\alpha\ln^{1/2+\varepsilon} n}{2})$ for all sufficiently large n. Tail bounds on the normal distribution then guarantee that $\Pr_{y\leftarrow\mathcal{N}}(|y| \leq \frac{\alpha\ln^{1/2+\varepsilon} n}{2})^n$ converges to 1, which concludes the first item.

For the second statement, we use a classical result from extreme value theory: the maximum X_n of n independent normal variables follows approximately a Gumbel distribution. More precisely, for any real x, the probability that $X_n \leq a_n + xb_n$ converges to $e^{-e^{-x}}$ as n grows to ∞. To conclude, we choose x such that $e^{-e^{-x}} = \beta \pm \varepsilon$ and use the previous tail bounds for the χ^2 distribution with $m = \sqrt{n \pm \sqrt{n}\ln n}$. □

Note that asymptotically: $\frac{a_n - b_n\ln(-\ln(\beta+\varepsilon))}{\sqrt{n-\sqrt{n}\ln n}} \sim \frac{a_n - b_n\ln(-\ln(\beta-\varepsilon))}{\sqrt{n+\sqrt{n}\ln n}} \sim \sqrt{\frac{2\ln n}{n}}$.

6.2 Algorithmic Bounds

Lemma 9 immediately gives rise to efficient algorithms to compute lower and upper bounds on the probability $p(n, H)$. It suffices to perform an exhaustive search over m, using the range of the χ^2 distribution: for instance, one selects the best lower bound and the best upper bound among many values of m. To compute the bound, it suffices to compute the cdf of the normal distribution, and that of the χ^2 distribution, which are readily available in `sage` and the `boost` library. Depending on H, the bounds obtained may not be tight, especially when $p(n, H)$ is small, but they are rigorous and easy to compute. They can also be helpful to guess when Monte Carlo sampling estimates are feasible.

7 Experiments and NTRU's NIST submission

We implemented the new hybrid attack and performed various experiments to check estimates. In this section, we report on these experiments, and discuss the security estimates given in NTRU's NIST submission with respect to the hybrid attack.

7.1 Reproducing Howgrave-Graham's experiments

In [19], Howgrave-Graham reported limited experimental results: he implemented a toy example for $N = 53$. We experimented with the same toy example, taking the same distribution. We checked the reliability of the admissibility model, by comparing the actual probability with that of the model. For a given

NTRU key, we identified all the triplets $(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v})$ such that $\boldsymbol{t}$ had the right number of zeros in the last k coordinates (as in [19]). Then we generated a large number of random LLL-reduced bases for M/Λ and for each triplet, we counted how many bases made the triplet admissible. The average admissibility probability was $0.01837 \approx 2^{-5.7}$, which is a bit higher than reported in [19]: however, it should be noted that the admissibility defined in [19] is slightly different from ours, because [19] also took into account the q-vectors, while we separate the LSH part from the admissibility part.

Under the admissibility model, we computed the idealized admissibility probability by repeatedly sampling over a sphere and a box: we obtained an average probability of $2^{-5.3}$ but with a high standard deviation of the order of the expectation. This means that the admissibility probability is not precise, and should only be interpreted as a rough order of magnitude. This can intuitively be explained as follows: the admissibility probability involves a randomization over a box, but it is well-known that in high dimension, most of the mass is located in the boundaries of the box, so when the box gets moved randomly, it will change significantly. We note that the experimental probability is consistent with the admissibility model. We also observe that in accordance with the admissibility model, the triplets for which the norm of $\tau(\boldsymbol{t})$ was shorter had experimentally the highest probability of being admissible.

Table 2 compares the figures of [19, Table p164] with approximate admissibility probability obtained by sampling, as well as the bounds of Sect. 6. Our figures are consistent with [19].

N	Row number in [19, Table p164]	p_s from [19]	Lower bound on p_s	p_s by sampling	Upper bound on p_s
53	1	$2^{-6.3}$	$2^{-6.6}$	$2^{-5.3}$	$2^{-3.6}$
107	2	$2^{-8.6}$		$2^{-9.2}$	
251	3	$2^{-6.8}$		$2^{-6.4}$	
251	4	2^{-13}		$2^{-11.8}$	
251	5	$2^{-20.4}$	$2^{-19.3}$	$2^{-19.2}$	2^{-12}

Table 2. Comparison of average admissibility probabilities

7.2 Security estimates of NTRU's NIST submission

First, we discovered several inconsistencies between the NTRU documentation [7] and what is actually implemented in the pari/gp scripts (included in the NTRU submission package) used to evaluate the cost of various attacks and to generate safe parameters.

In particular, the parameter s is defined in [7, Sect. 6.4.1] as $s = (q/8 - 2)/(n-1)$ with a correct numerical example, while the script actually computes

s as $s = \sqrt{(q/8-2)/n}$. This parameter s is needed to approximate the norm of $\tau(\boldsymbol{t})$ for checking admissibility, as well as to estimate the cost of the meet-in-the-middle-search.

The document [7] assumes that the meet-in-the-middle search is performed over k coordinates of f, even though f is much more dense than g. However, the script works as if the meet-in-the-middle search was performed over k coordinates of g, as if the attack was able to replace f by g, which is not consistent with the description of the attack: we are able to do that because we can use the reverse HNF basis and other permuted HNF.

To illustrate the issues, we consider the `ntruhps2048677` parameter set: [7, Table 6] and the script identifies the best non-local hybrid attack as follows. After some exhaustive search, the script outputs $k = 217$ and $r = 676 - 550 = 126$, and estimates the cost of the MITM stage to be 2^{144}, matching approximately the cost of a SVP-oracle in blocksize 494. If we were performing an exhaustive search over 217 ternary coordinates, it would cost 3^{217} so the cost of the MITM stage should have been $\approx \sqrt{3^{217}} \approx 2^{172} \gg 2^{144}$. This proves that the script is not consistent with [7]. Looking at the script, the announced 2^{144} is obtained by estimating the entropy η of 217 coordinates of g, and return $2^{\eta/2}$. However, if we target the coordinates of g, we can obtain much better figures thanks to our permuted HNF, as explained in Sect. 5.6. We performed experiments to see how good was the complexity 2^{144}: if we select 218 coordinates at random among the first n coordinates, there is an experimental probability $\geq 20\%$ that at least one of the n target vectors has exactly 64 non-zero coordinates (among the 218), and a probability $\geq 50\%$ that at least one of the n target vectors has at most 64 non-zero coordinates. By Sect. 5.6, we would find such a combination for an MITM cost of $2^{126.7}$, nearly 160,000 times faster. Here, the figure $2^{126.7}$ does not take into account the 20% probability: if we take it into account, then we can obtain even better trade-offs by targetting 52 non-zero coordinates for an overall MITM cost of $2^{125.4}$.

However, this does not mean that the whole attack can be sped up by the same factor, since the script selects parameters to balance the cost of lattice reduction with that of MITM: here, only MITM would be much faster, whereas lattice reduction would remain as expensive. So we modified the NTRU scripts to take into account our improved MITM search. The revised scripts output new figures for the best hybrid attack, for a total cost of 2^{136}, that is 8 bits less of security.

However, we turned our attention to the admissibility probability which was admittedly ignored in [7]. For again the initial best hybrid attack selected by the script (that is, $k = 217$ and $r = 676 - 550 = 126$), our methods of Sect. 6 show that the admissibility probability is $\leq 10^{-14} \approx 2^{-46}$, and possibly much smaller since we obtain an average lower bound on the order of $3 \times 10^{-21} \approx 2^{-68}$. This is because the condition on the last Gram-Schmidt norm of the reduced basis B was not correctly identified in [7]: we need a bigger norm to increase the admissibility probability.

Hence, the hybrid attack cost estimates provided by NTRU's scripts are not reliable: there is a mixture of significant underestimates and overestimates, which makes precise comparisons with other attacks very debatable. The security estimates need to be revised, but the same rigour must be applied to the estimates of other lattice attacks assessed by the script, such as the primal attack.

References

[1] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer. "Revisiting the Expected Cost of Solving uSVP and Applications to LWE". In: *Proc. ASIACRYPT 2017, Part I.* Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 297–322.

[2] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. "Post-quantum Key Exchange - A New Hope". In: *Proc. 25th USENIX*. USENIX, 2016, pp. 327–343.

[3] Y. Aono and P. Q. Nguyen. "Random sampling revisited: Lattice enumeration with discrete pruning". In: *Advances in cryptology—EUROCRYPT 2017 Part II.* Vol. 10211. LNCS. Full version on eprint. Springer, 2017, pp. 65–102.

[4] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS-Kyber (version 2.0) – Submission to round 2 of the NIST post-quantum project". Mar. 2019.

[5] L. Babai. "On Lovász' Lattice Reduction and the Nearest Lattice Point Problem". In: *Proc. STACS'85*. Vol. 182. LNCS. Springer, 1985, pp. 13–20.

[6] J. A. Buchmann, F. Göpfert, R. Player, and T. Wunderer. "On the Hardness of LWE with Binary Error: Revisiting the Hybrid Lattice-Reduction and Meet-in-the-Middle Attack". In: *Proc. AFRICACRYPT 2016*. Vol. 9646. Lecture Notes in Computer Science. Springer, 2016, pp. 24–43.

[7] C. Chen, O. Danba, J. Hoffstein, A. Hulsing, J. Rijneveld, J. M. Schanck, T. Saito, P. Schwabe, W. Whyte, K. Xagawa, T. Yamakawa, and Z. Zhang. "NTRU Algorithm Specifications And Supporting Documentation". Sept. 2020.

[8] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang. "NTRU Algorithm Specifications And Supporting Documentation". Mar. 2019.

[9] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren. "SABER: Mod-LWR based KEM (Round 2 Submission)". Mar. 2019.

[10] L. Ducas. "Shortest Vector from Lattice Sieving: A Few Dimensions for Free". In: *Proc. EUROCRYPT 2018, Part I.* Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 125–145.

[11] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS-Dilithium – Submission to round 2 of the NIST post-quantum project". Mar. 2019.

[12] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. "Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU". Mar. 2019.

[13] N. Gama and P. Q. Nguyen. "Predicting Lattice Reduction". In: *Proc. of Eurocrypt '08*. LNCS. Springer - Verlag, 2008, pp. 31–51.

[14] C. Hermite. "Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre". In: *J. Reine Angew. Math.* 40 (1850). Also available in the first volume of Hermite's complete works, published by Gauthier-Villars, pp. 279–290.

[15] P. S. Hirschhorn, J. Hoffstein, N. Howgrave-Graham, and W. Whyte. "Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches". In: *Proc. ACNS 2009*. Vol. 5536. Lecture Notes in Computer Science. 2009, pp. 437–455.

[16] J. Hoffstein, J. Pipher, and J. Silverman. "NTRU: A Ring Based Public Key Cryptosystem". In: *Proc. of ANTS III*. Vol. 1423. LNCS. Springer-Verlag, 1998, pp. 267–288.

[17] J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, W. Whyte, and Z. Zhang. *Choosing Parameters for NTRUEncrypt*. Cryptology ePrint Archive, Report 2015/708. `https://eprint.iacr.org/2015/708` Proceedings version at CT-RSA 2017. 2015.

[18] N. Howgrave-Graham, J. H. Silverman, and W. Whyte. *A Meet-In-The-Middle Attack on an NTRU Private Key*. Tech. rep. Technical Report #004, Version 2. NTRU Cryptosystems, 2003.

[19] N. Howgrave-Graham. "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU". In: *Proc. CRYPTO 2007*. Vol. 4622. Lecture Notes in Computer Science. Springer, 2007, pp. 150–169.

[20] P. Indyk and R. Motwani. "Approximate nearest neighbors: Towards removing the curse of dimensionality". In: *Proc. of 30th STOC*. ACM. 1998, pp. 604–613.

[21] G. Maze. "Natural density distribution of Hermite normal forms of integer matrices". In: *J. Number Theory* 131.12 (2011), pp. 2398–2408.

[22] P. Q. Nguyen and D. Pointcheval. "Analysis and Improvements of NTRU Encryption Paddings". In: *Advances in Cryptology - Proc. CRYPTO 2002*. Vol. 2442. Lecture Notes in Computer Science. Springer, 2002, pp. 210–225.

[23] P. Q. Nguyen and O. Regev. "Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures". In: *J. Cryptol.* 22.2 (2009), pp. 139–160.

[24] NIST. "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process". Available at `https://csrc.nist.gov/publications/detail/nistir/8309/final`.

[25] C. C. Rousseau and O. G. Ruehr. "Problems and Solutions". In: *SIAM Review* 39.4 (1997). Subsection: The Volume of the Intersection of a Cube and a Ball in N-space. Two solutions by Bernd Tibken and Denis Constales., pp. 779–786.

[26] J. M. Schanck. "Practical Lattice Cryptosystems: NTRUEncrypt and NTRUMLS". PhD thesis. University of Waterloo, 2015.
[27] Security Innovation. "NTRU Challenge". Available at `https://www.securityinnovation.com/products/ntru-crypto/ntru-challenge`.
[28] D. R. Stinson. "Some Baby-Step Giant-Step Algorithms for the Low Hamming Weight Discrete Logarithm Problem". In: *Mathematics of Computation* 71.237 (2002), pp. 379–391.
[29] C. van Vredendaal. "Reduced memory meet-in-the-middle attack against the NTRU private key". In: *LMS Journal of Computation and Mathematics* 19.A (2016), 43?57. DOI: `10.1112/S1461157016000206`.
[30] T. Wunderer. "A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack". In: *J. Mathematical Cryptology* 13.1 (2019), pp. 1–26.
[31] T. Wunderer. "Revisiting the Hybrid Attack: Improved Analysis and Refined Security Estimates". In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 733. URL: `http://eprint.iacr.org/2016/733`.

A Flaws in Analyses of Odlyzko's Attack

A.1 The 2003 Analysis of Howgrave-Graham, Silverman and Whyte

The analysis of [18] is informal, and makes several implicit assumptions.

A.2 The 2016 Analysis of van Vredendaal

In 2016, van Vredendaal [29] tried to make the analysis of [18] more formal, but we explain in this subsection that [29, Lemma 1]and its proof are actually incorrect. [29, Lemma 1] claims that if f and g are randomly chosen of degree $n-1$ with d coefficients set to 1, and under the assumption that the public key h is uniformly distributed over $\mathcal{R}$, the probability that g will change the address of $-f_2h$ is $(1-\frac{d}{nq})^n \approx e^{-d/q}$.

First, there is a typo in [29]: it should be the probability that g does not change the address, otherwise the statement would not match the proof. Second, we notice that the assumptions of the Lemma are inconsistent. The public key h is not independent from f and g, it is actually determined by them, so it is not meaningful to assume that h is uniformly distributed. Third, the proof of [29, Lemma 1] actually assumes that $-f_2h$ is uniformly distributed over $\mathcal{R}$, which does not follow from the assumption on h, as f_2 and h are not independent.

We now explain how one could try to correct [29, Lemma 1]. We know from our own analysis that g will change a sign in $-f_2h$ if and only if there is a 1-coefficient of g such that the corresponding coefficient in $-f_2h$ is equal to 0 or $q/2$. So if we assume that $-f_2h$ is uniformly distributed over $\mathcal{R}$, the probability that a coefficient in $-f_2h$ is equal to 0 or $q/2$ is equal to $2/q$. But there are exactly d coefficients of g equal to 1, the rest being equal to 0. So if we further assume that g is randomly chosen of degree $n-1$ with d coefficients set to

1, we get that the probability of g not changing any sign is $(1 - 2/q)^d$, which is independent of n. This already explains why [29, Lemma 1] was wrong: the probability should be independent of n. However, our model is still debatable, but less than in [29]: g and $-f_2 h$ are actually not independent, since h depends on g by definition.

The right way of avoiding these issues is to follow our methodology, by introducing randomness in the hashing.