New NTRU Records with Improved Lattice Bases

Elena Kirshanova[1,3] Alexander May[2] and Julian Nowakowski[2]

[1] Technology Innovation Institute, Abu Dhabi, UAE
elenakirshanova@gmail.com
[2] Ruhr-University Bochum, Bochum, Germany
{alex.may,julian.nowakowski}@rub.de
[3] I.Kant Baltic Federal University, Kaliningrad, Russia

Abstract. The original NTRU cryptosystem from 1998 can be considered the starting point of the great success story of lattice-based cryptography. Modern NTRU versions like NTRU-HPS and NTRU-HRSS are round-3 finalists in NIST's selection process, and also Crystals-Kyber and especially Falcon are heavily influenced by NTRU.

Coppersmith and Shamir proposed to attack NTRU via lattice basis reduction, and variations of the Coppersmith-Shamir lattice have been successfully applied to solve official NTRU challenges by Security Innovations, Inc. up to dimension $n = 173$.

In our work, we provide the tools to attack modern NTRU versions, both by the design of a proper lattice basis, as well as by tuning the modern *BKZ with lattice sieving* algorithm from the G6K library to NTRU needs.

Let n be prime, $\Phi_n := (X^n - 1)/(X - 1)$, and let $\mathbb{Z}_q[X]/(\Phi_n)$ be the cyclotomic ring. As opposed to the common belief, we show that switching from the Coppersmith-Shamir lattice to a basis for the cyclotomic ring provides benefits. To this end, we slightly enhance the *LWE with Hints* framework by Dachman-Soled, Ducas, Gong, Rossi with the concept of projections against *almost-parallel hints.*

Using our new lattice bases, we set the first cryptanalysis landmarks for NTRU-HPS with $n \in [101, 171]$ and for NTRU-HRSS with $n \in [101, 211]$. As a numerical example, we break our largest HPS-171 instance using the cyclotomic ring basis within 83 core days, whereas the Coppersmith-Shamir basis requires 172 core days.

We also break one more official NTRU challenges by Security Innovation, Inc., originally worth 1000$, in dimension $n = 181$ in 20 core years.

Our experiments run up to BKZ blocksizes beyond 100, a regime that has not been reached in analyzing cryptosystems so far.

Keywords: NTRU, Cryptanalysis, BKZ, Sieving

1 Introduction

Lattice-based cryptography has evolved as the most favourable candidate for building *efficient* post-quantum cryptosystems, because lattices seem to provide sufficiently hard problems in reasonable dimensions. This is in contrast to coding-based cryptography [ABB$^+$20,MAB$^+$21,ABC$^+$20] that usually requires significantly larger dimensions. However, when compared to code-based schemes, precisely estimating the security of lattice-based schemes is much more difficult, since the behavior of lattice reduction algorithms is not yet fully understood. Their analysis remains a tricky business that heavily relies on experimental data that sharpens the accuracy of lattice estimators.

The NTRU cryptosystem [HPS98] from 1998 can be considered a blueprint for most efficient lattice-based constructions [BDK$^+$18,FHK$^+$18]. Therefore, it may not come as a surprise that NTRU received a significant amount of cryptanalytic attention, both from the theoretical side as well as from experimental evaluations.

Lattice Attacks in Theory. NTRU is defined over the convolution polynomial ring $\mathbb{Z}_q[X]/(X^n-1)$. An NTRU secret key consists of two small norm polynomials $f, g \in \mathbb{Z}_q[X]/(X^n-1)$, f being invertible, with corresponding NTRU public key $h = f^{-1}g$. In 1997, Coppersmith and Shamir [CS97] already showed how to use h for defining a basis for a $2n$-dimensional lattice $\mathcal{L}_{\mathsf{CS}}$. Let $\mathbf{f}, \mathbf{g}$ denote the coefficient vectors of f, g. Then $(\mathbf{f}, \mathbf{g}) \in \mathcal{L}_{\mathsf{CS}}$, and $(\mathbf{f}, \mathbf{g})$ is presumably a shortest vector in $\mathcal{L}_{\mathsf{CS}}$.

The lattice $\mathcal{L}_{\mathsf{CS}}$ does not only contain $(\mathbf{f}, \mathbf{g})$, but also n rotations of $(\mathbf{f}, \mathbf{g})$. Observe that $X^i f \cdot h = X^i g$ for all $0 \leq i < n$. By construction, the coefficient vectors of $X^i f, X^i g$ are also contained in $\mathcal{L}_{\mathsf{CS}}$. Since we work modulo $X^n - 1$, a multiplication by X^i simply defines a cyclic rotation of the coefficient vector by i positions, and therefore all coefficient vectors have identical norm. These rotations were first used in May, Silverman [MS01] to speed up lattice basis reduction by dimension reduction. However, recently [DDGR20] showed that lattice reduction already benefits internally from the presence of many short rotations (i.e., it benefits even without dimension reduction).

Moreover, the n rotations of $(\mathbf{f}, \mathbf{g})$ define an n-dimensional sublattice $\mathcal{L}_{f,g} \subset \mathcal{L}_{\mathsf{CS}}$. Kirchner and Fouque [KF17] observed that for sufficiently large q, called the *overstretched* NTRU regime, lattice basis reduction finds a basis for $\mathcal{L}_{f,g}$ significantly earlier than predicted by the analysis for *secret key recovery* (SKR) of $(\mathbf{f}, \mathbf{g})$ in $\mathcal{L}_{\mathsf{CS}}$. Since $\mathcal{L}_{f,g}$ contains n exceptionally small vectors, we call $\mathcal{L}_{f,g}$ a *dense* sublattice of $\mathcal{L}_{\mathsf{CS}}$, and the detection of a basis of $\mathcal{L}_{f,g}$ when reducing $\mathcal{L}_{\mathsf{CS}}$ a *dense sublattice discovery* (DSD). Ducas and van Woerden [DvW21] showed that the overstretched regime, for which DSD happens early, requires $q = \Omega(n^{2.484})$.

NTRU with Hints. By design, NTRU parameters may fulfill further relations. E.g., for correctness of decryption most NTRU variants require $g(1) = 0$, or equivalently $\langle \mathbf{g}, \mathbf{1}^n \rangle = 0$. In the framework of [DDGR20] such a secret key relation is called a *perfect hint*. [DDGR20] provide a method for reducing the lattice dimension by 1 for every perfect hint.

For some parameter settings, we have $h(1) = 0$ implying the relation $\langle \mathbf{h}, \mathbf{1}^n \rangle = 0$. This implies that the short vector $\mathbf{v} = (\mathbf{1}^n, \mathbf{0}^n)$ is contained in the Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$. Although $\mathbf{v}$ is very short, it is useless for a cryptanalyst.

[DDGR20] provide a method to *project a lattice basis against* such useless vectors, thereby removing them from the lattice. [DDGR20] call this a *short vector hint*.

This projection also reduces the lattice dimension by 1, but may come at the cost of decreasing the lattice determinant, since a projection usually decreases vector lengths. Thus, it is not clear whether a *projection against* a useless lattice vector decreases the run time for finding a secret key in a lattice.

Lattice Attacks in Practice. There has been reasonable cryptanalytic effort for breaking instantiations of the original NTRU cryptosystem. This was further encouraged by *Security Innovation, Inc.* by publishing 11 challenges in dimensions $107 \leq n \leq 211$ with a prize money of 1000$ each [Incb]. These challenges stimulated the development of new attack techniques and their efficient implementations, such as Howgrave-Graham's lattice-hybrid technique [How07] that lead to the break of two challenges in dimension $n \in \{107, 113\}$. With Bounded Distance Decoding of Liu, Nguyen [LN13] five more challenges with $n \in \{131, 139, 149, 163, 173\}$ were solved by Ducas, Nguyen [Inca].

In their experiments, Ducas and Nguyen used *BKZ 2.0 lattice reduction with extreme pruning* [GNR10,CN11]. In the meantime, there has been significant progress in computing shortest vectors in theory via sieving [NV08,MV10,Duc18] and also in its practical G6K implementation [ADH+19,DSvW21] as a subroutine in BKZ reduction. This algorithmic improvement has however not led to improved NTRU cryptanalysis, though.

Moreover, modern NTRU versions such as the NIST round-3 finalists NTRU-HRS and NTRU-HPSS [HRSS17,CDH+19] have not yet experienced decent practical cryptanalysis.

1.1 Our results

Cyclotomic Ring. The polynomial $X^n - 1$ factors into $X^n - 1 = \prod_{d|n} \Phi_d$, where Φ_d denotes the *d-th cyclotomic polynomial*. Gentry [Gen01] showed that in the case of composite n, where $X^n - 1$ has many divisors, attacks on NTRU can improve significantly, when switching from the Coppersmith-Shamir lattice to a lattice defined over some ring $\mathbb{Z}_q[X]/(p)$, where $p \mid (X^n - 1)$ and $1 \ll \deg p \ll n$.

As a countermeasure against Gentry's attack, modern NTRU variants use prime n. In that case $X^n - 1$ has only the two divisors $\Phi_1 = X - 1$ and

$$\Phi_n = \frac{X^n - 1}{X - 1} = X^{n-1} + X^{n-2} + \ldots + X + 1.$$

Both Φ_1 and Φ_n have either too small or too large degree to successfully mount Gentry's attack.

Nevertheless, as an attacker, one still might be temped to work over the so-called *cyclotomic* ring $\mathbb{Z}_q[X]/(\Phi_n)$. Since Φ_n has degree $n - 1$, a canonical lattice for the cyclotomic ring analogous to the Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$ has dimension only $2(n - 1)$, thereby saving two dimensions over $\mathcal{L}_{\mathsf{CS}}$. However, the rotations of $(\mathbf{f}, \mathbf{g})$ modulo Φ_n have in general norm larger than $(\mathbf{f}, \mathbf{g})$ itself. Therefore, [DDGR20] conclude that the effect of saving two dimensions is likely outweighed by the increase in norm.

To avoid this issue, we define a new $2(n-2)$-dimensional lattice that retains the norm of all cyclic rotations. To this end, we take a closer look at the arithmetic of the cyclotomic ring $\mathbb{Z}_q[X]/(\Phi_n)$ and additionally introduce the following new concept of lattice hints, enriching the hint methodology of [DDGR20].

Almost-parallel Hint. Assume that we are looking for a short secret lattice vector $(\mathbf{f}, \mathbf{g})$ in a lattice $\mathcal{L}$, and we know another vector $\mathbf{v}$ (not necessarily in $\mathcal{L}$) *almost parallel to* $(\mathbf{f}, \mathbf{g})$. Then we may project $(\mathbf{f}, \mathbf{g})$ against $\mathbf{v}$, thereby eliminating the (long) component of $(\mathbf{f}, \mathbf{g})$ parallel to $\mathbf{v}$, and leaving the (short) component of $(\mathbf{f}, \mathbf{g})$ orthogonal to $\mathbf{v}$. As a result, our short lattice vector $(\mathbf{f}, \mathbf{g})$ is projected into an even shorter lattice vector, making the projection potentially easier to find by lattice reduction algorithms.

HPS, HRSS Results. Using our techniques, we define different lattice bases for NTRU-HPS and NTRU-HRSS, for $\mathbb{Z}_q[X]/(X^n - 1)$ and the cyclotomic ring $\mathbb{Z}_q[X]/(\Phi_n)$. Additionally, we include lattice hints by the design criteria of HPS and HRSS. We experimentally show that our lattice basis for the cyclotomic ring that retains rotation norms via an almost-parallel hint performs significantly better than the standard basis for $\mathcal{L}_{\mathsf{CS}}$. That is, we require smaller BKZ blocksizes to achieve secret key recovery (SKR), or if we are in the overstretched regime, for dense sublattice discovery (DSD).

Our smaller blocksizes result in a significant runtime decrease, e.g., we break HPS-161 with our cyclotomic lattice in 15 core days instead of 39 core days for $\mathcal{L}_{\mathsf{CS}}$, and HPS-171 in 83 core days instead of 172 core days. For HRSS the savings over $\mathcal{L}_{\mathsf{CS}}$ are even larger. As an example, we solved HRSS-161 with our cyclotomic lattice in 4 core hours versus 20 core hours for $\mathcal{L}_{\mathsf{CS}}$.

181-Challenge. Using our techniques, we are also able to solve an unbroken NTRU Challenge proposed by Security Innovation, Inc [Incb] with $n = 181$, thereby improving upon the previous $n = 173$ record of Ducas, Nguyen.

G6K Implementation. For the first time we apply the *BKZ with sieving* implementation G6K for NTRU cryptanalysis. We measure complexity by the minimal BKZ blocksize β that is required to achieve SKR or DSD. The current turnover point, where BKZ with sieving is superior to enumeration lies around $\beta = 65$ – for our implementations and our parallel hardware. In our experiments, we go beyond $\beta = 100$, a regime where sieving is clearly favourable, and that has not been reached so far for attacks on real-world lattice schemes. For NTRU-HPS we provide the first cryptanalysis landmarks in the range $101 \leq n \leq 171$ using ring modulus $q = 512$. For NTRU-HRSS we use the recommended larger moduli $q = 2048$ and $q = 4096$, which in turn allow us to break instances even in the range $101 \leq n \leq 211$.

1.2 Future Work

Existent estimators for NTRU [DDGR20,DvW21] consider the Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$. It appears to be difficult to translate these estimators to our new lattices. In particular, the Fatigue estimator from [DvW21] crucially relies on the circulant structure of $\mathcal{L}_{f,g}$, which is not preserved in our new lattices. We leave the estimates for our lattices and comparison with the existent ones for future work.

1.3 Organization of Our Paper

In Section 2 we provide some basic lattice facts and recall lattice estimates. Section 3 defines NTRU-HPS and NTRU-HRSS, as submitted to the NIST PQC competition [CDH+19]. In Section 4, we generalize the *LWE with a hint* framework of [DDGR20] and introduce our new concept of *almost-parallel hints*. Using the results of Section 4, we describe in Section 5 how to properly design a lattice basis over the cyclotomic ring for attacking NTRU-HPS and NTRU-HRSS, and we discuss its benefits over the Coppersmith-Shamir lattice in detail. Finally, in Section 6 we provide an extensive overview of our experimental results and discuss potential implications for the security of NTRU-HPS and NTRU-HRSS. We end in Section 7 with the details of our new record computation for $n = 181$.

The implementation accompanying our work, including a detailed documentation, can be found at https://github.com/ElenaKirshanova/ntru_with_sieving.

2 Preliminaries

2.1 Notations

We denote by $\mathbb{Z}_n$ the ring of integers modulo n and by $\mathbb{Z}_n^*$ its group of units. Lower case bold letters represent (row-)vectors. Upper case bold letters represent matrices. We denote by $\mathbf{I}_n$ the $n \times n$ identity matrix. The n-dimensional all-zero and all-one vectors are denoted by $\mathbf{0}^n$ and $\mathbf{1}^n$, respectively. For a polynomial $p \in \mathbb{Z}[X]$, we denote by $\mathbf{p}$ its coefficients vector. Conversely, for a vector $\mathbf{v} = (v_0, v_1, \ldots, v_n) \in \mathbb{Z}^{n+1}$, we denote by v the corresponding polynomial $v = \sum_{i=0}^{n} v_i X^i$. If a polynomial has coefficients in $\{0, 1, -1\}$, we call the polynomial and its coefficient vector *ternary*.

The Euclidean norm and the Euclidean inner product are denoted by $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$, respectively. For a vector $\mathbf{w} \in \mathbb{R}^n$, we denote by $\mathbf{w}^\perp \subseteq \mathbb{R}^n$ the subspace orthogonal to $\mathbf{w}$. We call $\mathbf{w}^\perp$ the orthogonal complement of $\mathbf{w}$.

For $\mathbf{v} \in \mathbb{R}^n$ we denote by $\pi_{\mathbf{w}}(\mathbf{v}) \in \mathbf{w}^\perp$ the orthogonal projection of $\mathbf{v}$ onto $\mathbf{w}^\perp$,

$$\pi_{\mathbf{w}}(\mathbf{v}) := \mathbf{v} - \frac{\langle \mathbf{v}, \mathbf{w} \rangle}{\|\mathbf{w}\|^2} \mathbf{w}. \tag{1}$$

We call the projection $\pi_{\mathbf{w}}$ onto $\mathbf{w}^\perp$ a projection *against* $\mathbf{w}$.

2.2 Lattices

For $\mathbf{B} \in \mathbb{Q}^{n \times m}$, we define $\mathcal{L}(\mathbf{B})$ as the lattice generated by the *rows* of $\mathbf{B}$:

$$\mathcal{L}(\mathbf{B}) := \{\mathbf{xB} \mid \mathbf{x} \in \mathbb{Z}^n\},$$

keeping our notation consistent with commonly used implementations of lattice-reduction algorithms. If the rows of $\mathbf{B}$ are linearly independent, we call $\mathbf{B}$ a *basis matrix* of $\mathcal{L}(\mathbf{B})$. The number of rows in a basis matrix is called the *dimension* of a lattice, denoted $\dim \mathcal{L}(\mathbf{B})$. The *determinant* of a lattice $\mathcal{L}$ with basis matrix $\mathbf{B}$ is defined as

$$\det \mathcal{L} := \det \sqrt{\mathbf{B}\mathbf{B}^T}.$$

Both the dimension and the determinant do not depend on the basis choice.

Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ denote the rows of a basis matrix $\mathbf{B}$. *Hadamard's inequality* states

$$\det \mathcal{L}(\mathbf{B}) \leq \prod_{i=1}^{n} \|\mathbf{b}_i\|.$$

The *dual* of a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ is defined as

$$\mathcal{L}^* := \{\mathbf{v} \in \operatorname{span}(\mathbf{B}) \mid \forall \mathbf{w} \in \mathcal{L} : \langle v, w \rangle \in \mathbb{Z}\}.$$

A vector $\mathbf{v} \in \mathcal{L}$ is called *primitive* with respect to some lattice $\mathcal{L}$, if it is not a multiple of a lattice vector, i.e., for every integer $k \geq 2$ it holds that $\frac{1}{k}\mathbf{v} \notin \mathcal{L}$.

A lattice $\mathcal{L} \subset \mathbb{Z}^n$ is called q-ary, if it contains $q\mathbb{Z}^n$.

2.3 Lattice Reduction

Every lattice of dimension at least two has infinitely many bases. In many applications (such as cryptanalysis), one wants to compute a *good* basis of a lattice, i.e., a basis consisting of short and almost orthogonal vectors.

The LLL lattice reduction algorithm [LLL82] computes in polynomial time a relatively good basis, whose shortest vector is exponentially (in the lattice dimension) longer than a shortest lattice vector. Its generalization, the BKZ algorithm [Sch87,CN11,AWHT16] provides a trade-off between runtime and basis quality.

The most important parameter of BKZ is the so-called *blocksize* β. The BKZ algorithm computes shortest vectors in projected sub-lattices of dimension β. This dominates the runtime of BKZ. The security of lattice-based cryptosystems is therefore usually measured in the required blocksize β to find a shortest vector. There are two main approaches to find a shortest vector: enumeration algorithms [Kan83,ABF$^+$20] implemented in [dt21a], and sieving algorithms [AKS01,NV08] implemented in [dt21b]. Sieving algorithms find a shortest vector in time and memory $2^{\mathcal{O}(n)}$, while enumeration requires $2^{\mathcal{O}(n \log n)}$ time and only $\operatorname{poly}(n)$ memory. Experiments with publicly available implementations of sieving and enumeration suggest [ADH$^+$19] that sieving algorithms are superior to enumeration starting from dimension ≈ 65.

Estimating the required β for a given lattice is an active area of research. The *NTRU Fatigue estimator* [DvW21] provides the most accurate estimates for NTRU lattices. It is based on the *probabilistic simulation method* introduced in [DDGR20], which in turn is a refinement of the so-called *2016 estimate* or *GSA intersect method* introduced in [ADPS16].

As the name suggests, this method is based on the *geometric series assumption (GSA)* [Sch03], which states that in a random lattice the Gram-Schmidt norms of the vectors of a BKZ-reduced basis decay geometrically. Intuitively, the 2016 estimate states, if a lattice $\mathcal{L}$ contains a sufficiently short vector $\mathbf{s}$, then for sufficiently large β, the GSA cannot hold. Hence, in that case the BKZ behaves differently than on a random lattice and therefore likely recovers $\mathbf{s}$.

More precisely, the 2016 estimate states that BKZ with blocksize β recovers a short vector $\mathbf{s}$ in a d-dimensional lattice $\mathcal{L}$, provided that

$$\sqrt{\beta/d}\|\mathbf{s}\| < \delta_\beta^{2\beta-d-1} \cdot \det(\mathcal{L})^{1/d},$$

where

$$\delta_\beta \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{1/\beta}\right)^{1/(2(\beta-1))}.$$

We refer the reader to the survey by Albrecht and Ducas [AD21] for further details.

A straight-forward calculation shows that for $\|\mathbf{s}\| := \gamma\sqrt{d}\det(L)^{1/d}$ with $0 < \gamma < 1$, the 2016 estimate predicts

$$\beta = \frac{d\log(d)}{\log(d) - 2\log(\gamma)} + o(d). \tag{2}$$

Hence, if $\|\mathbf{s}\|$ is very close to the so-called *Minkowski bound* of $\sqrt{d}\det(L)^{1/d}$, then the 2016 estimate predicts $\beta \approx d$. Conversely, if $\|\mathbf{s}\|$ is significantly smaller than $\sqrt{d}\det(L)^{1/d}$, then the estimate predicts $\beta \ll d$. As a consequence, it gets the easier for BKZ to find a shortest vector $\mathbf{s}$ in $\mathcal{L}$, the smaller the dimension d, the larger the determinant $\det\mathcal{L}$, and the shorter $\mathbf{s}$.

3 NTRU

In this section, we recall the definition of the NTRU cryptosystem, as defined in the submission to the NIST PQC competition [CDH+19], as well as the NTRU challenges published by Security Innovation, Inc [Incb].

3.1 NIST Submission

Let n be a prime number, where the order of both 2 and 3 in $\mathbb{Z}_n^*$ is $n-1$. We denote the set of all such primes by $\mathcal{N}$. The primes $n \in \mathcal{N}$ with $n \leq 1000$ are exactly the elements of the following set

$$\begin{aligned}\mathcal{N}_{\leq 1000} := \{&5, 19, 29, 53, 101, 139, 149, 163, 173, 197,\\ &211, 269, 293, 317, 379, 389, 461, 509, 557,\\ &653, 677, 701, 773, 797, 821, 859, 907, 941\}.\end{aligned}$$

The NTRU specification [CDH+19] defines

$$\begin{aligned}\Phi_n &:= \frac{X^n - 1}{X - 1} = X^{n-1} + X^{n-2} + \ldots + X + 1,\\ \mathcal{T} &:= \left\{ \sum_{i=0}^{n-2} v_i X^i \mid v_i \in \{-1, 0, 1\} \right\},\\ \mathcal{T}_+ &:= \left\{ \sum_{i=0}^{n-2} v_i X^i \in \mathcal{T} \mid \sum_i v_i v_{i+1} \geq 0 \right\},\\ \mathcal{T}(\omega) &:= \left\{ \sum_{i=0}^{n-2} v_i X^i \in \mathcal{T} \mid \begin{array}{l} v_i = +1 \text{ for } \frac{\omega}{2} \text{ coefficients } v_i, \\ v_i = -1 \text{ for } \frac{\omega}{2} \text{ coefficients } v_i \end{array} \right\}, \end{aligned} \tag{3}$$

where ω is an even positive integer. Notice, since n is prime, Φ_n is the n-th cyclotomic polynomial.

An NTRU private key is a tuple (n, q, f, g), where $n \in \mathcal{N}$, $q \in \mathbb{N}$ is a power of two and $f, g \in \mathbb{Z}[X]$ are polynomials with *small* coefficients. The corresponding public key is a tuple (n, q, h), where

$$h := 3gf_q \mod (q, X^n - 1) \tag{4}$$

and

$$f_q := f^{-1} \mod (q, \Phi_n). \tag{5}$$

If f is sampled from (a subset of) $\mathcal{T} \setminus \{0\}$, then the following lemma shows that f_q is well-defined.

Lemma 3.1. *Let $n \in \mathcal{N}$, $f \in \mathcal{T} \setminus \{0\}$ and let $q \in \mathbb{N}$ be a power of two. Then f is invertible in $\mathbb{Z}_q[X]/(\Phi_n)$.*

Proof. By definition of $\mathcal{N}$, the order of 2 in $\mathbb{Z}_n^*$ is $n - 1$. This implies that the n-th cyclotomic polynomial Φ_n is irreducible over $\mathbb{Z}_2$. (This easily follows from the fact that the subgroup $\{2^i \mod n \mid i \in \mathbb{Z}\} \subseteq \mathbb{Z}_n^*$ is isomorphic to the Galois group of the field extension $\mathbb{Z}_2 \subset \mathbb{Z}_2(\zeta_n)$, where ζ_n denotes a formal primitive n-th root of unity.) Hence, $\mathbb{Z}_2[X]/(\Phi_n)$ is a finite field. Since $f \in \mathcal{T}$ and $f \neq 0$, it follows that $f \not\equiv 0 \mod (2, \Phi_n)$. Therefore, f is invertible in $\mathbb{Z}_2[X]/(\Phi_n)$ and consequently also in $\mathbb{Z}_q[X]/(\Phi_n)$. □

The NTRU submission defines two different variants of the scheme, called *NTRU-HPS* and *NTRU-HRSS*. The variants use slightly different sample spaces for f and g, and have different constraints on q.

NTRU-HPS. In the HPS variant, the modulus q may be set to any power of 2. The polynomials f and g are sampled from the following two sets

$$\begin{aligned}\mathcal{L}_{f,\mathsf{HPS}} &:= \mathcal{T},\\ \mathcal{L}_{g,\mathsf{HPS}} &:= \mathcal{T}(\omega),\end{aligned} \tag{6}$$

where $\omega := \min\{q/8 - 2, 2\lfloor n/3 \rfloor\}$. The specification recommends to either use $q = 2048$ and $n \in \{509, 677\}$ or $q = 4096$ and $n = 821$. (The former imposes $\omega = 254$, the latter $\omega = 510$.)

NTRU-HRSS. In the HRSS variant, the modulus q is

$$q := 2^{\lceil 7/2 + \log_2(n) \rceil}. \tag{7}$$

The polynomials f, g are sampled from the following two sets

$$\begin{aligned}\mathcal{L}_{f,\mathsf{HRSS}} &:= \mathcal{T}_+,\\ \mathcal{L}_{g,\mathsf{HRSS}} &:= (X-1)\cdot\mathcal{T}_+.\end{aligned} \tag{8}$$

The specification recommends to set $n = 701$. (This imposes $q = 8192$.)

The NTRU Key Equation. From the definitions of $\mathcal{L}_{g,\mathsf{HPS}}$ and $\mathcal{L}_{g,\mathsf{HRSS}}$ it follows that in both NTRU-HPS and NTRU-HRSS $g(1) = 0$ and by Equation (4) consequently also $h(1) \equiv 0 \mod q$. Hence, from the Chinese remainder theorem

$$\mathbb{Z}_q[X]/(X^n - 1) \simeq \mathbb{Z}_q[X]/(\Phi_n) \times \mathbb{Z}_q[X]/(X-1)$$

it follows that in both variants the keys satisfy the *NTRU key equation*

$$fh \equiv 3g \mod (q, X^n - 1). \tag{9}$$

3.2 NTRU Challenges

When compared to the NTRU variants submitted to NIST PQC competition, the NTRU challenges by Security Innovation, Inc. use a quite different key format. Let $n \in \mathbb{N}$ and define

$$\mathcal{T}_{\mathsf{Ch.}}(d_i) := \left\{ \sum_{i=0}^{n-1} v_i X^i \;\middle|\; \begin{array}{l} v_i = +1 \text{ for } d_i\text{-many } v_i, \\ v_i = -1 \text{ for } d_i\text{-many } v_i \end{array} \right\} \tag{10}$$

for some parameter $d_i \in \mathbb{N}$.[1]

The sample spaces for the secret polynomials f and g are defined as follows

$$\mathcal{L}_{f,\mathsf{Ch.}}(d_1, d_2, d_3) := \left\{ f_1 f_2 + f_3 \mod X^n - 1 | f_i \in \mathcal{T}_{\mathsf{Ch.}}(d_i) \right\}, \tag{11}$$

[1] As opposed to the set $\mathcal{T}(\omega)$, defined in Equation (3), the elements of $\mathcal{T}_{\mathsf{Ch.}}(d_i)$ are of degree at most $n-1$, instead of $n-2$. In addition, they have $2d_i$ non-zero coefficients, instead of ω.

$$\mathcal{L}_{g,\mathsf{Ch.}}(d_g) := \left\{ \sum_{i=0}^{n-1} g_i X^i \;\middle|\; \begin{array}{l} g_i = +1 \text{ for } (d_g+1)\text{-many } g_i, \\ g_i = -1 \text{ for } d_g\text{-many } g_i \end{array} \right\}, \tag{12}$$

where $d_1, d_2, d_3, d_g \in \mathbb{N}$ are some positive integers.

The public polynomial h is defined via the equation

$$(1+3f)h \equiv g \mod (q, X^n - 1),$$

where q is a power of two.

Security Innovation, Inc. published 27 different challenges. 11 of the challenges used to be worth 1000$ each, the other 16 challenges used to be worth 5000$ each. The concrete parameters for all 1000$ challenges are given in Table 1.

n	q	d_1	d_2	d_3	d_g	Solved by
107	512	4	4	4	36	Howgrave-Graham
113	1024	5	4	3	38	Howgrave-Graham
131	1024	5	4	4	44	Ducas, Nguyen
139	1024	5	5	3	46	Ducas, Nguyen
149	1024	5	5	3	50	Ducas, Nguyen
163	1024	5	5	4	54	Ducas, Nguyen
173	1024	6	5	4	58	Ducas, Nguyen
181	1024	6	5	4	60	**This work**
191	1024	6	5	4	64	-
199	1024	6	5	6	66	-
211	1024	6	6	4	70	-

Table 1. Parameters of the 1000$ NTRU challenges.

4 Lattice Reduction With a Hint

When attacking a cryptographic lattice $\mathcal{L}$, one often knows some side information about the secret short vector $\mathbf{s} \in \mathcal{L}$. Dachman-Soled, Ducas, Gong and Rossi (DDGR) introduced in [DDGR20] a framework for integrating such *hints* into $\mathcal{L}$ to improve the performance of lattice reduction algorithms. In this section, we recall two types of hints and additionally introduce a new type of hint.

Perfect Hints. Suppose we know a vector $\mathbf{v}$, which is orthogonal to the secret vector $\mathbf{s}$, i.e.,

$$\langle \mathbf{s}, \mathbf{v} \rangle = 0.$$

This type of hint is called a *perfect hint.*

Instead of searching for $\mathbf{s}$ in $\mathcal{L}$, we can then search in the sub-lattice $\mathcal{L} \cap \mathbf{v}^{\perp}$. As shown by DDGR, this may make reducing the lattice a bit easier, since it decreases the dimension by one and additionally may increase the determinant (assuming that $\mathbf{v}$ is not too far from being primitive with respect to the dual):

Lemma 4.1 (Generalization of Lemma 12 in [DDGR20]). *Let $\mathcal{L}$ be a lattice and let $\mathbf{v} \in \mathcal{L}^*$. Let $k \in \mathbb{N}$ such that $\frac{1}{k}\mathbf{v}$ is primitive with respect to $\mathcal{L}^*$. Then $\mathcal{L} \cap \mathbf{v}^\perp$ is a lattice of dimension $\dim \mathcal{L} - 1$. Its determinant is given by*

$$\det(\mathcal{L} \cap \mathbf{v}^\perp) = \frac{\|\mathbf{v}\|}{k} \cdot \det \mathcal{L}.$$

Worth noting, if k is significantly larger than $\|\mathbf{v}\|$, incorporating a perfect hint may actually be counterproductive for lattice reduction algorithms, as the disadvantage of having a smaller determinant then may outweigh the benefit of losing one dimension. DDGR heuristically assume that k always equals 1. However, in Section 5.3 we show that this is not the case for typical NTRU lattices.

Given a basis $\mathbf{B}$ for $\mathcal{L}$, DDGR suggest the following polynomial time algorithm to compute a basis for the sub-lattice $\mathcal{L} \cap \mathbf{v}^\perp$.

1. Compute the dual basis $\mathbf{D}$ of $\mathbf{B}$. (Recall that $\mathbf{D}$ is given by $\mathbf{D} = (\mathbf{B}^+)^T$, where $\cdot^+$ and $\cdot^T$ denote the Moore-Penrose pseudoinverse and transpose, respectively.)
2. Compute $\mathbf{D}_\perp := \pi_\mathbf{v}(\mathbf{D})$, where $\pi_\mathbf{v}$ is applied row-wise to $\mathbf{D}$.
3. Apply the LLL algorithm to $\mathbf{D}_\perp$ to eliminate linear dependencies. Then delete the first (all-zero) row.
4. Output the dual of the resulting matrix.

Short Vector Hints. Many cryptographic lattices contain short-ish vectors that neither reveal the secret key, nor help the decryption. For instance, in some NTRU variants, the lattices contain the all-one vector. Even though this vector is very short, it cannot be used for decryption. Futhermore, almost all cryptographic lattices are q-ary for some small $q \in \mathbb{N}$ and thus contain the rather short *q-vectors* $(0, \dots, 0, q, 0, \dots, 0)$.

It can be sometimes beneficial for lattice reduction algorithms to *remove* these vectors from the lattice, i.e., to project the lattice onto their orthogonal complement. DDGR call this a *short vector hint.*

The benefit of projecting a lattice $\mathcal{L}$ against $\mathbf{v} \in \mathcal{L}$ is that the dimension decreases by one. However, as the following lemma shows, at the same time the determinant shrinks by a factor of $\|\mathbf{v}\|$. A short vector hint is therefore always a trade-off between decreased dimension and decreased determinant.

Lemma 4.2 (Fact 14 in [DDGR20]). *Let $\mathcal{L}$ be a lattice and let $\mathbf{v} \in \mathcal{L}$ be primitive with respect to $\mathcal{L}$. Then*

$$\det(\pi_\mathbf{v}(\mathcal{L})) = \det(\mathcal{L})/\|\mathbf{v}\|.$$

In contrast to a perfect hint, where the constraint on $\mathbf{v}$ being primitive is a potential disadvantage, this is not the case for a short vector hint: if $\mathbf{v}$ is not primitive, i.e., if there exists $k \geq 2$, such that $\frac{1}{k}\mathbf{v}$ is a primitive vector of $\mathcal{L}$, then projecting actually shrinks the determinant by less than a $\|\mathbf{v}\|$-factor, since in that case we have

$$\det(\pi_{\mathbf{v}}(\mathcal{L})) = \det(\pi_{\frac{1}{k}\mathbf{v}}(\mathcal{L})) = k \cdot \frac{\det(\mathcal{L})}{\|\mathbf{v}\|} > \frac{\det(\mathcal{L})}{\|\mathbf{v}\|}.$$

A potential drawback from projecting against $\mathbf{v}$ is a loss of information. While $\mathbf{s}$ is contained in $\mathcal{L}$, the projection $\pi_{\mathbf{v}}(\mathcal{L})$ only contains $\pi_{\mathbf{v}}(\mathbf{s})$, from which one has to recover $\mathbf{s}$.

Almost-Parallel Hints (new). In our attacks, the secret vector $\mathbf{s}$ is sometimes *almost parallel* to some known vector $\mathbf{v}$ (not necessarily included in our lattice), i.e., it has a decomposition into

$$\mathbf{s} = c\mathbf{v} + \mathbf{s}',$$

for some vector $\mathbf{s}'$ significantly shorter than $\mathbf{s}$, and some scalar c. We call this an *almost-parallel hint.*

Projecting the lattice against $\mathbf{v}$ makes the secret target vector $\mathbf{s}$ significantly shorter:

$$\|\pi_{\mathbf{v}}(\mathbf{s})\| = \|\pi_{\mathbf{v}}(\mathbf{s}')\| \leq \|\mathbf{s}'\|.$$

In addition, by integrating an almost-parallel hint, we also decrease the dimension of $\mathcal{L}$ by one.[2]

As with a short vector hint, using an almost-parallel hint also comes with a disadvantage: both types of hints decrease the determinant. In fact, the following straight-forward generalization of Lemma 4.2 shows, in contrast to a short vector hint, where the determinant only shrinks by a $\|\mathbf{v}\|$-factor, an almost-parallel may shrink way more significantly (assuming that only large multiples of $\mathbf{v}$ are contained in $\mathcal{L}$):

Lemma 4.3. *Let $\mathcal{L}$ be lattice and let $\mathbf{v}$ be a vector. If there exists $\lambda \in \mathbb{R}$ such that $\lambda\mathbf{v}$ is a primitive lattice vector of $\mathcal{L}$, then*

$$\det(\pi_{\mathbf{v}}(\mathcal{L})) = \det(\mathcal{L})/(\lambda\|\mathbf{v}\|).$$

Notice that for an integral vector $\mathbf{v} \in \mathbb{Z}^n \backslash \mathbf{0}^n$ and a q-ary lattice $\mathcal{L}$ we shrink, however, at most by a $q\|\mathbf{v}\|$-factor, because in that case there exists $\lambda \in \{1, 2, \dots, q\}$, such that $\lambda\mathbf{v}$ is primitive.

As in the case of short vector hints we potentially lose information with projecting against $\mathbf{v}$, since $\mathcal{L}$ contains $\mathbf{s}$, whereas $\pi_{\mathbf{v}}(\mathcal{L})$ only contains $\pi_{\mathbf{v}}(\mathbf{s})$. However, in our applications of almost-parallel hints we are always able to efficiently recover $\mathbf{s}$ from its projection $\pi_{\mathbf{v}}(\mathbf{s})$.

[2] We assume that a multiple of $\mathbf{v}$ is included in $\mathcal{L}$. For an integral vector $\mathbf{v}$ and a q-ary lattice $\mathcal{L}$, this certainly is the case, since $q\mathbf{v} \in \mathcal{L}$. If $\mathcal{L}$ contains no multiple of $\mathbf{v}$, then $\pi_{\mathbf{v}}(\mathcal{L})$ might not be a lattice.

5 Choosing Lattices for NTRU-HPS and NTRU-HRSS

5.1 The Coppersmith-Shamir Lattice

Let (n, q, h) be an NTRU-HPS or NTRU-HRSS public key with corresponding secret key (n, q, f, g). The most straight-forward approach for attacking NTRU is to consider the following lattice

$$\mathcal{L}_{\mathsf{CS}} := \{(\mathbf{v}, \mathbf{w}) \in \mathbb{Z}^{2n} \mid vh \equiv 3w \mod (q, X^n - 1)\},$$

which was first introduced by Coppersmith and Shamir (CS) in [CS97]. A basis matrix for $\mathcal{L}_{\mathsf{CS}}$ is given by

$$\mathbf{B}_{\mathsf{CS}} := \begin{pmatrix} \mathbf{I}_n & \mathbf{H} \\ \mathbf{0} & q\mathbf{I}_n \end{pmatrix}, \tag{13}$$

where for $i = 0, \ldots, n-1$, the $(i+1)$-st row of $\mathbf{H} \in \mathbb{Z}^{n \times n}$ is defined as the coefficient vector of

$$3^{-1} X^i h \mod (q, X^n - 1).$$

By Equation (9), $\mathcal{L}_{\mathsf{CS}}$ contains the vector $(\mathbf{f}, \mathbf{g}) \in \mathbb{Z}^{2n}$ corresponding to the secret polynomials f and g. Since f and g have very small coefficients, $(\mathbf{f}, \mathbf{g})$ likely is a shortest vector in $\mathcal{L}_{\mathsf{CS}}$. Hence, we can compute the secret key by running lattice reduction on $\mathbf{B}_{\mathsf{CS}}$.

Presence of Many Short Vectors. A remarkable property of $\mathcal{L}_{\mathsf{CS}}$ is that the lattice not only contains the short vector $(\mathbf{f}, \mathbf{g})$, but also all *rotations* of the secret key, i.e., the coefficient vectors corresponding to $X^i f$ and $X^i g$ for every $i \in \{0, 1, ..., n-1\}$. It is well known that the rotations also serve as valid secret keys.

Notice that the rotations have the same norm as $(\mathbf{f}, \mathbf{g})$, since multiplication by X in $\mathbb{Z}_q[X]/(X^n - 1)$ simply corresponds to a rotation of the coefficients. As discussed in [DDGR20, Section 6.3], the presence of these many short vectors makes finding the secret key a bit easier than it would be in a lattice containing (up to sign) only one short vector. Intuitively this is caused by the fact that the probability of BKZ finding *at least one* of the short vectors is higher than the probability of finding *one fixed* short vector.

Dense Sublattice. The rotations of the secret key generate an n-dimensional sub-lattice $\mathcal{L}_{f,g} \subset \mathcal{L}_{\mathsf{CS}}$. This sub-lattice is unusually *dense*, i.e., its determinant is much smaller than what we would expect from a random lattice: using Hadamard's inequality, we find that the determinant is upper bounded by

$$\det \mathcal{L}_{f,g} \leq \|(\mathbf{f}, \mathbf{g})\|^n.$$

As shown in [KF17] and refined in [DvW21], if q is sufficiently large, the presence of such a dense sub-lattice violates a prediction on the behavior of sublattices based on the GSA and thus forces BKZ to behave differently on $\mathcal{L}_{\mathsf{CS}}$, than it would on a random lattice. Indeed, it turns out that we can recover the secret key in that case using significantly smaller blocksizes.

According to the 2016 estimate, *secret key recovery (SKR)* normally would happen at blocksize $\beta = \tilde{\Theta}(n/\ln q)$. However, as shown in [KF17, Theorem 9], if q is large, BKZ recovers at significantly smaller blocksize $\beta = \tilde{\Theta}(n/\ln^2 q)$ a basis $\mathcal{B}_{f,g}$ for $\mathcal{L}_{f,g}$ – from which one easily obtains the secret key. (This event is called *dense sub-lattice discovery (DSD).*) For instance, instead of running BKZ on the $2n$-dimensional lattice $\mathcal{L}_{\mathsf{CS}}$, one then may run it on the n-dimensional sub-lattice $\mathcal{L}_{f,g}$, which is significantly easier.

NTRU parameters, that have this property, are called *overstretched*. Ducas and van Woerden [DvW21, Claim 3.5] showed that NTRU variants with $\|(\mathbf{f},\mathbf{g})\| = \mathcal{O}(n^{1/2})$ (such as HPS and HRSS) become overstretched when $q = \Omega(n^{2.484})$. As shown in [DvW21, Section 5.3], the asymptotic bound already holds for reasonably small values of n.

5.2 The Cyclotomic Lattice and the Projected Cyclotomic Lattice

The NTRU key equation holds not only over the convolution polynomial ring $\mathbb{Z}_q[X]/(X^n-1)$, but also over the cyclotomic polynomial ring $\mathbb{Z}_q[X]/(\Phi_n)$. Instead of working with the CS lattice, one might therefore be tempted to work with the following lattice

$$\mathcal{L}_\Phi := \{(\mathbf{v},\mathbf{w}) \in \mathbb{Z}^{2(n-1)} \mid vh \equiv 3w \mod (q,\Phi_n)\},$$

which we call the *cyclotomic lattice*. Analogously to the CS lattice, one can easily compute a basis matrix for $\mathcal{L}_\Phi$ as

$$\mathbf{B}_\Phi := \begin{pmatrix} \mathbf{I}_{n-1} & \mathbf{H}_\Phi \\ \mathbf{0} & q\mathbf{I}_{n-1} \end{pmatrix}, \tag{14}$$

where for $i = 0,\dots,n-2$, the $(i+1)$-st row of $\mathbf{H}_\Phi$ is defined as the coefficient vector of

$$3^{-1}X^i h \mod (q,\Phi_n).$$

Since the cyclotomic lattice has dimension only $2(n-1)$ instead of $2n$, one might hope that reducing it may be a bit easier than reducing the CS lattice.

Dachman-Soled, Ducas, Gong and Rossi [DDGR20, Section 6.3], however, doubt whether using $\mathcal{L}_\Phi$ really is beneficial. They argue, since multiplication by X in $\mathbb{Z}[X]/(\Phi_n)$ does not correspond to a simple rotation of the coefficients (as it does in $\mathbb{Z}[X]/(X^n-1)$), the length of the vectors corresponding to $X^i f$, $X^i g$ may be increased “significantly” in $\mathcal{L}_\Phi$. Accordingly, $\mathcal{L}_\Phi$ will contain fewer short secret key vectors than $\mathcal{L}_{\mathsf{CS}}$ and recovering them should therefore probably be harder.

This issue, however, can easily be fixed. To this end, let us take a closer look at the arithmetic in $\mathbb{Z}[X]/(\Phi_n)$.

Lemma 5.1. *Let* $p = \sum_i p_i X^i$ *be a polynomial of degree at most* $n-2$*. For every* $k \in \{0, 1, \ldots, n-1\}$ *it holds that*

$$X^k p \equiv \sum_{i=0}^{k-2} p_{i+n-k} X^i + \sum_{i=k}^{n-2} p_{i-k} X^i - \sum_{i=0}^{n-2} p_{n-k-1} X^i \mod \Phi_n. \tag{15}$$

Furthermore, for every $k \in \mathbb{N}$ *it holds that*

$$X^k p \equiv X^{(k \bmod n)} p \mod \Phi_n. \tag{16}$$

Proof. From $X^n - 1 = \Phi_n(X-1) \equiv 0 \mod \Phi_n$ it follows that

$$X^n \equiv 1 \mod \Phi_n. \tag{17}$$

Notice that this already proves Equation (16).

Writing $\Phi_n = X^{n-1} + X^{n-2} + \ldots X + 1$, we obtain

$$X^{n-1} \equiv -(X^{n-2} + X^{n-3} + \ldots X + 1) \mod \Phi_n. \tag{18}$$

To prove Equation (15), we now simply apply Equations (17) and (18) to the following identity:

$$X^k p = \sum_{i=k}^{n-2} p_{i-k} X^i + p_{n-k-1} X^{n-1} + \sum_{i=0}^{k-2} p_{i+n-k} X^{n+i}.$$

□

From Lemma 5.1 it follows that the arithmetic in $\mathbb{Z}[X]/(\Phi_n)$ is actually quite similar to that in $\mathbb{Z}[X]/(X^n - 1)$. First, in both rings multiplication by X is n-periodic. Second, while in $\mathbb{Z}[X]/(\Phi_n)$ a multiplication by X does not perfectly correspond to a rotation of the coefficients, one might still view it as a *rotation with an extra step*: in $\mathbb{Z}[X]/(X^n - 1)$, we have the following identity[3]

$$X^k p = \sum_{i=0}^{k-2} p_{i+n-k} X^i + \sum_{i=k}^{n-1} p_{i-k} X^i \mod X^n - 1. \tag{19}$$

Comparing Equation (19) with Equation (15), we find that in $\mathbb{Z}[X]/(\Phi_n)$ the coefficients first get rotated exactly as they would in $\mathbb{Z}[X]/(X^n - 1)$, but then the leading coefficient p_{n-k-1} gets removed and is instead subtracted from all other coefficients.

Let us illustrate with an example. Consider the polynomial

$$p := 1 + X^2 - X^3.$$

[3] Notice that there is no monomial of degree $k-1$ in Equation (19), since p has no monomial of degree $n-1$.

The coefficient vectors of $X^0 p, \dots, X^4 p$ modulo $X^5 - 1$ are

$$\begin{gathered}(1,0,1,-1,0),\\ (0,1,0,1,-1),\\ (-1,0,1,0,1),\\ (1,-1,0,1,0).\end{gathered}$$

The coefficient vectors modulo $\varPhi_5$ on the other hand are

$$\begin{aligned}(1,0,1,-1) - 0 \cdot (1,1,1,1) &= (1,0,1,-1),\\ (0,1,0,1) - (-1) \cdot (1,1,1,1) &= (1,2,1,2),\\ (-1,0,1,0) - 1 \cdot (1,1,1,1) &= (-2,-1,0,-1),\\ (1,-1,0,1) - 0 \cdot (1,1,1,1) &= (1,-1,0,1),\\ (0,1,-1,0) - 1 \cdot (1,1,1,1) &= (-1,0,-2,-1).\end{aligned}$$

Hence, the rotations of p modulo $\varPhi_n$ are the sum of a short vector and a multiple of $\mathbf{1}^{n-1}$, i.e., the rotations are *almost parallel* to $\mathbf{1}^{n-1}$.

This suggests to incorporate two almost-parallel hints to $\mathcal{L}_{\varPhi}$, i.e., to work with the following lattice, which we call the *projected cyclotomic lattice*

$$\begin{aligned}\mathcal{L}_{\varPhi,\perp} &:= \{(\pi_{\mathbf{1}^{n-1}}(\mathbf{v}), \pi_{\mathbf{1}^{n-1}}(\mathbf{w})) \in \mathbb{Z}^{2(n-1)} \mid vh \equiv 3w \mod (q, \varPhi_n)\}\\ &= \pi_{(\mathbf{0}^{n-1},\mathbf{1}^{n-1})}\left(\pi_{(\mathbf{1}^{n-1},\mathbf{0}^{n-1})}\left(\mathcal{L}_{\varPhi}\right)\right).\end{aligned}$$

Remark 5.2. Since $\pi_{\mathbf{1}^{n-1}}(\cdot)$ maps into $\frac{1}{n-1} \cdot \mathbb{Z}^{n-1}$, one may want to work in practice with the scaled lattice $(n-1) \cdot \mathcal{L}_{\varPhi,\perp}$ to avoid a non-integral basis.

Remark 5.3. We can easily recover the secret $\mathbf{f}$ from its projection $\pi_{\mathbf{1}^{n-1}}(\mathbf{f})$, by simply brute-forcing the inner product $\langle \mathbf{f}, \mathbf{1}^{n-1} \rangle$ (which is a small integer between $-(n-1)$ and $n-1$) and then obtain $\mathbf{f}$ via Equation (1).

Remark 5.4. Interestingly, Coppersmith and Shamir similarly suggest to project the vectors in their lattice $\mathcal{L}_{\mathsf{CS}}$ orthogonally against $\mathbf{1}^n$ – although for a completely different reason: they showed that any vector $\mathbf{v}$, which is almost parallel to $\mathbf{1}^n$ (i.e., for which $\pi_{\mathbf{1}^n}(\mathbf{v})$ is short), already serves as a valid NTRU private key. If $\mathcal{L}_{\mathsf{CS}}$ contains (besides the rotations of the secret key) additional such vectors, then BKZ has an increased success probability for finding a secret key on the projected variant of the CS lattice. However, in practice, we never encountered such vectors.

As opposed to the (non-projected) cyclotomic lattice, the projected cyclotomic lattice still contains *many* short vectors. In addition, it has a smaller dimension than both $\mathcal{L}_{\mathsf{CS}}$ and $\mathcal{L}_{\varPhi}$. Indeed, it has dimension only $2n-4$: we lose two dimensions by working modulo $\varPhi_n$, and two more dimensions by projecting. (The latter follows from the fact that the q-ary lattice $\mathcal{L}_{\varPhi}$ contains the vectors $q(\mathbf{1}^{n-1}, \mathbf{0}^{n-1})$ and $q(\mathbf{0}^{n-1}, \mathbf{1}^{n-1})$, see also the discussion on almost-parallel hints in Section 4.)

We may therefore hope that attacks using the projected cyclotomic lattice outperform the other two lattices.

It should be noted, however, that $\mathcal{L}_{\Phi,\perp}$ has the smallest determinant out of our three lattices.

Theorem 5.5. *The determinant of $\mathcal{L}_{\Phi,\perp}$ equals*

$$\det \mathcal{L}_{\Phi,\perp} = \frac{q^{n-3}}{n-1}.$$

Proof. Let $d := n-1$. We define a polynomial

$$v := \left((3^{-1} \mod q) \cdot h \cdot \sum_{i=0}^{n-2} X^i \right) \mod \Phi_n. \tag{20}$$

Notice that the coefficient vector $\mathbf{v} \in \mathbb{Z}^d$ of v is the sum over the rows of the matrix $\mathbf{H}_\Phi$ in Equation (14). Hence, by Equation (14) we have the following equivalence for any $s \in \mathbb{Z}$:

$$s(\mathbf{1}^d, \mathbf{0}^d) \in \mathcal{L}_\Phi \iff s\mathbf{v} \in q\mathbb{Z}^d \iff \frac{s}{q}\mathbf{v} \in \mathbb{Z}^d. \tag{21}$$

Using this equivalence we now compute the smallest integer $s > 0$, that satisfies $s(\mathbf{1}^d, \mathbf{0}^d) \in \mathcal{L}_\Phi$.

From Equations (20), (4) and (5), it follows that

$$v \equiv 3^{-1} \cdot h \cdot \sum_{i=0}^{n-2} X^i \equiv g \cdot f^{-1} \cdot \sum_{i=0}^{n-2} X^i \mod (q, \Phi_n).$$

By Lemma 3.1, both g and $\sum_{i=0}^{n-2} X^i$ are invertible in the ring $\mathbb{Z}_q[X]/(\Phi_n)$.[4] As f^{-1} is obviously also invertible, v is invertible as well. In particular, it follows that at least one coefficient of v is odd. (Since q is a power of 2, polynomials with only even coefficients are not invertible in $\mathbb{Z}_q[X]/(\Phi_n)$.) Combing this observation with Equation (21), it follows that the smallest $s > 0$, satisfying $s(\mathbf{1}^d, \mathbf{0}^d) \in \mathcal{L}_\Phi$, is $s = q$.

This shows that $q(\mathbf{1}^d, \mathbf{0}^d)$ is primitive with respect to $\mathcal{L}_\Phi$. Together with Lemma 4.3 this yields

$$\det \left(\pi_{(\mathbf{1}^d, \mathbf{0}^d)}(\mathcal{L}_\Phi) \right) = \frac{\det \mathcal{L}_\Phi}{\|q(\mathbf{1}^d, \mathbf{0}^d)\|} = \frac{q^{n-2}}{\sqrt{d}}. \tag{22}$$

Let us now compute the smallest integer $t > 0$, satisfying $t(\mathbf{0}^d, \mathbf{1}^d) \in \pi_{(\mathbf{1}^d, \mathbf{0}^d)}(\mathcal{L}_\Phi)$.

From Equation (14) it follows that $\pi_{(\mathbf{1}^d, \mathbf{0}^d)}(\mathcal{L}_\Phi)$ is generated by the following matrix

$$\mathbf{B}_\pi := \begin{pmatrix} \pi_{\mathbf{1}^d}(\mathbf{I}_d) & \mathbf{H}_\Phi \\ \mathbf{0} & q\mathbf{I}_d \end{pmatrix}, \tag{23}$$

[4] Even though g is not a ternary polynomial in NTRU-HRSS, Lemma 3.1 still implies that g is invertible, since g is the product of two ternary (and therefore invertible) polynomials, see Equation (8).

where $\pi_{\mathbf{1}^d}$ is applied row-wise to $\mathbf{I}_d$. Let $(\mathbf{w}_1, \mathbf{w}_2) \in \mathbb{Z}^{2d}$, such that

$$(\mathbf{w}_1, \mathbf{w}_2) \cdot \mathbf{B}_\pi = t(\mathbf{0}^d, \mathbf{1}^d) \tag{24}$$

for the smallest possible integer $t > 0$. Using Equation (23), we conclude that

$$\mathbf{w}_1 \in \ker \pi_{\mathbf{1}^d} \cap \mathbb{Z}^d = \mathbb{Z} \cdot \mathbf{1}^d.$$

Hence, there exists $m \in \mathbb{Z}$, such that $\mathbf{w}_1 = m \cdot \mathbf{1}^d$. This implies that $\mathbf{w}_1$ is the coefficient vector of $m \cdot \sum_{i=0}^{n-2} X^i$. Using Equation (24) it follows that

$$\left(m \cdot \sum_{i=0}^{n-2} X^i\right) \cdot \left(3^{-1} \cdot h\right) \equiv t \cdot \sum_{i=0}^{n-2} X^i \mod (q, \Phi_n).$$

By Lemma 3.1, the ternary polynomial $\sum_{i=0}^{n-2} X^i$ is invertible, so we may divide it from the above congruence and obtain

$$m \cdot 3^{-1} \cdot h \equiv t \mod (q, \Phi_n). \tag{25}$$

Multiplying the polynomial $3^{-1} \cdot h$ by an integer m can result in another integer t, if and only if t is congruent to 0 modulo q. Hence, the smallest $t > 0$, for which Equation (25) can hold is $t = q$.

By definition of m, this implies that the smallest $t > 0$, satisfying $t(\mathbf{0}^d, \mathbf{1}^d) \in \pi_{(\mathbf{1}^d,\mathbf{0}^d)}(\mathcal{L}_\Phi)$, is $t = q$. Hence, $q(\mathbf{0}^d, \mathbf{1}^d)$ is primitive with respect to $\pi_{(\mathbf{1}^d,\mathbf{0}^d)}(\mathcal{L}_\Phi)$.

Now applying Lemma 4.3 and using Equation (22), we finish the proof:

$$\begin{aligned}\det \mathcal{L}_{\Phi,\perp} &= \det\left(\pi_{(\mathbf{0}^d,\mathbf{1}^d)}\left(\pi_{(\mathbf{1}^d,\mathbf{0}^d)}\left(\mathcal{L}_\Phi\right)\right)\right) \\ &= \frac{\det\left(\pi_{(\mathbf{1}^d,\mathbf{0}^d)}(\mathcal{L}_\Phi)\right)}{\|q(\mathbf{0}^d, \mathbf{1}^d)\|} = \frac{q^{n-3}}{n-1}.\end{aligned}$$

□

Recall that by Equation (2), the required BKZ blocksize does not directly depend on the determinant of the lattice, but on its *root-determinant*. Since we have

$$(\det \mathcal{L}_{\mathsf{CS}})^{1/(2n)} = \sqrt{q},$$

and by Theorem 5.5

$$(\det \mathcal{L}_{\Phi,\perp})^{1/(2(n-2))} = \frac{1}{((n-1)q)^{1/(2n-4)}} \sqrt{q},$$

the root-determinants only differ by a factor $\frac{1}{((n-1)q)^{1/(2n-4)}}$, which rapidly converges to 1. Thus, the decrease in determinant should not significantly effect the required blocksize, and instead should be outweighed by the decrease in secret's norm and lattice dimension. Our experimental results in Section 6 confirm that this is the case.

5.3 Further Improvement by Exploiting Design Choices

For correctness of decryption, it is necessary in both NTRU-HPS and NTRU-HRSS that 1 is a root of g. HPS ensures this property by simply distributing 1's and -1's evenly among the coefficients of g, see Equation (6). HRSS, on the other hand, defines g as a product of $X-1$ and a ternary polynomial, see Equation (8). We can exploit these properties by incorporating them into our lattices.

NTRU-HRSS. Since the secret polynomial g is the product of $X-1$ and a ternary polynomial, the coefficient vector $\frac{g}{X-1}$ is significantly shorter than the coefficient vector of g. Instead of searching for a short vector $(\mathbf{v}, \mathbf{w})$ with

$$vh \equiv 3w \mod (q, \Phi_n),$$

we should therefore rather search for $(\mathbf{v}, \mathbf{w}')$ with

$$vh \equiv 3(X-1)w' \mod (q, \Phi_n),$$

To do so, we simply replace the matrix $\mathbf{H}_\Phi$ in Equation (14) by a matrix $\mathbf{H}'_\Phi$, where for $i = 0, \dots, n-2$, the $(i+1)$-st row of $\mathbf{H}'_\Phi$ is defined as the coefficient vector of

$$3^{-1}(X-1)^{-1}X^i h \mod (q, \Phi_n).$$

Notice that by Lemma 3.1, $X-1$ is indeed invertible in $\mathbb{Z}_q[X]/(\Phi_n)$.

Interestingly, we cannot as easily apply this trick when working modulo $X^n - 1$, since $X-1$ is not invertible modulo X^n-1. In fact, we fail to see how to explicitly compute a basis for the lattice

$$\mathcal{L}_{\mathsf{HRSS}} := \{(\mathbf{v}, \mathbf{w}) \in \mathbb{Z}^{2n} \mid vh \equiv 3(X-1)w \mod (q, X^n - 1)\}. \tag{26}$$

NTRU-HPS. Even though g is also divisible by $X-1$ in NTRU-HPS, we should not divide that factor out here, because the resulting polynomial would not have as small coefficients. We can nevertheless still incorporate the fact $g(1) = 0$ to our lattice, by instead interpreting it as a perfect hint.

Geometrically, the equation $g(1) = 0$ is equivalent to the fact that the coefficient vector of g is orthogonal to the all-one vector. Hence, instead of working directly with the CS lattice or the projected cyclotomic lattice, we may first intersect them with $(\mathbf{0}^n, \mathbf{1}^n)^\perp$ or $(\mathbf{0}^{n-1}, \mathbf{1}^{n-1})^\perp$, respectively, i.e., we may work with the following lattices

$$\mathcal{L}_{\mathsf{CS},\cap} := \mathcal{L}_{\mathsf{CS}} \cap (\mathbf{0}^n, \mathbf{1}^n)^\perp,$$
$$\mathcal{L}_{\Phi,\perp,\cap} := \pi_{(\mathbf{1}^{n-1},\mathbf{0}^{n-1})}\left(\pi_{(\mathbf{0}^{n-1},\mathbf{1}^{n-1})}\left(\mathcal{L}_\Phi \cap (\mathbf{0}^{n-1}, \mathbf{1}^{n-1})^\perp\right)\right).$$

We would like to point out that Dachman-Soled, Ducas, Gong and Rossi [DDGR20, Section 6.3] also suggest to incorporate the fact $g(1) = 0$ as a perfect hint for NTRU-HPS. Worth noting, they claim that, according to Lemma 4.1 (respectively Lemma 12 in their work), this increases the determinant of the CS lattice by a factor of $\sqrt{n}$. This claim, however, is not correct. It would be correct if the vector $(\mathbf{0}^n, \mathbf{1}^n)$ was primitive with respect to the dual of $\mathcal{L}_{\mathsf{CS}}$. This is, however, not the case:

Theorem 5.6. *The vector* $(\mathbf{0}^n, \mathbf{1}^n)$ *is not primitive with respect to the dual of* $\mathcal{L}_{\mathsf{CS}}$*, but* $\frac{1}{q}(\mathbf{0}^n, \mathbf{1}^n)$ *is.*

Proof. Since $g(1) = 0$, it follows from Equation (4) that $h(1) \equiv 0 \mod q$. This implies that the sum over all coefficients of h is a multiple of q. Hence, for every row $\mathbf{H}_i$ of $\mathbf{H}$, as defined in Equation (13), the inner product between $\frac{1}{q}\mathbf{1}^n$ and $\mathbf{H}_i$ is an integer. Clearly, the inner product between $\frac{1}{q}\mathbf{1}^n$ and a row of $q\mathbf{I}_n$ is also an integer (namely 1).

Combing these two observations with Equation (13), it follows that for every vector $\mathbf{v} \in \mathcal{L}_{\mathsf{CS}}$ the inner product $\langle \frac{1}{q}(\mathbf{0}^n, \mathbf{1}^n), \mathbf{v}\rangle$ is an integer. Hence, $\frac{1}{q}(\mathbf{0}^n, \mathbf{1}^n)$ lies in the dual of $\mathcal{L}_{\mathsf{CS}}$.

To finish the proof, it remains to show that for any $k \geq 2$, the vector $\frac{1}{kq}(\mathbf{0}^n, \mathbf{1}^n)$ is not included in $\mathcal{L}^*_{\mathsf{CS}}$. This easily follows from the fact that for any such k, the inner product between $\frac{1}{kq}(\mathbf{0}^n, \mathbf{1}^n)$ and the q-vector $(\mathbf{0}^n, q, \mathbf{0}^{n-1}) \in \mathcal{L}_{\mathsf{CS}}$ equals $\frac{1}{k}$ and thus is non-integral. □

According to Lemma 4.1 and Theorem 5.6, the hint $g(1) = 0$ does not increase the determinant by a factor of $\sqrt{n}$, but decreases it by a factor of $\sqrt{n}/q$.

6 Experimental Results for HRSS and HPS

We implemented all the lattices described in Section 5. The source code is available at https://github.com/ElenaKirshanova/ntru_with_sieving.

We provide an interface to generate NTRU-HPS and NTRU-HRSS keys as specified in the documentation [CDH⁺19]. Our interface also allows to input explicit public parameters n, q, h, instead of generating random instances, e.g. in order to solve the challenges from [Incb].

Our implementation supports the following types of lattices: the Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$, the cyclotomic lattice $\mathcal{L}_{\Phi}$, the projected cyclotomic lattice $\mathcal{L}_{\Phi,\perp}$ as well as the lattices $\mathcal{L}_{\mathsf{CS},\cap}$ and $\mathcal{L}_{\Phi,\perp,\cap}$. Upon receiving the type of the NTRU lattice together with the parameters n, q (and optionally h), our implementation constructs the corresponding basis and starts lattice reduction.

We use progressive BKZ [AWHT16] that internally calls either enumeration from the FPyLLL library [dt21a] (with the default pruning strategies [GNR10] for enumeration), or sieving from the G6K library [ADH⁺19]. Choosing which SVP oracle to use is left to the user. In our experiments, for BKZ blocksizes higher than 65, we use sieving. For smaller blocksizes we run enumeration.[5]

For each BKZ tour, we check either of the two events: *dense sublattice discovery* (DSD) or *secret key recovery* (SKR). In case of DSD, we extract the dense

[5] In [ADH⁺19], the crossover point between enumeration and sieving was observed at dimension 70. However, we gain additional speed-up from parallelized sieving.

sublattice, which is half of the dimension of the original lattice, and continue with progressive BKZ on this smaller lattice until we find the NTRU secret key.

In all our experiments we use an AMD EPYC 7763 with 1 TB of RAM, as well as an AMD EPYC 7742 processor with 2 TB of RAM. Each EPYC is equipped with 128 physical cores that with parallelization give 256 threads. This number of cores was mostly used to run multiple parallel experiments.

6.1 NTRU-HRSS

Unlike in most other NTRU variants, the parameter q cannot be chosen freely in NTRU-HRSS. Instead, it is fixed to $q = 2^{\lceil 7/2+\log_2(n)\rceil}$, as specified in Equation (7). For medium sized values of n, this formula sets q to a value significantly larger than n. For instance, for $91 \leq n \leq 181$, it sets $q = 2048$. (In contrast, for NTRU-HPS such a large q is recommended for $n \in \{509, 677\}$.) As a consequence, NTRU-HRSS parameters with medium sized n lie in the overstretched regime. Indeed, according to the NTRU Fatigue estimator, all NTRU-HRSS parameters with $n < 261$ are overstretched.

We would like to stress that all HRSS parameters, that we can currently attack in a reasonable amount of time, are therefore overstretched. The recommended parameters $n = 701$ and $q = 8192$, on the other hand, are not overstretched.

We ran experiments from $n = 101$ up to $n = 211$ for NTRU-HRSS. Note that only $n = 101, 211$ are elements of $\mathcal{N}$, as defined in Section 3, but we are not exploiting any structure for speeding up lattice reduction for $n \notin \mathcal{N}$.

As expected, in 100% of our experiments the DSD event occurred, confirming that NTRU-HRSS with medium sized n indeed is overstretched. Once the DSD event was detected at blocksize β, the SKR event followed within the next 5 blocksizes, i.e., at blocksize at most $\beta + 5$. In larger dimensions $n \geq 151$, the SKR event usually happened even in the next progressive BKZ call. In some experiments, DSD and SKR events happened at the exact same block size.[6]

Observed Speedup from $\mathcal{L}_{\Phi,\perp}$. In our experiments, we tried all different types of lattices that our implementation supports. Out of all lattices, the Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$ performs worst, whereas the projected cyclotomic lattice $\mathcal{L}_{\Phi,\perp}$ performs best. In the left half of Figure 1, we plot the required average blocksize β for DSD on $\mathcal{L}_{\mathsf{CS}}$ and on $\mathcal{L}_{\Phi,\perp}$ for $101 \leq n \leq 171$. The exact numbers are given in Table 2. As the figure and table show, $\mathcal{L}_{\Phi,\perp}$ performs significantly better than $\mathcal{L}_{\mathsf{CS}}$.

Changing to the cyclotomic ring and using almost-parallel hints therefore is indeed beneficial for lattice reduction algorithms. As expected, the benefits are not outweighed by decreasing the determinant from q^n to $q^{n-3}/(n-1)$, see Theorem 5.5.

[6] When DSD and SKR happen at the same blocksize, we are still in the overstretched regime. We are in the non-overstretched regime only if SKR happens *before* DSD.

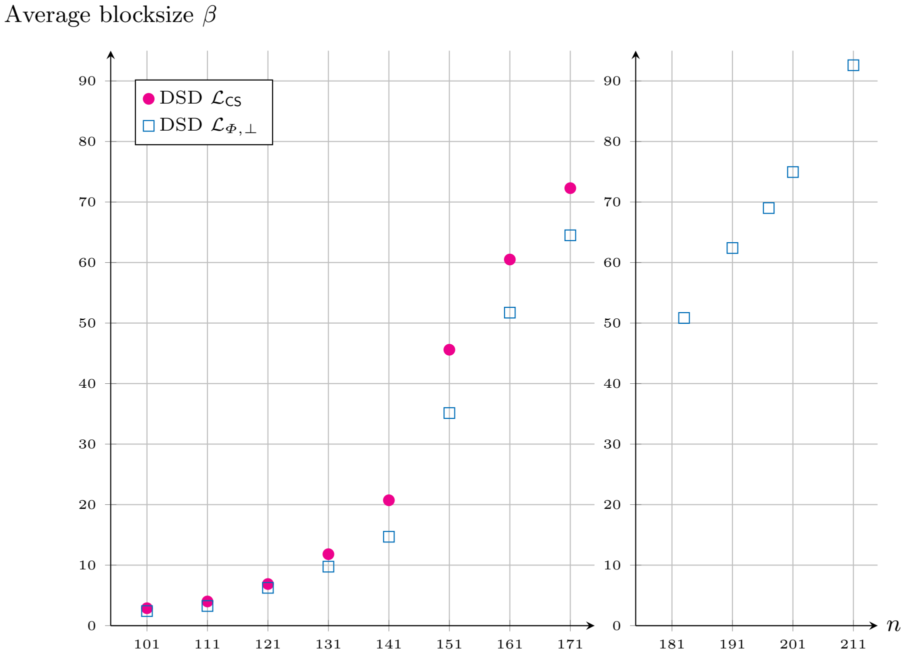

Fig. 1. Comparison between $\mathcal{L}_{\mathsf{CS}}$ and $\mathcal{L}_{\Phi,\perp}$ in the overstretched regime of NTRU-HRSS with $q = 2048$ (left) and $q = 4096$ (right). Averaged over 32 keys each for every $n \leq 171$. For $n = 191, 201$, the average is taken over 20 experiments. For $n = 211$, the blocksize is averaged over 5 experiments.

For $n \geq 183$, we ran experiments only on the superior lattice $\mathcal{L}_{\Phi,\perp}$. The results are shown in the right half of Figure 1. Comparing both halves of Figure 1, the reader may notice that for $n = 161$ and $n = 183$ we require roughly the same blocksize $\beta \approx 51$. This is due to the increase in q, caused by Equation (7): for $n \geq 182$, we switch from $q = 2048$ to $q = 4096$.

n	101	111	121	131	141	151	161	171
$\beta_{\Phi,\perp}$	2.4	3.3	6.3	9.8	14.7	35.1	51.7	64.5
β_{CS}	2.9	4.0	6.9	11.8	20.7	45.6	60.5	72.3

Table 2. Average required blocksizes for NTRU-HRSS, as per Figure 1

6.2 NTRU-HPS

In contrast to our NTRU-HRSS experiments, we used for NTRU-HPS a significantly smaller modulus $q = 512$ to ensure that we are far off from the overstretched regime. As one might expect, this decrease in q results in significantly larger required BKZ blocksizes. For instance, with NTRU-HPS we require for $n = 131$ a blocksize of $\beta \approx 57$ for SKR, whereas with NTRU-HRSS we require only $\beta \approx 10$ for DSD. Therefore, we cannot provide results for n as large as in NTRU-HRSS ($n = 211$), but only up to $n = 171$.

Nevertheless, our NTRU-HPS computations can still be considered new records in the field of practical NTRU cryptanalysis: The former NTRU record computation by Ducas and Nguyen [Inca] were in similar dimension $n = 173$, but with a larger modulus of $q = 1024$, and therefore (presumably) required smaller blocksizes than our computations. We go up to blocksizes $\beta > 100$, a regime that to the best of our knowledge has not been reached in practical cryptanalysis so far.

As with NTRU-HRSS, we also ran our experiments on NTRU-HPS with all different types of lattices, that are available in our implementation. The Coppersmith-Shamir lattice $\mathcal{L}_{\mathsf{CS}}$ again performed worst, whereas the projected cyclotomic lattice $\mathcal{L}_{\Phi,\perp,\cap}$ (with additional integrated hints, see Section 5.3) performed best.

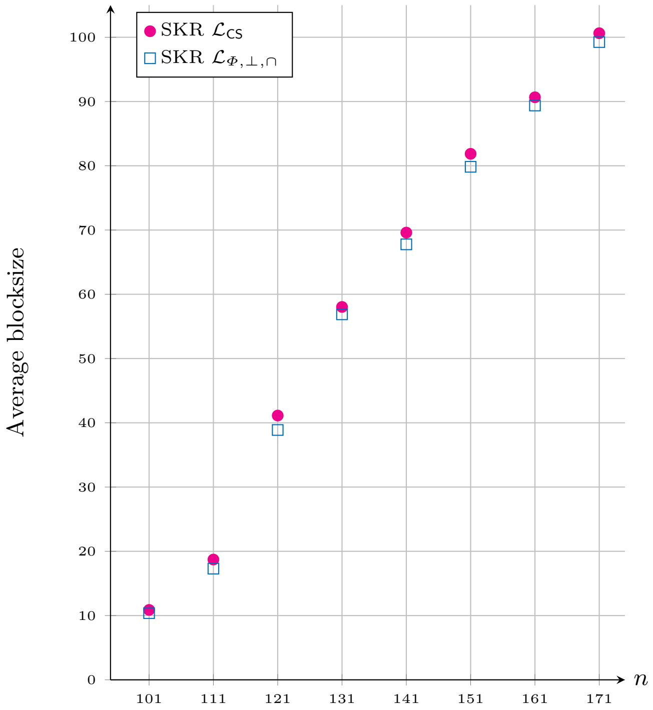

Fig. 2. Required blocksize for the secret key recovery on NTRU-HPS with $q = 512$. Averaged over 32 keys each for every n.

Observed Speedup from $\mathcal{L}_{\Phi,\perp,\cap}$. In Figure 2, we plot the average required β for $\mathcal{L}_{\mathsf{CS}}$ and $\mathcal{L}_{\Phi,\perp,\cap}$. In contrast to NTRU-HRSS, the gap between the two lattices here is not as large. (See Section 6.3 for an explanation.)

n	101	111	121	131	141	151	161	171
β_{CS}	10.9	18.7	41.1	58.0	69.6	81.9	90.7	100.6
$\beta_{\Phi,\perp,\cap}$	10.4	17.3	38.9	56.8	67.8	79.9	89.4	99.2

Table 3. Average required blocksizes for NTRU-HPS, as per Figure 2

In Table 3, we provide all averaged blocksizes β_{CS} and $\beta_{\Phi,\perp,\cap}$ required for SKR in $\mathcal{L}_{\mathsf{CS}}$ and $\mathcal{L}_{\Phi,\perp,\cap}$, respectively. While the differences in blocksizes may seem rather small, we note that the difference in runtimes is still significant in practice. For instance, in our $n = 171$ experiment, $\mathcal{L}_{\mathsf{CS}}$ required on average 172 core days, whereas $\mathcal{L}_{\Phi,\perp,\cap}$ took on average only 83 core days.

As expected, and similarly as in NTRU-HRSS, the benefits of reducing the dimension thus outweigh the disadvantage of a decreased determinant also in NTRU-HPS.

6.3 Comparison Between HRSS and HPS, and Implications

Let us now explain why the gap between β's is for HRSS significantly larger than it is for HPS.

Recall that we decrease the dimension in HRSS by 4 (by switching to the projected cyclotomic lattice) and in HPS by 5 (by additionally integrating one perfect hint). In both variants, we decrease the root determinant only by a negligible amount. With respect to dimension and determinant, both lattices are thus very similar.

The main difference between our lattices for HRSS and HPS is that $\mathcal{L}_{\mathsf{CS}}$ contains for HRSS the polynomial g that is a multiple of $(X - 1)$ and ternary polynomial, see Equation (8). As proposed in Section 5.3, in the construction of $\mathcal{L}_{\Phi,\perp}$ we divide out $(X - 1)$. This reduces g's norm by roughly a $\sqrt{2}$-factor. Hence, for HRSS we not only decrease the dimension of the lattice, but also decrease the norm of the shortest vectors – which results in a larger gap between β's.

We note that in the HRSS specification [CDH$^+$19] the authors analyzed the lattice $\mathcal{L}_{\mathsf{HRSS}}$ as defined in Equation (26). Although [CDH$^+$19] does not explicitly provide a basis for $\mathcal{L}_{\mathsf{HRSS}}$ – and constructing one might actually be hard – $\mathcal{L}_{\mathsf{HRSS}}$ was nevertheless used in [CDH$^+$19] to conservatively estimate HRSS security. (In other words, the authors of [CDH$^+$19] already anticipated our improvement in lattice basis construction.) Since the $(X - 1)$ factor is already divided out in $\mathcal{L}_{\mathsf{HRSS}}$, the gap in β's thus do not imply a security loss for HRSS.

7 New NTRU Record: $n = 181$

7.1 Choosing a Lattice for the NTRU Challenges

Due to the different key format in the NTRU challenges, the lattices introduced in Section 5 are not the best choice for attacking the challenges. While in NTRU-HPS and NTRU-HRSS the keys satisfy the equation

$$fh \equiv 3g,$$

in the NTRU challenges they satisfy

$$(1 + 3f)h \equiv g. \tag{27}$$

Hence, for the challenges, it is likely not the optimal strategy to search for a short vector $(\mathbf{v}, \mathbf{w})$ satisfying

$$vh \equiv w,$$

since such a vector would be significantly longer than the coefficient vector $(\mathbf{f}, \mathbf{g})$.

As a better strategy, Ducas and Nguyen interpreted Equation (27) in their record computations as an instance of the *bounded distance decoding problem (BDD)*. They constructed a variant of the CS lattice, namely

$$\mathcal{L}_{\mathsf{DN}} := \{(\mathbf{v}, \mathbf{w}) \in \mathbb{Z}^{2n} \mid 3vh \equiv w \mod (q, X^n - 1)\}, \tag{28}$$

and then searched for a lattice vector $(\mathbf{v}, \mathbf{w}) \in \mathcal{L}_{\mathsf{DN}}$, close to the (non-lattice) vector $(\mathbf{0}^n, \mathbf{h})$. With this strategy one likely finds the vector $(\mathbf{f}, \mathbf{g} - \mathbf{h}) \in \mathcal{L}_{\mathsf{DN}}$, since it is close to $(\mathbf{0}^n, \mathbf{h})$.

We choose to follow a different strategy based on the framework of lattice reduction with hints. Instead of interpreting Equation (27) as a BDD instance, we choose to interpret it as an almost-parallel hint. The equation implies that $\mathcal{L}_{\mathsf{DN}}$ contains with $(\mathbf{f}, \mathbf{g} - \mathbf{h})$ a vector almost parallel to $(\mathbf{0}^n, \mathbf{h})$. This suggests to project $\mathcal{L}_{\mathsf{DN}}$ orthogonally against $(\mathbf{0}^n, \mathbf{h})$ and then search for

$$\pi_{(\mathbf{0}^n, \mathbf{h})}\big((\mathbf{f}, \mathbf{g} - \mathbf{h})\big) = (\mathbf{f}, \pi_{\mathbf{h}}(\mathbf{g})). \tag{29}$$

as a shortest vector.

We can further improve this lattice, by additionally incorporating a perfect hint. From Equation (11) we know that f satisfies $f(1) = 0$, since f is composed out of three polynomials f_1, f_2, f_3, which all satisfy $f_i(1) = 0$ (see Equation (10)). Hence, we have $\langle \mathbf{f}, \mathbf{1}^n \rangle = 0$. Thus, to improve the attack we may intersect $\mathcal{L}_{\mathsf{DN}}$ with $(\mathbf{1}^n, \mathbf{0}^n)^\perp$.

Remark 7.1. Similarly as with the projected cyclotomic lattice, one may want to work with the scaled lattice $\|\mathbf{h}\|^2 \cdot \pi_{(\mathbf{0}, \mathbf{h})}\big((\mathcal{L}_{\mathsf{DN}} \cap (\mathbf{1}^n, \mathbf{0}^n)^\perp)\big)$ in practice to avoid a non-integral basis (see also Remark 5.2).

Remark 7.2. Here we do not have to worry, whether we can efficiently invert the projection against $(\mathbf{0}, \mathbf{h})$, since the left half of the secret still contains the (non-projected) vector $\mathbf{f}$, see Equation (29).

On Further Possible Improvements. As in Section 3.1, we could theoretically further improve our lattice, by working over the cyclotomic ring $\mathbb{Z}_q[X]/(\Phi_n)$ and incorporating the two almost parallel hints of $(\mathbf{1}^{n-1}, \mathbf{0}^{n-1})$ and $(\mathbf{0}^{n-1}, \mathbf{1}^{n-1})$. However, in that case we would already project our lattice against three vectors in total. This would make the denominators of the coefficients of our lattice vectors very large (or equivalently it would require scaling the lattice by a large factor). To avoid issues of numerical stability in practice, we therefore choose to not include these improvements.

One might ask, whether we can use special properties of g to further improve our lattice (since we have only used the structure of f and the almost-parallel hint of $(\mathbf{0}, \mathbf{h})$ so far).

From Equation (12) it follows that $g(1) = 1$ or, equivalently,

$$\langle \mathbf{g}, \mathbf{1}^n \rangle = 1.$$

Theoretically, we could also incorporate such a non-homogeneous type of perfect hint into our lattice by using the framework of [DDGR20]. However, for that we would first have to embed our NTRU problem into a non-homogeneous LWE problem, see [DDGR20, Section 4.1]. This would increase the lattice dimension by one and therefore negate the effect of introducing the perfect hint $\langle \mathbf{f}, \mathbf{1}^n \rangle = 0$.

As an alternative, one may try to obtain a short vector hint from the structure of g: Applying $f(1) = 0$ and $g(1) = 1$ to Equation (27), it follows that

$$h(1) = 1 \mod q. \tag{30}$$

Combining this with the fact that for every polynomial $p \in \mathbb{Z}_q[X]/(X^n - 1)$ it holds that[7]

$$(X^{n-1} + \ldots + X + 1) \cdot p \equiv \left(X^{n-1} + \ldots + X + 1\right) p(1), \tag{31}$$

we obtain

$$(X^{n-1} + X^{n-2} \ldots + X + 1) \cdot h \equiv X^{n-1} + X^{n-2} \ldots + X + 1.$$

Hence, by Equation (28), the lattice $\mathcal{L}_{\mathsf{DN}}$ contains the short vector $(\mathbf{1}^n, \mathbf{3}^n)$.

One might be tempted to incorporate this fact as a short vector hint. Unfortunately, we have, however, already removed the vector $(\mathbf{1}^n, \mathbf{3}^n)$ from our lattice, because we intersected it with $(\mathbf{1}^n, \mathbf{0}^n)^\perp$.

Interestingly, both the perfect hint $\langle \mathbf{f}, \mathbf{1}^n \rangle = 0$ and the short vector hint $(\mathbf{1}^n, \mathbf{3}^n)$ therefore decrease the dimension by one by *removing* $(\mathbf{1}^n, \mathbf{3}^n)$ from the lattice. The following theorem shows, however, that the perfect hint is superior:

Theorem 7.3. *The determinants of the lattices* $\pi_{(\mathbf{1}^n, \mathbf{3}^n)}(\mathcal{L}_{\mathsf{DN}})$ *and* $\mathcal{L}_{\mathsf{DN}} \cap (\mathbf{1}^n, \mathbf{0}^n)^\perp$ *are given by*

$$\det\left(\mathcal{L}_{\mathsf{DN}} \cap (\mathbf{1}^n, \mathbf{0}^n)^\perp\right) = \sqrt{n} q^n, \tag{32}$$

and

$$\det\left(\pi_{(\mathbf{1}^n, \mathbf{3}^n)}(\mathcal{L}_{\mathsf{DN}})\right) = \frac{q^n}{\sqrt{10n}}. \tag{33}$$

[7] Equation (31) easily follows from the Chinese Remainder Theorem.

Proof. Since $\mathcal{L}_{\mathsf{DN}} \subset \mathbb{Z}^{2n}$ is an integer lattice, the vector $(\mathbf{1}^n, \mathbf{3}^n) \in \mathcal{L}_{\mathsf{DN}}$ clearly is primitive with respect to $\mathcal{L}_{\mathsf{DN}}$. Hence, Equation (33) immediately follows from Lemma 4.2.

To prove Equation (32), it suffices to show that $(\mathbf{1}^n, \mathbf{0}^n)$ is primitive with respect to the dual $\mathcal{L}^*_{\mathsf{DN}}$, see Lemma 4.1. This, in turn, easily follows from the fact that for every integer $k \geq 2$ the inner product between $\frac{1}{k}(\mathbf{1}^n, \mathbf{0}^n)$ and the lattice vector $(1, \mathbf{0}^{n-1}, 3\mathbf{h}) \in \mathcal{L}_{\mathsf{DN}}$ (see Equation (28)) equals $\frac{1}{k}$ and thus is not integral. □

7.2 Record Computation Details

The idea of incorporating almost-parallel hints enables us to establish a new record for the NTRU challenges from [Incb]. The former record holders are Ducas and Nguyen [Inca] who managed to solve NTRU with $n = 173$. We went up one challenge further and solved NTRU for $n = 181$ with $q = 1024$. These parameters do not lie in the overstretched regime. To solve the challenge, we implemented the approach described above. That is, we run BKZ on the lattice $\mathcal{L}_{\mathsf{DN}}$ from Equation (28) intersected with $(\mathbf{1}^n, \mathbf{0}^n)^{\perp}$ and projected orthogonally against $(\mathbf{0}^n, \mathbf{h})$. The shortest vector of the form (f, g) was found at blocksize $\beta = 109$ after 20 core years of computations. The solution is posted at https://github.com/ElenaKirshanova/ntru_with_sieving.

References

ABB+20. Nicolas Aragon, Paulo S. L. M. Barreto, Slim Bettaieb, Loïc Bidoux, Oliver Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, Carols Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Jan Richter-Brockmann, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. BIKE: bit flipping key encapsulation, 2020. Available at https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf.

ABC+20. Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varum Maram, Ingo von Maurich, Rafael Misocki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic mceliece: conservative code-based cryptography, 2020. Available at https://classic.mceliece.org/nist/mceliece-20201010.pdf.

ABF+20. Martin R. Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. Fasterenumeration-basedlatticereduction: Roothermitefactor $k^{1/2k}$ time $k^{k/8+o(k)}$. In *Advances in Cryptology – CRYPTO 2020*, pages 186–212, 2020.

AD21. Martin R. Albrecht and Léo Ducas. *Lattice Attacks on NTRU and LWE: A History of Refinements*, page 1540. 2021.

ADH+19. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In *Advances in Cryptology - EUROCRYPT 2019*, pages 717–746, 2019.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium*, pages 327–343, 2016.

AKS01. Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. STOC '01, page 601610, 2001.

AWHT16. Yoshinori Aono, Yuntao Wang, Takuya Hayashi, and Tsuyoshi Takagi. Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In *Advances in Cryptology - EUROCRYPT 2016*, pages 789–819, 2016.

BDK+18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE EuroS&P*, pages 353–367, 2018.

CDH+19. Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Tsunekazu Saito, Peter Schwabe, William Whyte, Keita Xagawa, Takashi Yamakawa, and Zhenfei Zhang. PQC round-3 candidate: NTRU. technical report, 2019. Available at https://ntru.org/f/ntru-20190330.pdf.

CN11. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *Advances in Cryptology – ASIACRYPT 2011*, pages 1–20, 2011.

CS97. Don Coppersmith and Adi Shamir. Lattice attacks on NTRU. In *Advances in Cryptology - EUROCRYPT '97*, volume 1233, pages 52–61, 1997.

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In *Advances in Cryptology - CRYPTO 2020*, pages 329–358, 2020.

DSvW21. Léo Ducas, Marc Stevens, and Wessel van Woerden. Advanced lattice sieving on gpus, with tensor cores. In *Advances in Cryptology – EUROCRYPT 2021*, pages 249–279, 2021.

dt21a. The FPLLL development team. fpylll, a Python wraper for the fplll lattice reduction library, Version: 0.5.7. Available at https://github.com/fplll/fpylll, 2021.

dt21b. The G6K development team. The general sieve kernel (g6k). Available at https://github.com/fplll/g6k, 2021.

Duc18. Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In *Advances in Cryptology - EUROCRYPT 2018*, pages 125–145, 2018.

DvW21. Léo Ducas and Wessel P. J. van Woerden. NTRU fatigue: How stretched is overstretched? In *Advances in Cryptology - ASIACRYPT 2021*, Lecture Notes in Computer Science, pages 3–32, 2021.

FHK+18. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhan. FALCON: Fast-fourier lattice-based compact signatures over NTRU. Available at https://www.di.ens.fr/~prest/Publications/falcon.pdf, 2018.

Gen01. Craig Gentry. Key recovery and message attacks on NTRU-composite. In *Advances in Cryptology - EUROCRYPT 2001*, pages 182–194, 2001.

GNR10. Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology - EUROCRYPT 2010*, pages 257–278, 2010.

How07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Advances in Cryptology - CRYPTO 2007*, pages 150–169, 2007.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *ANTS-III*, pages 267–288, 1998.

HRSS17. Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In *Cryptographic Hardware and Embedded Systems - CHES 2017*, pages 232–252, 2017.

Inca. NTRU Securty Innovation Inc. NTRU challenge - answers. Available at https://web.archive.org/web/20151229220714/https://www.securityinnovation.com/uploads/ntru-challenge-parameter-sets-and-public-keys-answers.pdf.

Incb. NTRU Securty Innovation Inc. NTRU challenge parameter sets and public keys. Available at https://web.archive.org/web/20160310141551/https://www.securityinnovation.com/uploads/ntru-challenge-parameter-sets-and-public-keys-new.pdf.

Kan83. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. STOC '83, 1983.

KF17. Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In *Advances in Cryptology - EUROCRYPT 2017*, pages 3–26, 2017.

LLL82. Arjen K Lenstra, Hendrik W Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(ARTICLE):515–534, 1982.

LN13. Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In *Topics in Cryptology - CT-RSA 2013*, pages 293–309, 2013.

MAB$^+$21. Carols Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Oliver Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, and Gilles Zémor. Hamming quasi-cyclic (hqc), 2021. Available at https://pqc-hqc.org/doc/hqc-specification_2021-06-06.pdf.

MS01. Alexander May and Joseph H. Silverman. Dimension reduction methods for convolution modular lattices. In *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, pages 110–125, 2001.

MV10. Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *SODA '10*, page 14681480, 2010.

NV08. Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2):181–207, 2008.

Sch87. Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

Sch03. Claus-Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS 2003*, volume 2607, pages 145–156, 2003.