Finding Dense Submodules with Algebraic Lattice Reduction

Alexander Karenin[1,2] and Elena Kirshanova[1,2]

[1] Technology Innovation Institute, Abu Dhabi, UAE.
{alexander.karenin, elena.kirshanova}@tii.ae
[2] I.Kant Baltic Federal University, Kaliningrad, Russia

Abstract. We prove an algebraic analogue of Pataki-Tural lemma (Pataki-Tural, *arXiv:0804.4014, 2008*) – the main tool in analysing the so-called *overstretched* regime of NTRU. Our result generalizes this lemma from Euclidean lattices to modules over any number field enabling us to look at NTRU as rank-2 module over cyclotomic number fields with a rank-1 dense submodule generated by the NTRU secret key.

For Euclidean lattices, this *overstretched* regime occurs for large moduli q and enables to detect a dense sublattice in NTRU lattices leading to faster NTRU key recovery. We formulate an algebraic version of this event, the so-called Dense Submodule Discovery (DSD) event, and heuristically predict under which conditions this event happens. For that, we formulate an algebraic version of the Geometric Series Assumption – an heuristic tool that describes the behaviour of algebraic lattice reduction algorithms. We verify this assumption by implementing an algebraic LLL – an analog of classical LLL lattice reduction that operates on the module level. Our experiments verify the introduced heuristic, enabling us to predict the algebraic DSD event.

Keywords: NTRU · Cryptoanalysis · LLL algorithm · Module lattices

1 Introduction

Modern lattice based cryptographic constructions, including the recent standards [17,7,15], rely on hard problems on module lattices, i.e., lattices that are modules over the rings of integers of number fields.

Let K be a number field and $\mathcal{O}_K$ be its ring of integers. In this work we focus on cyclotomic number fields of power-of-two conductor f, that is $K = \mathbb{Q}[x]/(x^d+1)$, $\mathcal{O}_K = \mathbb{Z}[x]/(x^d+1)$ for $d = f/2$. For $n \geq 1$, an $\mathcal{O}_K$-module $M \in K^n$ is a finitely generated set of vectors from K^n stable under addition and multiplication by elements from $\mathcal{O}_K$. Any module admits a representation $M = \sum_{i=0}^{n-1} \mathbf{b}_i \cdot \mathfrak{b}_i$, where $\mathbf{b}_i \in K^n$ are K-linearly independent and $\mathfrak{b}_i$ are non-zero fractional ideals.

One of the cryptographically interesting example of a module arises from the the NTRU key equation [18] $h = g\phi^{-1} \bmod q$, where q is some integer, ϕ, g are elements from $\mathcal{O}_K$ with small coefficients, and ϕ is invertible. When given h as a public key (with (ϕ, g) being a secret key), the secret key recovery of the NTRU cryptosystem translates into the problem of finding a short vector (ϕ, g) in the module $M_{\mathsf{NTRU}} = \begin{pmatrix} 1 \\ h \end{pmatrix} \mathcal{O}_K \oplus \begin{pmatrix} 0 \\ q \end{pmatrix} \mathcal{O}_K$. The recovery of (ϕ, g) from h is called the NTRU Problem.

Any module $M \subset K^n$ forms a lattice in $\mathbb{C}^{nd}$ under the canonical embedding. Up until now the NTRU problem has been treated as a problem in Euclidean lattice over $\mathbb{C}^{2d}$ (in fact, over $\mathbb{Z}^{2d}$): indeed, the NTRU problem can be seen as a problem of finding a short vector in an integral lattice $\mathcal{L}_{\mathtt{NTRU}}$ of dimension $2d$ [11], without taking into account any module structure.

Dense sublattices in NTRU. Specific to NTRU is the property that the secret (ϕ, g) forms what is called a dense d-dimensional sublattice $\mathcal{L}_{\phi,g} \subset \mathcal{L}_{\mathtt{NTRU}}$. This is due to the fact that $x^i \phi h = x^i g, 0 \leq i < d$, and hence the embeddings of $(x^i \phi, x^i g)$ are also in $\mathcal{L}_{\mathtt{NTRU}}$ and they are all short and linearly independent. Hence, the sublattice $\mathcal{L}_{\phi,g}$ generated by these rotations is dense.

It has been observed [1,21,14] that for sufficiently large modulus q, called the *overstretched* NTRU regime, lattice basis reduction [29] finds a basis for $\mathcal{L}_{\phi,g}$ significantly faster than predicted by the analysis for key recovery (ϕ, g). The detection of a basis of $\mathcal{L}_{\phi,g}$ when reducing $\mathcal{L}_{\mathtt{NTRU}}$ is called *Dense Submodule Discovery (DSD).* Ducas and van Woerden [14] showed that DSD happens for $q = \Omega(d^{2.484})$.

In order to show the existence of the *overstretched* regime in NTRU, [21,14] used the so-called Pataki-Tural result [28, Lemma 1]. Informally, it gives a lower bound on the volume of sublattices relative to the shape of a basis of the full lattice. Due to the presence of the dense $\mathcal{L}_{\phi,g}$, its volume becomes smaller than the "expected" smallest volume of a sublattice in $\mathcal{L}_{\mathtt{NTRU}}$, causing a contradiction to the Pataki-Tural result. Kirchner-Fouque [21] argue that lattice reduction somehow detects this event, while Ducas-van Woerden [14] explain why and under which conditions this overstretched regime happens.

Our contributions. In this work we ask whether all these results can be translated to the algebraic setting. Indeed, M_{NTRU} contains a rank-1 dense *submodule* $\begin{pmatrix} \phi \\ g \end{pmatrix} \mathcal{O}_K$. To study the hardness of finding this dense submodule in M_{NTRU}, we contribute with the following results.

1. We formulate a generalization of Pataki-Tural lemma in the algebraic setting. We prove an analogous result that provides a lower bound on the volume of submodules with respect to the geometry of the full module. The translation of the result from 'classical' setting to the algebraic one is not straightforward: first, we are working with *pseudobases* (not bases), second, the norms we are dealing with are not Euclidean but algebraic; and third, some of the relevant tools like computation of the Hermite Normal Form in a form of *a matrix* is not available for modules.

2. We develop an algebraic analogue of the so-called Geometric Series Assumption (GSA) – an heuristic that dictates the geometry of a reduced lattice basis.
3. We combine the GSA with our algebraic version of Pataki-Tural lemma which provides us with a tool to analyze the *algebraic DSD* event – the Dense Submbodule Discovery. This enables us to analyse NTRU modules from the algebraic perspective.
4. In order to validate our heuristics we need an algorithm for algebraic lattice reduction. That is, a reduction that provides guarantees not on Euclidean norms of basis vectors, but on their algebraic norms. Kirchner, Espitau, and Fouque describe in [19,20] a version of an LLL algorithm for free modules over cyclotomic fields. Lee et al. [24] give a complete generalization of LLL to $\mathcal{O}_K$-modules for arbitrary fields K. However, this later result seems hard to implement in practice as it requires costly precomputations on high dimensional Euclidean lattices. Due to the lack of working algebraic lattice reduction[3], we implemented a version of algebraic LLL. We provided some tricks to speed up our implementation and with that we were able to verify our algebraic GSA and our analysis for the Dense Submbodule Discovery event. Our code is available at `https://github.com/mooninjune/AlgebraicLLL`.

Comparison to classical DSD. Our experiments show that so far algebraic techniques are inferior to classical lattice reduction techniques in the tasks of detecting DSD event in practice. Concretely, our implementation of algebraic LLL requires larger moduli q to succeed in detecting DSD event rather than classical BKZ [29] reduction. In order to be competitive with classical lattice reduction tools, algebraic techniques require an algorithm that finds short lattice vectors in the algebraic norm, which is so far not available. However, we believe that the theoretical tools developed in this work are independent from the development of practical algebraic reduction techniques.

2 Preliminaries

We use bold capital letters to denote matrices, bold letters for vectors, Gothic letters for ideals. The transposition of a matrix $\mathbf{B}$ is denoted as $\mathbf{B}^T$, and for matrices over $\mathbb{C}$, we denote with $\dagger$ their transposition and conjugation. The j-th element of i-th column of $\mathbf{B}$ is denoted as $\mathbf{b}_i[j]$.

Lattices. A lattice is a free $\mathbb{Z}$-module with its field of scalars being $\mathbb{R}$ or $\mathbb{C}$. We describe a lattice by its bases written as columns of a matrix $\mathbf{B} \in \mathbb{C}^{m \times n}$, where m is the dimension of the ambient space and n is the rank of the lattice. Each lattice of rank more than 1 has an infinite amount of bases. If $\mathbf{B}$ and $\mathbf{B}'$ are two $m \times n$ matrices corresponding to the two bases of the same lattice then one can

[3] The available LLL PARI-GP implementation from `https://espitau.github.io/fastlll.html` described in [19] does not seem to terminate on a 31-bit modulus q, $d = 2^6, 2^7$ within reasonable time frame

write $\mathbf{B}' = \mathbf{B} \cdot \mathbf{U}$ for a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$. The *determinant* of a lattice with a basis $\mathbf{B}$ is $\det \left(\mathbf{B}^\dagger \cdot \mathbf{B}\right)^{1/2}$.

The *ith successive minima* of a lattice $\mathcal{L}$ is denoted by $\lambda_i(\mathcal{L})$ and is the least radius r such that the ball centered at the origin contains at least i linearly independent vectors. The Euclidean norm of a shortest nonzero vector is $\lambda_1(\mathcal{L})$.

The problem of finding a nonzero lattice vector $\mathbf{v} \in \mathcal{L}$ such that it is no longer than $\gamma \cdot \lambda_1(\mathcal{L})$ for a given $\gamma \geqslant 1$ is considered to be hard in general. This problem is called the *approxSVP problem* (approximate Shortest Vector). For $\gamma = 1$, the problem is known as SVP. Let $\mathbf{t} \in \mathbb{R}^m$ be a vector with $\mathbf{u} \in \mathcal{L}$ being closest to $\mathbf{t}$. The *approxCVP problem* (approximate Closest Vector Problem) asks to find $\mathbf{v} \in \mathcal{L}$ given $\mathbf{t} \in \mathrm{Span}_{\mathbb{R}}(\mathcal{L})$ such that $\|\mathbf{t} - \mathbf{v}\| \leq \gamma \cdot \|\mathbf{t} - \mathbf{u}\|$. For $\gamma = 1$, the problem is called CVP.

Number fields. Let K be a number field of degree d and $\mathcal{O}_K$ be its ring of integers. The number field K has r_1 real embeddings and $2r_2$ complex embeddings into $\mathbb{C}^d$ with $r_1 + 2r_2 = d$. We denote them as σ_i for $0 \leqslant i < d$. There are two ways to embed elements from K. The canonical (Minkowski) embedding $\mathcal{F}$ of an element $k \in K$ into $\mathbb{C}^d$ is defined as the vector $(\sigma_i(k))_{0 \leqslant i < d}$. The coefficient embedding of an element $k = \sum_{l<d} c_l \cdot \zeta^l \in K$ into $\mathbb{R}^d$ is defined as the vector $(c_0, \ldots, c_{d-1})$.

For $k \in K$, the field norm is defined as $\mathcal{N}(k) = \prod_{\sigma_i} \sigma_i(k)$. For a fractional ideal $\mathfrak{a} \subset K$ its norm $\mathcal{N}(\mathfrak{a})$ is defined as the cardinality of the factor-ring $K/\mathfrak{a}$. The trace $\mathrm{Tr}(k)$ of k is defined as $\sum_{\sigma_i} \sigma_i(k)$. Both norm and trace of an element $k \in K$ are in $\mathbb{Q}$. Let L be a subfield of a number field K. The relative norm $\mathcal{N}_{K/L}(k)$ for some $k \in K$ is defined as the determinant (over L) of the linear map: $K \to K : x \mapsto k \cdot x$. The trace of this linear map (over L) is denoted as $\mathrm{Tr}_{K/L}(k)$ and is called the relative trace. An $\mathcal{O}_K$ element of algebraic norm ± 1 is called a unit. The set of all $\mathcal{O}_K$ units forms a multiplicative group called the unit group.

For a given fractional principal ideal $\mathfrak{a}$ the problem of finding $a \in K$ such that $a \cdot \mathcal{O}_K = \mathfrak{a}$ is called the Principal ideal Problem (PIP). For cyclotomic fields it can be solved classically in sub-exponential time [3] and in polynomial time using quantum computers [5].

We define $K_{\mathbb{R}}$ as the tensor product $K \otimes_{\mathbb{Q}} \mathbb{R}$. We write $K_{\mathbb{R}}^+$ as the subset of $K_{\mathbb{R}}$ with nonnegative coordinates under the canonical embedding. We can take the square root of $k \in K_{\mathbb{R}}^+$ by applying it coordinate-wise after the canonical embedding. For $k \in K_{\mathbb{R}}$, its conjugation $\bar{k} \in K_{\mathbb{R}}$ is well-defined since $K_{\mathbb{R}} \subset \mathbb{C}^d$.

The ring of integers $\mathcal{O}_K$ is a lattice of rank d under the canonical embedding. The absolute value of the discriminant of K, denoted $\mathit{\Delta}_K$, is the squared volume of $\mathcal{O}_K$, namely $\mathit{\Delta}_K = |\det(\sigma_i((\mathbf{b}_j))_{i,j}|^2$ for any $\mathbb{Z}$-basis $\{\mathbf{b}_j\}_j$ of $\mathcal{O}_K$.

For $n \in \mathbb{N}^+$, K^n is a vector space equipped with Hermitian inner product $\langle \mathbf{u}, \mathbf{v} \rangle_{K_{\mathbb{R}}} = \sum (\mathbf{u})_i \cdot \overline{(\mathbf{v})}_i, 0 \leqslant i < n$ where $\overline{\mathbf{v}}$ denotes the conjugation over $K_{\mathbb{R}}$ applied to $\mathbf{v}$ component-wise. For an intermediate field $L \subset K$, we define $\|k\|_{K/L} = (\mathrm{Tr}_{K/L}(k \cdot \bar{k}))^{1/2}$ and $\|k\| = \|k\|_{K/\mathbb{Q}}$. The Euclidean norm of vector $\mathbf{v}$ over $\mathbb{Q}$ is defined as $\|\mathbf{v}\| = \mathrm{Tr}_{K/\mathbb{Q}}(\langle \mathbf{v}, \mathbf{v} \rangle_{K_{\mathbb{R}}})^{1/2}$. The algebraic norm $\mathbf{v} \in K^n$ is defined as $\mathcal{N}(\langle \mathbf{v}, \mathbf{v} \rangle_{K_{\mathbb{R}}})^{1/2}$. By abuse of notations, we write $\langle \mathbf{a}, \mathbf{b} \rangle := \langle \mathbf{a}, \mathbf{b} \rangle_{K_{\mathbb{R}}}$

when $\mathbf{a}, \mathbf{b} \in K^n$. The canonical embedding of $\mathbf{v} = (v_0, \dots, v_{n-1}) \in K^n$ if defined as $\mathcal{F}(\mathbf{v}) = (\mathcal{F}(v_0), \dots, \mathcal{F}(v_{n-1})) \in \mathbb{C}^{nd}$. A matrix $\mathbf{U} \in \mathcal{O}_K^{n \times n}$ is called unimodular if $\mathcal{N}(\det \mathbf{U}) = \pm 1$.

Vectors $\mathbf{v}_0, \dots, \mathbf{v}_{n-1}$ are said to be *$K_{\mathbb{R}}$-linearly independent* if there is no nontrivial linear combination with the coefficients $c_i \in K_{\mathbb{R}}$ such that $\sum c_i \cdot \mathbf{v}_i = 0$.

Cyclotomic Number Fields. Let $\zeta \in \mathbb{C}$ be an f-th root of unity for some $f \in \mathbb{N}$. The number field $K = \mathbb{Q}(\zeta)$ is called the f-th cyclotomic field. Its ring of integers $\mathcal{O}_K$ coincides with $\mathbb{Z}[\zeta]$ and admits the orthogonal integral basis $\{1, \zeta, \dots, \zeta^{d-1}\}$ under the coefficient embedding. Its degree is given by $d = \varphi(f)$ for φ being the Euler totient function. Cyclotomic fields of degree d that is a power of two are called power-of-2 cyclotomic fields. In that case we have $\lambda_1(\mathcal{O}_K)$ is $\sqrt{d}$. For a power-of-2 cyclotomic field K of degree d, as the direct consequence of [32, Proposition 2.1], we have that $\log |\Delta_K| = d \cdot \log(d)$.

To bound the algebraic norm of a cyclotomic number field element, we need the following lemma which sometimes being referred to as the algebraic-geometric inequality.

Lemma 1. *Let K be a cyclotomic field. Then for all $k \in K$:*

$$\mathcal{N}(k) \leqslant d^{-d/2} \cdot \|k\|^d$$

Algebraic Lattices. A projective $\mathcal{O}_K$ module M of rank n is defined as $M = \mathbf{b}_0 \cdot \mathfrak{b}_0 \oplus \ldots \oplus \mathbf{b}_{n-1} \cdot \mathfrak{b}_{n-1}$, where all $\mathbf{b}_i$'s are $K_{\mathbb{R}}$-linearly independent and $\mathfrak{b}_i$'s are fractional nonzero ideals. We will be focusing on the case $\mathfrak{b}_i = \mathcal{O}_K, \forall i$.

A tuple of pairs $((\mathbf{b}_0, \mathfrak{b}_0), \dots, (\mathbf{b}_{n-1}, \mathfrak{b}_{n-1}))$ is called a pseudobasis of M. If an algebraic module admits a pseudobasis with all ideals equal to $\mathcal{O}_K$, the module is said to be free. In that case we refer to $\mathbf{b}_i$'s as just a basis. We can represent a (pseudo)basis as a matrix over K with $\mathbf{b}_i$'s being its columns, and an ordered set of n fractional ideals. Let $\mathbf{B}$ be such a matrix. The ideal $\det_K M = \sqrt{\det(\mathbf{B}^\dagger \cdot \mathbf{B})} \cdot \prod_i \mathfrak{b}_i$ is called the determinant of M. In the special case of a free module, we have $\det_K M = \sqrt{\det(\mathbf{B}^\dagger \cdot \mathbf{B})} \cdot \mathcal{O}_K$.

An algebraic module M endowed with the inner product $\langle \mathbf{v} , \mathbf{u} \rangle$ for every $\mathbf{v}, \mathbf{u} \in M$ is called an *algebraic lattice $\mathcal{L}$*. The rank of an algebraic lattice is its rank as a module. An algebraic lattice $\mathcal{L}$ of rank n forms a lattice over $\mathbb{C}^{nd}$ under the canonical embedding, e.g., $\mathcal{F}(\mathcal{L}) = \{\mathcal{F}(\mathbf{v}) \mid \mathbf{v} \in \mathcal{L}\} \subset \mathbb{C}^{nd}$ is a lattice. The determinant of $\mathcal{L}$ is $\det \mathcal{L} = \Delta_K^{n/2} \cdot \mathcal{N}(\det_K(M))$. Any submodule of M with the same inner product is called an *algebraic sublattice*.

An algebraic lattice can have infinitely many pseudobases so it is crucial to determine a criteria which tells if two pseudobases represent the same module. For this task we use the definition given in [10, Proposition 1.4.2].

Proposition 1. *Two algebraic lattices given by $((\mathbf{a}_0, \mathfrak{a}_0), \dots, (\mathbf{a}_{n-1}, \mathfrak{a}_{n-1}))$ and $((\mathbf{b}_0, \mathfrak{b}_0), \dots, (\mathbf{b}_{n-1}, \mathfrak{b}_{n-1}))$ form the same lattice if and only if there exists an invertible matrix $\mathbf{U} \in K^{n \times n}$ such that $\mathbf{B} = \mathbf{AU}$, every $\mathbf{u}_i[j] \in \mathfrak{a}_j \cdot \mathfrak{b}_i^{-1}$ and $\mathbf{u}'_i[j] \in \mathfrak{a}_i^{-1} \cdot \mathfrak{b}_j$ for $\mathbf{u}'_i$ being columns of the inverse matrix $\mathbf{U}' = \mathbf{U}^{-1}$ for $0 \leqslant i < n$. When the module is free, the determinant of such $\mathbf{U}$ is an $\mathcal{O}_K$ unit.*

We shall make use of the following definition of a primitive vector from K^n.

Definition 1 (Primitive vector). *Let* $\mathbf{B}$ *be a basis of an algebraic lattice. A vector* $\mathbf{v}$ *of that lattice with coefficients* $(c_0, \ldots, c_{n-1})$ *with respect to the basis* $\mathbf{B}$ *is said to be primitive if* $\bigoplus_{0 \leqslant i < n} c_i \cdot \mathcal{O}_K = \mathcal{O}_K$.

This definition is correct in the sense that if a vector is primitive with respect to a given basis of an algebraic lattice, then it is primitive for every other basis.

The algebraic minimum of an algebraic lattice $\mathcal{L}$ is defined as

$$\lambda_1^{\mathcal{N}}(\mathcal{L}) = \inf_{\mathbf{v} \in \mathcal{L} \setminus \{0\}} \mathcal{N}(\mathbf{v}).$$

The problem of finding a vector $\mathbf{v} \in \mathcal{L}$ that is a $\gamma_{\mathcal{N}}$ approximation of $\lambda_1^{\mathcal{N}}(\mathcal{L})$ for $\gamma_{\mathcal{N}} \geqslant 1$ is called the *algebraic approxSVP problem.* The algebraic minimum can be bounded using the Euclidean minima both from below and above [24, Lemma 2.2].

Lemma 2 ([24, Lemma 2.2]). *Let* K *be a number field of degree* d *with* $\mathcal{O}_K$ *– its ring of integers. For a lattice* $\mathcal{L}$ *over* $\mathcal{O}_K$*, the following holds:*

$$d^{-d/2} \lambda_1(\mathcal{L})^d \cdot \Delta_K^{-1/2} \leqslant \lambda_1^{\mathcal{N}}(\mathcal{L}) \leqslant d^{-d/2} \lambda_1(\mathcal{L})^d$$

Log-unit Lattice. An $\mathcal{O}_K$ element of algebraic norm ±1 is called a unit. The set of all $\mathcal{O}_K$ units forms a multiplicative group called the unit group. In the case of cyclotomic fields we consider its finite subgroup consisting of the cyclotomic units. In number field of prime power conductor f, the cyclotomic group is generated by the elements of the form: $\frac{\zeta^i - 1}{\zeta - 1}$ for all i coprime to $d = \varphi(f)$ [32].

We define the log-embedding Log : $K_{\mathbb{R}}^{\times} \to \mathbb{R}^d$ for some $k \in K$ as $\mathrm{Log}(k) = (\log |\sigma_0(k)|, \ldots, \log |\sigma_{d-1}(k)|)$.

After the log-embedding all units of $\mathcal{O}_K$ belong to the hyperplane $H \subset \mathbb{R}^{d-1}$ that is orthogonal to the all-ones vector. It consists of vectors $\mathbf{h} \in \mathbb{R}^{d/2-1}$ such that the sum of their coordinates $\sum_{i<d/2} (\mathbf{h})_i$ is equal to zero. Moreover, under the log-embedding, units form a lattice called the log-unit lattice.

Gram-Schmidt orthogonalization. Let $\mathbf{B} \in K^{m \times n}$ be a basis of some lattice $\mathcal{L}$. The Gram matrix of $\mathbf{B}$ is defined as $\mathbf{G} = \mathbf{B}^{\dagger} \cdot \mathbf{B}$. It contains the information about the Hermitian inner product between every basis vector.

The Gram Schmidt vectors $\{\mathbf{b}_i^*\}_i$ for a basis $\mathbf{B} = \{\mathbf{b}_i\}_i$ are defined as:

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{k<i} \frac{\langle \mathbf{b}_i, \mathbf{b}_k^* \rangle}{\langle \mathbf{b}_k^*, \mathbf{b}_k^* \rangle} \cdot \mathbf{b}_k^* \quad \text{for } 0 \leqslant i < n. \tag{1}$$

Following the LLL algorithm described in [27], we set $r_{i,i} = \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle \in K_{\mathbb{R}} \subset \mathbb{R}^+$ for every $i < n$ and $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_k^* \rangle}{\langle \mathbf{b}_k^*, \mathbf{b}_k^* \rangle}$ for $i \geqslant j$. Then, $\mu_{i,j} = r_{i,j}/r_{j,j}$ for $i > j$, and

$$r_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle - \sum_{k=0}^{j-1} \mu_{j,k} \cdot r_{i,k}, \; i > j. \tag{2}$$

As the cyclotomic fields are CM-fields, we have that Gram-Schmidt vectors are over K^n and all $\mu_{i,j}, r_{i,j}$ are in K. The projection of a vector $\mathbf{v}$ on a vector $\mathbf{u}$ is defined as: $\pi_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v},\mathbf{u}\rangle}{\langle \mathbf{u},\mathbf{u}\rangle} \cdot \mathbf{u}$. We shall make use of basis vectors projected orthogonally to the vector space spanned by $\{\mathbf{b}_0^*, \dots \mathbf{b}_{i-1}^*\}$. We denote such projection as $\pi_i(\mathbf{v}) = \mathbf{v} - \sum_{j<i} \pi_{\mathbf{b}_j^*}(\mathbf{v})$.

Closely related to Gram-Schmidt orthogonalization is QR-decomposition: for $\mathbf{B} \in \mathcal{O}_K^{m\times n}$, there exist matrices $\mathbf{Q} \in \mathcal{O}_K^{m\times m}$ and $\mathbf{R} \in \mathcal{O}_K^{m\times n}$ such that $\mathbf{B} = \mathbf{Q}\cdot\mathbf{R}$ and $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ is orthonormal. The diagonal entries of the R-factor are given by $r_{i,i}^{1/2}$, off-diagonal entries are $\left(r_{i,j} \cdot r_{j,j}^{1/2}\right)$ for $i > j$.

We show in Algorithm 2.1, following [27, Fig. 4], how to compute the Gram-Schmidt coefficients in a lazy manner using the Cholesky factorization algorithm. On input, the algorithm receives the Gram matrix of a lattice basis and operates on it in order to compute (or update) $\{\mu_{i,j}\}, \{r_{i,j}\}$. As proposed in [27], working with a Gram matrix, rather than a basis, improves some of the precision issues that arise in practice. As we are interested in making the computations practical, our implementation follows this approach. The correctness of the algorithm relies on Equation (2), we do not show it here but instead refer to [27, Section 3.2]. For our lazy implementation of LLL we update only the values relevant for a certain step. This is why in Algorithm 2.1 we additionally provide on input the positions for which we want $\{\mu_{i,j}\}, \{r_{i,j}\}$ to be computed or updated.

Algorithm 2.1 compute_GSO

```
Input:  $\mathbf{G} \in K^{n\times n}$ – Gram matrix of a lattice.
        $\{\mu_{i,j}\}_{0\leqslant i,j<s}, \{r_{i,j}\}_{0\leqslant i<s,\forall j}$ – GSO coefficients
Output: $\{\mu_{i,j}\}_{0\leqslant i,j\leqslant e}, \{r_{i,j}\}_{0\leqslant i\leqslant e,\forall j}$ – Gram Schmidt coefficients up to position $e \geq s$.
for $i = s, \dots, e$ do
    for $j < i$ do
        $r_{i,j} := \mathbf{G}_{i,j}$
        Set $r_{i,j} := r_{i,j} - \mu_{j,k} \cdot r_{i,k}$ for $k < j$
        $\mu_{i,j} := r_{i,j}/r_{j,j}$
    Set $s_0^{(i)} := \mathbf{G}_{i,i}$;
    for $1 \leqslant j \leqslant i$ do
        $s_j^{(i)} := s_{j-1}^{(i)} - \mu_{i,i-1} \cdot r_{i,j-1}$
    $r_{i,i} := s_i^{(i)}$
return $\{\mu_{i,j}\}_{0\leqslant i,j\leqslant e}, \{r_{i,i}\}_{0\leqslant i\leqslant e}$
```

Classical LLL. We call non-algebraic LLL algorithms classical, and by classical lattices we mean lattices defined over $\mathbb{R}^n$ with no underlying module structure.

A basis $\mathbf{B} \in \mathbb{R}^{m\times n}$ of a rank m lattice $\mathcal{L} \subset \mathbb{R}^n$ is δ-LLL reduced if for all $0 \leqslant i < n, 0 \leqslant j < m$ and some $1/4 < \delta < 1$ the following conditions are met:

$$\begin{aligned} \mu_{i,j} &\leqslant 1/2 \quad \text{(Size reducedness)},\\ \delta\pi_i(\mathbf{b}_i) &\leqslant \pi_i(\mathbf{b}_{i+1}) \quad \text{(Lovász condition)}. \end{aligned}$$

Classical LLL reduction can be computed in time $\mathrm{poly}(n, \log \max_i\{\|\mathbf{b}_i\|\})$ [25].

As we focus on power-of-2 cyclotimic number fields, we know an orthogonal $\mathbb{Z}$-basis of $\mathcal{O}_K$. This basis allows us to solve CVP on $\mathcal{O}_K$ efficiently and exactly. Since $\zeta^i, 0 \leqslant i < d$ form a basis of $\mathcal{O}_K$ and all $\mathcal{F}(\zeta^i)$ have Euclidean norm $\sqrt{d}$, it holds that $\lambda_d(\mathcal{O}_K) = \sqrt{d}$.

Lemma 3. *Let K be cyclotomic field of degree d. Let $\mathcal{L} = \mathcal{F}(\mathcal{O}_K)$ be lattice obtained from $\mathcal{O}_K$ using the coefficient embedding. For a given target vector $\mathbf{t} \in \mathrm{Span}_{K_\mathbb{R}}(\mathcal{L})$ one can find vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{t} - \mathbf{v}\| \leqslant d \cdot 2^{d/2-1}$.*

Proof. Notice that $\lambda_i(\mathcal{L}) \leqslant \sqrt{d}\ \forall i$. Suppose we are given a target vector $\mathbf{t}$. Babai nearest plane algorithm [2] can solve approxCVP with approximation factor $2^{d/2}$. We can bound distance from $\mathbf{t}$ to $\mathcal{L}$ with the covering radius of $\mathcal{L}$ as $\sqrt{d}/2 \cdot \sqrt{\lambda_d}$ [26, Exercise 11]. Finally $\|\mathbf{t} - \mathbf{v}\| \leqslant 2^{d/2} \cdot d/2 \leqslant d \cdot 2^{d/2-1}$. □

Let k be and element of $\mathcal{O}_K$. Let $\mathbf{t} = \mathcal{F}(k)$. We define the rounding $\lfloor k \rceil$ over $\mathcal{O}_K$ as $v \in K$ corresponding to a vector $\mathbf{v}$ obtained by applying Lemma 3 to the vector $\mathbf{t}$. In case of power-of-two cyclotomic fields, such rounding is a 'usual' coordinate-wise rounding over $\mathbb{Z}$. The efficiency of this rounding is important for the size-reduction process described in the next section.

BKZ reduction. In this work we rely on classical BKZ lattice reduction algorithm [29,9,31]. We do not focus on the details of the algorithm, but use the following facts about the quality of its output. BKZ algorithm has an important integral parameter $\beta \geqslant 2$ called *blocksize* that controls the quality of the output basis. BKZ algorithm outputs shorter vectors than the LLL algorithm, but requires more time to terminate. The runtime of the BKZ algorithm is at least exponential in β.

The quality of a reduced basis is usually studied using so-called log-profile originally introduced by Schnorr in [30].

Definition 2. *Let $\mathbf{B} \in \mathbb{Z}^{m \times n}$ be a basis of some lattice $\mathcal{L}$. The vector $\mathbf{p} = (\log(\|\mathbf{b}_0^*\|), \ldots, \log(\|\mathbf{b}_{n-1}^*\|))$ is called the log-profile of $\mathbf{B}$.*

In this work we focus on a special case of lattices called q-ary. An n-dimensional lattice $\mathcal{L} \subset \mathbb{Z}^n$ is called q-ary for some $q \in \mathbb{N}, q > 1$ if $q \cdot \mathbb{Z}^n \subset \mathcal{L}$. In this work we will consider the case of lattices that admit a basis with profile $\mathbf{p}$ given by

$$\mathbf{p} = (\overbrace{\log q, \ldots, \log q}^{n/2}, \overbrace{0, \ldots, 0}^{n/2}).$$

The BKZ output quality relates the decay of $\mathbf{p}_i = \log(\|\mathbf{b}_i^\star\|)$ using the value $\alpha_\beta \approx \left(\frac{\dim(\mathcal{L})}{2\pi e} \cdot \det(\mathcal{L})^{\frac{2}{\dim(\mathcal{L})}}\right)^{1/(\beta-1)}$. That is, α_β controls the slope of the log-profile of the basis output by BKZ. The following heuristic, called Z-shape Geometric Series Assumption (ZGSA), provides a fairly accurate prediction of this log-profile for large enough $\beta' s$.

Heuristic 1 ([14, Heuristic 2.8]) *Let $\mathbf{B} \in \mathbb{Z}^{n\times n}$ be a basis of an n-dimensional q-ary lattice (for n even) with its log-profile given by*

$$\mathbf{p} = (\overbrace{\log q, \dots, \log q}^{n/2}, \overbrace{0, \dots, 0}^{n/2})$$

After BKZ-β reduction called on $\mathbf{B}$ the profile vector $\mathbf{p}'$ of a resulting reduced basis is given by

$$\mathbf{p}'_i = \begin{cases} \log q, & i \leqslant n/2 - n' \\ \log q \cdot (1 - \frac{i-n/2+n'}{2n'}), & n/2 - n' < i < n/2 + n' - 1 \\ 0, & i \geqslant n/2 + n' - 1, \end{cases} \tag{3}$$

for $n' = (1 + \ln q / \ln \alpha_\beta)/2$.

2.1 Algebraic Lattice Reduction

Below we give a definition of an algebraically LLL reduced basis following [24, Definition 3.1].

Definition 3 (LLL reduced basis). *A pseudobasis $(\mathbf{B}, \{\mathfrak{b}_i\}_i)$ of an algebraic lattice is said to be α-LLL reduced for some real $\alpha > 1$ if $\alpha \cdot \mathcal{N}(r_{i+1,i+1} \cdot \mathfrak{b}_{i+1}) \geqslant \mathcal{N}(r_{i,i} \cdot \mathfrak{b}_i)$.*

In order to achieve the condition from Definition 3, algebraic LLL algorithm looks at projective rank-2 submodules of B defined by the bases

$$M_i = \begin{pmatrix} \mathbf{b}_i^* \\ \pi_i(\mathbf{b}_{i+1}) \end{pmatrix}, \quad 0 \leq i < n-1. \tag{4}$$

Recall that $r_{i,i} = \|\mathbf{b}_i^*\|^2_{K_\mathbb{R}}$. The purpose of LLL is to bound the decay in the norms of $r_{i,i}$. As in the classical LLL, it is achieved by finding a short vector in some rank-2 sublattice and replacing $\mathbf{b}_i^*$ with this short vector. The difference to the classical case is that we are interested in the *algebraic norm* rather than the Euclidean norm.

To study the behavior of the algebraic LLL algorithm we generalize the concept of the log-profile which carries information about algebraic properties of bases and is a measure of reducedness.

Definition 4. *Let $(\mathbf{B} \in K^{m\times n}, \{\mathfrak{b}_i\}_i)$ be a pseudobasis of an algebraic lattice. Let $\mathbf{Q}\cdot\mathbf{R}$ be its QR-decomposition. Consider $\{r_{i,i}^{1/2} \in K_\mathbb{R}^+ \subset \mathbb{R}^+\}$, the set of diagonal elements of $\mathbf{R}$. The ordered set $\mathbf{p}((\mathbf{B}, \{\mathfrak{b}_i\}_i)) = \{\log(|r_{i,i}| \cdot \mathcal{N}(\mathfrak{b}_i))/2\}_{0\leqslant i<n}$ is called the log-profile of $\mathbf{B}$.*

In the case when $K = \mathbb{Q}$ this definition coincides with Definition 2. Notice that $\sum_{i=0}^{n-1} \mathbf{p}_i((\mathbf{B}, \{\mathfrak{b}_i\}_i)) = \log(\mathcal{N}(\det \mathbf{B}) \cdot \prod_{i=0}^{n-1} \mathcal{N}(\mathfrak{b}_i))$. Together with the condition $\alpha \cdot \mathcal{N}(r_{i+1,i+1}) \geqslant \mathcal{N}(r_{i,i})$ this guarantees that the log-profile on an LLL reduced basis cannot decrease too rapidly.

2.2 NTRU Modules

The NTRU problem asks to find $\phi, g \in \mathcal{O}_K$ such that ϕ is invertible in $\mathbb{Z}[X]/(q, X^d+1)$ and the coefficients of both ϕ, g are chosen uniformly at random[4] from $\{-1, 0, 1\}$ under the coefficient embedding, when given

$$h = g \cdot \phi^{-1} \mod q. \tag{5}$$

Algebraic NTRU lattice. The NTRU problem can be viewed as finding a short element in the rank-2 $\mathcal{O}_K$-module defined as

$$M_{\mathsf{NTRU}} = \begin{pmatrix} 1 \\ h \end{pmatrix} \mathcal{O}_K \oplus \begin{pmatrix} 0 \\ q \end{pmatrix} \mathcal{O}_K. \tag{6}$$

In particular, we have $(\phi, g) \in M$ since $\begin{pmatrix} 1 \\ h \end{pmatrix} \phi + \begin{pmatrix} 0 \\ q \end{pmatrix} k_q = \begin{pmatrix} \phi \\ g \end{pmatrix}$ for some $k_q \in \mathcal{O}_K$ that satisfies $h = g\phi^{-1} + qk_q$. Furthermore, $M_{\phi,g} = \begin{pmatrix} \phi \\ g \end{pmatrix} \mathcal{O}_K \subset M_{\mathsf{NTRU}}$ is a so-called *dense* rank-1 submodule of M_{NTRU}. It has been observed in [1] (and further studied in [21,14]) that for sufficiently large q finding a basis for this dense submodule is easier than the recovery of (ϕ, g). In that case for a large enough blocksize β, the BKZ algorithm recovers a basis of the dense submodule. When this happens, such event is called *the DSD event.* Precisely,

Definition 5. *The dense submodule Discovery. Let $\mathbf{B} \in \mathbb{Z}^{n \times n}$ be a $\mathbb{Z}$-basis of an NTRU module. We define the DSD as an event when BKZ-β called on $\mathbf{B}$ returns a basis $[\mathbf{M}|\mathbf{B}']$ for $\mathbf{B}', \mathbf{M} \in \mathbb{Z}^{n \times (n/2)}$ and a module spanned by $\mathbf{M}$ contains the secret vector $(\mathbf{f}, \mathbf{g})$ corresponding to the coefficient embedding of (ϕ, g).*

The NTRU modules viewed as a $\mathbb{Z}$-lattices are q-ary. The larger q is, the easier it is to recover $M_{\phi,g}$ (for a fixed d). Note that once a basis for this dense rank-1 submodule is found, one can focus on finding (ϕ, g) in this smaller dimensional rank-1 submodule. Experiments suggest [22] that indeed in practice the problem of finding (ϕ, g) is not significantly harder than obtaining a basis for $M_{\phi,g}$ in the case of a sufficiently large q.

In [21] the authors combine Heuristic 1 with the Pataki-Tural lemma [28, Lemma 1] to obtain a criteria to deduce which BKZ blocksize β is sufficient to trigger the DSD event on NTRU lattices. A more precise statement following the same arguments can be found in [14, Claim 2.12]:

Heuristic 2 ([14, Claim 2.12]) *Let $\mathcal{L}_q$ be an NTRU lattice of dimension $2d$ over $\mathbb{Z}$ with a dense submodule $\mathcal{L}_{\phi,g}$. Under the ZGSA, BKZ-β triggers the DSD event if:*

$$\det \mathcal{L}_{\phi,g} < q^{\frac{n'-1}{2}} \cdot \alpha_\beta^{-\frac{1}{2}(n'-1)^2},$$

where $n' = (1 + \ln q / \ln \alpha_\beta)/2$.

[4] Several versions of NTRU with varying Hamming weights of ϕ, g exist [8], our results extend to these other versions too.

The asymptotic analysis provided in [14] suggests that the DSD event (as per definition above) precedes the recovery of the secret vector for $\log q \geqslant d^{2.783+o(1)}$. The NTRU modules with q satisfying this inequality are called *overstretched*. For fixed values of $\det \mathcal{L}_{\phi,g}$ and d, Heuristic 2 suggests that the larger q is, the smaller β is required to trigger the DSD event. The value of blocksize β sufficient to trigger the DSD event is estimated as $\tilde{\Theta}(d/\log(q)^2)$ where $\tilde{\Theta}(f(x))$ means that there exist some constant $c \geqslant 0$ such that $f(x) = O(f(x) \cdot |\log f(x)|^c)$.

3 Pataki-Tural Lemma for Modules

In this section we are generalising the concept of DSD events to the algebraic setting. As in the classical case the DSD event should lead to a discovery of a smaller rank sublattice that still contains the required short vector. This will reduce the search problem to an easier one. We start our study of algebraic DSD events with introducing the algebraic analogues of necessary lemmas. After that we describe a technique that allows us to descend NTRU modules defined over a number field K to a some proper subfield $L \subset K$ in Section 3.1. In Section 3.2 we formulate an algebraic analogue of ZGSA and, after all necessary tools are developed, generalize the definition of DSD events to the algebraic setting. All these results combined yield an estimator for algebraic DSD.

Lemma 4. *Let $\mathcal{L}$ be an algebraic lattice in K^m given by a pseudobasis $(\mathbf{B}, \{\mathfrak{b}_i\}_{i<n})$. Let $\mathcal{P}$ be rank-k algebraic sublattice. Then there exists an ordered set $\{\mathbf{y}_i\}_{i<k}$ of linearly independent vectors of $\mathcal{P}$ such that:*

$$\mathbf{y}_{k-1} \in \operatorname{Span}\{\mathbf{b}_i\}_{0 \leqslant i \leqslant n-1}, \ldots, \mathbf{y}_0 \in \operatorname{Span}\{\mathbf{b}_i\}_{0 \leqslant i \leqslant n-k}, \text{ and}$$
$$\mathbf{y}_{k-1} \notin \operatorname{Span}\{\mathbf{b}_i\}_{0 \leqslant i \leqslant n-2}, \ldots, \mathbf{y}_0 \notin \operatorname{Span}\{\mathbf{b}_i\}_{0 \leqslant i \leqslant n-k-1}.$$

Proof. Without loss of generality all $\mathbf{b}_i \in \mathcal{O}_K^m$ and all $\mathfrak{b}_i \subset \mathcal{O}_K$, otherwise we can scale $\mathcal{L}$ accordingly.

Let $\mathbf{X} \in K^{m \times k}$ be a rank-k matrix with its columns $\mathbf{x}_i \in \mathcal{P}$. Each $\mathbf{x}_\kappa$ is a $\mathcal{O}_K$-linear combination of $\mathbf{b}_j$ for $0 \leqslant j < n$. More precisely, for all $\kappa < k, j < n$ there exist $u_{\kappa,j} \in \mathfrak{b}_j \subset \mathcal{O}_K$ such that we can write

$$\mathbf{x}_\kappa = \sum_{0 \leqslant j < n} u_{\kappa,j} \cdot \mathbf{b}_j, \quad 0 \leqslant \kappa < k.$$

In matrix from the equation above can be written as $\mathbf{X} = \mathbf{B} \cdot \mathbf{U}$ for $\mathbf{X} \in K^{m \times k}, \mathbf{U} \in \mathcal{O}_K^{n \times k}, \mathbf{B} \in K^{m \times n}$.

Every $\mathcal{O}_K$-linear combination of vectors $\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}$ is also a vector from $\mathcal{P}$ as it holds that $\zeta^i \mathbf{x}_j \in \mathcal{P}$ for ζ – a primitive root of K, any $i \in \mathbb{Z}$ and any $0 \leq j \leq k-1$. Hence $\mathcal{L}(\mathbf{X}, \{\mathcal{O}_K\}^k)$ is a free submodule of $\mathcal{P}$.

Next we apply a transformation to $\mathbf{U}$ reminiscent to the column-echelon form for Euclidean lattice bases. Notice that any transformation of the columns of $\mathbf{U}$ given by $\mathbf{u}_\kappa \leftarrow \alpha \cdot \mathbf{u}_\kappa + \beta \cdot \mathbf{u}_\ell$ for some $\alpha, \beta \in \mathcal{O}_K, \alpha \neq 0, \kappa \neq \ell$ sends $\mathbf{x}_\kappa$ to $\alpha \cdot \mathbf{x}_\kappa + \beta \cdot \mathbf{x}_\ell$ accordingly. In addition, any such transformation preserves both the rank of $\mathbf{X}$ and the inclusion $\mathcal{L}(\mathbf{X}, \{\mathcal{O}_K\}^k) \subset \mathcal{P}$ (since every $\mathbf{B} \cdot \mathbf{u}_\kappa \in \mathcal{O}_K^m$ remains to be a linear $\mathcal{O}_K$ combination of vectors from $\mathcal{P}$). Any permutation of the columns preserves the module and, thus, the inclusion as well.

Now we mimic the column-echelon form computation algorithm for $\mathbf{U}$ in Algorithm 3.1. On input $\mathbf{U} \in \mathcal{O}_K^{n \times k}$, it returns $\mathbf{T} \in \mathcal{O}_K^{n \times k}$ such that the following inclusions hold

$$\mathcal{L}(\mathbf{BUT}, \{\mathcal{O}_K\}^k) \subseteq \mathcal{L}(\mathbf{X}, \{\mathcal{O}_K\}^k) \subseteq \mathcal{P}.$$

The routine is described in Algorithm 3.1. It takes $\mathbf{U}$ as an input and uses a subroutine $\textsc{lne}(\mathbf{u}) = \max\{i \mid \mathbf{u}[i] \neq 0\}$ that returns the index of the last nonzero element of a vector $\mathbf{u}$.

Algorithm 3.1 Echelon Form for Algebraic lattices

```
Input:  $\mathbf{U} \in K^{m \times k}$ – a matrix for $k \leqslant m$ of rank $k$.
Output: $\mathbf{T} \in K^{m \times k}$ – an upper-triangular matrix.
1: for $\ell = m-1, \dots, m-k$ do
2:     Sort $\{\mathbf{u}_0, \dots, \mathbf{u}_{\ell-(m-k)}\}$ in non decreasing order of $\textsc{lne}(\mathbf{u}_i)$.
3:     if $\mathbf{u}_{\ell-(m-k)}[\ell] \neq 0$ then
4:         for $\kappa = \ell - 1 - (m-k), \dots, 0$ do
5:             Find $\alpha, \beta \in \mathcal{O}_K, \alpha \neq 0$ such that $\alpha \mathbf{u}_\kappa[\ell] + \beta \mathbf{u}_{\ell-(m-k)}[\ell] = 0$
6:             $\mathbf{u}_\kappa := \alpha \cdot \mathbf{u}_\kappa + \beta \cdot \mathbf{u}_{\ell-(m-k)}$
7: return $\mathbf{T} = (\mathbf{u}_0, \dots, \mathbf{u}_{k-1})$
```

The algorithm iterates for $\ell = m-1, \dots, m-k$ (corresponding to rows) and $\kappa = \ell - 1 - (m-k), \dots, 0$ (corresponding to columns). At a fixed ℓ we sort the first ℓ columns such that the indices of the last nonzero entry of consequent columns do not decrease. Now it holds that either the new value of $\mathbf{u}_{\ell-(m-k)}[\ell]$ is nonzero, or the entire ℓ-th row is zero. The latter situation can only occur if $\operatorname{rank} \mathbf{U} < k$ which contradicts the rank of $\mathbf{X}$. By applying the transformation described in Line 6 of Algorithm 3.1, we can ensure that for all $0 \leqslant \kappa < \ell - (m-k)$ we have $\mathbf{u}_\kappa[\ell] = 0$ by solving the equation $\alpha \mathbf{u}_\kappa[\ell] + \beta \mathbf{u}_{\ell-(m-k)}[\ell] = 0$ for arbitrary α and β from $\mathcal{O}_K$ with $\alpha \neq 0$. Once the outer loop over ℓ is finished, we obtain an upper triangular $\mathbf{T}$ which can be expressed as $\mathbf{U} \cdot \mathbf{W}$ for some $\mathbf{W} \in \mathcal{O}_K^{k \times k}$ corresponding to k linear combinations of the vectors from $\mathbf{X}$.

Consider the ordered set $\{\mathbf{y}_i = \mathbf{X} \cdot \mathbf{W}_i = \mathbf{B} \cdot \mathbf{T}_i\}_{0 \leqslant i < k} \subset \mathcal{P}$. Each $\mathbf{y}_i = \sum_{\kappa=0}^{n-1} \mathbf{t}_i[\kappa] \cdot \mathbf{b}_\kappa$. By construction of $\mathbf{T}$ last $\max(0, k-i-1)$ coordinates of $\mathbf{t}_i$ are zero which implies $\mathbf{y}_i = \sum_{\kappa=0}^{n-k+i} \mathbf{t}_i[\kappa] \cdot \mathbf{b}_\kappa$ for some nonzero $\mathbf{t}_i[n-k+i]$. Such $\{\mathbf{y}_i\}_{0 \leqslant i < k}$ satisfy the statement of the lemma since each $\mathbf{y}_i$ is a $\mathcal{O}_K$-linear combination of exactly $n-k+i$ first vectors of $\mathbf{B}$. □

Now we need a tool to transform k linearly independent vectors of a module $\mathcal{L}$ into a pseudobasis that preserves the algebraic norms of Gram-Schmidt vectors. We use the following lemma for this task.

Lemma 5 ([16, Theorem 4]). *Let $\mathcal{L} \subset K^m$ be a rank-n algebraic lattice. Let $\{\mathbf{s}_i\}_i$ be a full rank set of vectors in $\mathcal{L}$. Then there exists a pseudobasis $(\mathbf{B}, \{\mathfrak{b}_i\})$ of $\mathcal{L}$ such that for all $i < n$: $\mathbf{b}_i \in \mathcal{L}, \mathbf{b}_i \in \mathrm{Span}\{\mathbf{s}_j\}_{j \leqslant i}, \mathbf{b}_i^* = \mathbf{s}_i^*$.*

To prove the main theoretical result of this section we need the following technical lemma.

Lemma 6 ([10, Theorem 1.2.35]). *Let $\mathcal{L} \subset K^m$ be an algebraic lattice of rank n. Let $\mathcal{P}$ be its algebraic sublattice of rank $k \leqslant n$. Then there exist pseudobases $(\mathbf{X}, \{\mathfrak{x}_i\}_{i<n})$ of $\mathcal{L}$ and $((\mathbf{x}_i), \{\mathfrak{d}_i \mathfrak{x}_i\}_i)_{n-k<i<n-1}$ of $\mathcal{P}$ for some $\mathbf{X} \in K^{m \times n}$, fractional ideals $\mathfrak{x}_i$ and integral ideals $\mathfrak{d}_i$ such that:*

$$\mathcal{L} = \bigoplus_{0 \leqslant i < n} \mathfrak{x}_i \cdot \mathbf{x}_i \textit{ and } \mathcal{P} = \bigoplus_{n-k \leqslant j < n} \mathfrak{d}_j \cdot \mathfrak{x}_j \cdot \mathbf{x}_j. \tag{7}$$

The Euclidean norm of a vector cannot increase after an orthogonal projection. Similarly, the algebraic norm cannot increase after the orthogonal projection over $K_\mathbb{R}$ which is stated in the following claim.

Claim 1 *For all vectors $\mathbf{u}, \mathbf{v} \in K_\mathbb{R}^n$ such that $\mathbf{u} \perp \mathbf{v}$ we have $\mathcal{N}(\mathbf{u} + \mathbf{v}) \geqslant \max\{\mathcal{N}(\mathbf{u}), \mathcal{N}(\mathbf{v})\}$. This also implies $\mathcal{N}(\mathbf{v}) \geqslant \mathcal{N}(\pi_\mathbf{w}(\mathbf{v}))$ for all $\mathbf{w} \in K_\mathbb{R}^n$.*

Proof. Consider mutually orthogonal $\mathbf{u}, \mathbf{v} \in K_\mathbb{R}^n$ and construct the matrix $\mathbf{B} \in K_\mathbb{R}^{n \times 2}$ with first column being $\mathbf{u}$ and the second one being $\mathbf{v}$. Perform the QR-factorization $\mathbf{B} = \mathbf{Q} \cdot \mathbf{R}$. Now we have an upper triangular $\mathbf{R} \in K_\mathbb{R}^{2 \times 2}$. Since $\mathbf{u} \perp \mathbf{v}$ we have that $\mathbf{R}$ is also diagonal. In addition $\mathcal{N}(\mathbf{u}) = \mathcal{N}(\mathbf{r}_0)$ and $\mathcal{N}(\mathbf{v}) = \mathcal{N}(\mathbf{r}_1)$ implying $\mathcal{N}(\mathbf{u}+\mathbf{v}) = \mathcal{N}(\mathbf{r}_0 + \mathbf{r}_1)$. The latter is $\mathbf{r}_0[0] \cdot \bar{\mathbf{r}}_0[0] + \mathbf{r}_1[1] \cdot \bar{\mathbf{r}}_1[1] \in K_\mathbb{R}^+$ – a sum of two non-negative real numbers. Hence $\mathcal{N}(\mathbf{r}_0 + \mathbf{r}_1) \geqslant \max\{\mathcal{N}(\mathbf{r}_0), \mathcal{N}(\mathbf{r}_1)\}$ which gives the first part of the claim.

To prove the second part of the claim we rewrite $\mathbf{v} = \pi_\mathbf{w}(\mathbf{v}) + p \cdot \mathbf{w}$ for some $p \in K_\mathbb{R}$. We have $\pi_\mathbf{w}(\mathbf{v}) \perp \mathbf{w}$ which implies $\mathcal{N}(\mathbf{v}) \geqslant \max\{\mathcal{N}(\pi_\mathbf{w}(\mathbf{v})), \mathcal{N}(p \cdot \mathbf{w})\} \geqslant \mathcal{N}(\pi_\mathbf{w}(\mathbf{v}))$. □

For us to proceed we need to connect the Gram-Schmidt vectors of some projective lattice $\mathcal{L}(\mathbf{D}) = \pi_{n-k}(\mathcal{L}(\mathbf{B}))$ with those of $\mathcal{L}(\mathbf{B})$. For this we prove Lemma 7.

Lemma 7. *Let $\mathbf{B} \in K^{m \times n}$ and $\mathbf{U}' \in K^{k \times k}$ be a rank-n and rank-k matrices respectively with $\mathbf{D} = \pi_{n-k}([\mathbf{b}_{n-k}, \ldots, \mathbf{b}_{n-1}]) \cdot \mathbf{U}'$. Let $\mathbf{B}^*, \mathbf{D}^*$ be a matrix of Gram-Schmidt vectors for $\mathbf{B}$ and $\mathbf{D}$ respectively. Suppose also that $\mathbf{d}_\kappa \in \mathrm{Span}\{\mathbf{b}_j^*\}_{n-k \leqslant j \leqslant n-k+\kappa}$ and $\mathbf{d}_\kappa \notin \mathrm{Span}\{\mathbf{b}_j^*\}_{n-k \leqslant j \leqslant n-k+\ell}$ for $\ell < \kappa$. Then $\mathbf{d}_\kappa^* = \mathbf{u}'_\kappa[\kappa] \cdot \mathbf{b}_{n-k+\kappa}^*$.*

Proof. Let $\{\mathbf{d}_\kappa^*\}_{0 \leqslant \kappa < k}$ be the Gram-Schmidt vectors of $\{\mathbf{d}_\kappa\}_{0 \leqslant \kappa < k}$. We have

$$\mathrm{Span}\{\mathbf{d}_\iota^*\}_{\iota < \kappa} = \mathrm{Span}\{\mathbf{d}_\iota\}_{\iota < \kappa} = \mathrm{Span}\{\pi_{n-k}(\mathbf{b}_{n-k+\iota})\}_{\iota < \kappa} = \mathrm{Span}\{\mathbf{b}_{n-k+\iota}^*\}_{\iota < \kappa}$$

Since for $\kappa < k$ we have $\mathbf{d}_\kappa^\star \in \mathrm{Span}\{\mathbf{b}_{n-k+\iota}^*\}_{\iota \leqslant \kappa}$ is an orthogonal projection away from $\mathrm{Span}\{\mathbf{b}_{n-k+\iota}^*\}_{\iota < \kappa}$ it lies in $\mathrm{Span}\{\mathbf{b}_{n-k+\iota}^*\}_{\iota \leqslant \kappa} \cap \mathrm{Span}\{\mathbf{b}_{n-k+\kappa}^*\} = \mathrm{Span}\{\mathbf{b}_{n-k+\kappa}^*\}$. Then we can write $\mathbf{d}_\kappa^* = \mathbf{d}_\kappa - \sum_{\iota=0}^{\kappa-1} \left(\frac{\langle \mathbf{d}_\kappa, \mathbf{d}_\iota^* \rangle}{\langle \mathbf{d}_\iota^*, \mathbf{d}_\iota^* \rangle} \cdot \mathbf{d}_\iota^* \right)$ as

$$\begin{aligned}
&\frac{\langle \mathbf{d}_\kappa - \sum_{\iota=0}^{\kappa-1}\left(\frac{\langle \mathbf{d}_\kappa, \mathbf{d}_\iota^*\rangle}{\langle \mathbf{d}_\iota^*, \mathbf{d}_\iota^*\rangle}\cdot \mathbf{d}_\iota^*\right), \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^* = \\
&\frac{\langle \mathbf{d}_\kappa, \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^* = && \text{Since } \langle \mathbf{d}_\iota^*, \mathbf{b}_{n-k+\kappa}^*\rangle = 0,\ 0 \le \iota \le \kappa - 1 \\
&\frac{\langle \sum_{i=0}^{k} \mathbf{u}'_\kappa[i]\pi_{n-k}(\mathbf{b}_{n-k+i}), \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^* = \\
&\frac{\langle \sum_{i=0}^{k}\left(\mathbf{u}'_\kappa[i]\mathbf{b}_{n-k+i} - \mathbf{u}'_\kappa[i]\left(\sum_{j=0}^{n-k-1}\frac{\langle \mathbf{b}_{n-k+i}, \mathbf{b}_j^*\rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^*\rangle}\cdot \mathbf{b}_j^*\right)\right), \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^* = \\
&\frac{\langle \sum_{i=0}^{k} \mathbf{u}'_\kappa[i]\mathbf{b}_{n-k+i}, \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^*. && \text{Since } \langle \mathbf{d}_\iota^*, \mathbf{b}_{n-k+\kappa}^*\rangle = 0,\ 0 \le \iota \le \kappa - 1
\end{aligned}$$

Overall,

$$\mathbf{d}_\kappa^* = \frac{\langle \sum_{i=0}^{k} \mathbf{u}'_\kappa[i]\mathbf{b}_{n-k+i}, \mathbf{b}_{n-k+\kappa}^*\rangle}{\langle \mathbf{b}_{n-k+\kappa}^*, \mathbf{b}_{n-k+\kappa}^*\rangle}\cdot \mathbf{b}_{n-k+\kappa}^*. \tag{8}$$

Since $\langle \mathbf{u}'_\kappa[i]\mathbf{b}_{n-k+i}, \mathbf{b}_{n-k+\kappa}^*\rangle$ is non-zero only for $i = \kappa$, Equation (8) implies $\mathbf{d}_\kappa^* = \mathbf{u}'_\kappa[\kappa]\cdot \mathbf{b}_{n-k+\kappa}^*$. □

The following result is an analogue of [28, Lemma 1] generalized to the setting of algebraic lattices. In the classical setting it shows that the determinant of any rank-k lattice $\mathcal{P}$ of a lattice $\mathcal{L}$ cannot exceed the product of k least norms of Gram-Schmidt vectors of *any* basis of $\mathcal{L}$. To prove the analogous result for the case when $\mathcal{P}$ and $\mathcal{L}$ are algebraic, we first consider an arbitrary submodule $\mathcal{P} \subset \mathcal{L}$ and construct its overlattice $\mathcal{L}'$ that is "primitive", that is all ideals of the pseudobasis of $\mathcal{L}'$ satisfy $\mathfrak{d}_j = \mathcal{O}_K$ for $n-k \leqslant j < n$ in the context of Lemma 6. This is done to reduce an amount of the ideals considered during the proof to simplify it. We then consider the projection $\pi_{n-k}(\mathcal{L}')$ with respect to a fixed basis of $\mathcal{L}$ and deduce relations between the pseudobases of each member of the chain $\mathcal{L} \supseteq \mathcal{L}' \supseteq \mathcal{P}$ and $\pi_{n-k}(\mathcal{L}')$. These inclusions enable us to argue on the lower bound on $\mathcal{N}(\det \mathcal{P})$.

Theorem 1. *Let* $(\mathbf{B}, \{\mathfrak{b}_i\}_i)$ *be a pseudobasis of an algebraic lattice* $\mathcal{L}$ *and* $\mathbf{B}^*$ *its Gram-Schmidt vectors. Let* $\mathcal{P}$ *be a rank* k *algebraic sublattice of* $\mathcal{L}$*. Then*

$$\mathcal{N}(\det \mathcal{P}) \geqslant \min_{\substack{J \subset \{0,\ldots,n-1\} \\ |J| = k}} \prod_{k \in J} \mathcal{N}(\mathbf{b}_j^*)\cdot \mathcal{N}(\mathfrak{b}_i).$$

Proof. For $\mathcal{L}$ and its submodule $\mathcal{P}$, there exist pseudobases $(\mathbf{X}, \{\mathfrak{x}_i\}_{i<n})$ of $\mathcal{L}$ and $((\mathbf{x}_j)_j, \{\mathfrak{d}_j \cdot \mathfrak{x}_j\}_j)_{n-k\leqslant j<n}$ of $\mathcal{P}$ as in Lemma 6. Without loss of generality we can assume that all $\mathfrak{x}_i$ are integral. Since all $\mathfrak{d}_j \subset \mathcal{O}_K$, we have

$$\mathcal{P} \subset \bigoplus_{n-k\leqslant j<n} \mathfrak{x}_j \cdot \mathbf{x}_i := \mathcal{L}'.$$

The latter is a sublattice of $\mathcal{L}$ and an overlattice for $\mathcal{P}$, so proving the statement for such $\mathcal{L}'$ suffices.

Invoking Lemma 4 on $\mathcal{L}$ and its sublattice $\mathcal{L}'$, we obtain a set $\{\mathbf{v}_\kappa\}_{\kappa<k}$ of k linearly independent vectors of $\mathcal{L}'$ such that $\mathbf{v}_\kappa \in \operatorname{Span}\{\mathbf{b}_i\}_{0\leqslant i\leqslant n-k+\kappa}$ and $\mathbf{v}_\kappa \notin \operatorname{Span}\{\mathbf{b}_i\}_{0\leqslant i\leqslant n-k+\kappa-1}$ for all $0 \leqslant \kappa < k$.

We then apply Lemma 5 to $\mathcal{L}'$ and $\{\mathbf{v}_\kappa\}_{\kappa<k}$ to obtain a pseudobasis $(\mathbf{C}, (\mathfrak{c}_\kappa))$ of $\mathcal{L}'$ such that

$$\mathbf{c}_\kappa \in \operatorname{Span}\{\mathbf{v}_j\}_{j\leqslant\kappa} \subseteq \operatorname{Span}\{\mathbf{b}_j\}_{j\leqslant n-k+\kappa};\ \mathbf{c}_\kappa \notin \operatorname{Span}\{\mathbf{v}_j\}_{j<\kappa} \subseteq \operatorname{Span}\{\mathbf{b}_j\}_{j<n-k+\kappa} \tag{9}$$

Since both $\mathbf{X}$ and $\mathbf{B}$ define the same module $\mathcal{L}$, we have $\mathbf{X} = \mathbf{B}\cdot\mathbf{W}$ for some $\mathbf{W} \in K^{n\times n}$. Decompose $\mathbf{W} = [\mathbf{W}_L \mid \mathbf{W}_R]$, where $\mathbf{W}_R \in K^{n\times k}$ consists of the last k columns of $\mathbf{W}$. The last k columns of $\mathbf{X}$, denoted by $\mathbf{X}_R$ can be now expressed as $\mathbf{X}_R = \mathbf{B}\cdot\mathbf{W}_R$, hence $((\mathbf{X}_R, \mathfrak{x}_{n-k+\kappa})_{\kappa<k})$ is a pseudobasis of $\mathcal{L}'$.

Similarly, consider $\mathbf{M} \in K^{k\times k}$ such that $\mathbf{X}_R \cdot \mathbf{M} = \mathbf{C}$. Since $(\mathbf{C}, \{\mathfrak{c}_\kappa\}_\kappa)$ and $(\mathbf{X}_R, (\mathfrak{x}_{n-k+\kappa}))$ are pseudobases of the same lattice $\mathcal{L}'$, we have that $\mathbf{M}$ is a transformation matrix and hence $\mathbf{m}_i[\ell] \in \mathfrak{x}_\ell \cdot \mathfrak{c}_i^{-1}$ by Proposition 1. It holds that

$$\mathbf{B}\cdot\mathbf{W}_R\cdot\mathbf{M} = \mathbf{X}_R\cdot\mathbf{M} = \mathbf{C}. \tag{10}$$

Consider the entries $\mathbf{u}_i[j]$ of $\mathbf{U} := \mathbf{W}_R\cdot\mathbf{M} \in K^{n\times k}$. We want to show that $\mathbf{u}_i[j] \in \mathfrak{b}_j\mathfrak{c}_i^{-1}$. Indeed, we have that $\mathbf{u}_i[j] = \sum_{\ell=0}^{k-1}\mathbf{w}_\ell[j]\cdot\mathbf{m}_i[\ell]$. For $\ell < n$ the fact that $\mathbf{W}$ is a transformation matrix implies that we have inclusions $\mathbf{w}_\ell[j] \in \mathfrak{b}_j\cdot\mathfrak{x}_\ell^{-1}$ and $\mathbf{m}_i[\ell] \in \mathfrak{x}_\ell\cdot\mathfrak{c}_i^{-1}$, again by Proposition 1. From here, $\mathbf{w}_\ell[j]\cdot\mathbf{m}_i[\ell] \in \mathfrak{b}_j\cdot\mathfrak{c}_i^{-1}$ for $0 \leqslant i < k$ and $0 \leqslant j < n$. Hence,

$$\mathbf{u}_i[j] \in \mathfrak{b}_j\mathfrak{c}_i^{-1} \quad \text{for all } 0 \leqslant i < \kappa, 0 \leqslant j < n. \tag{11}$$

Thanks to Equation (10), we have that $\mathbf{c}_\kappa = \sum_{j=0}^{n-1}\mathbf{u}_\kappa[j]\cdot\mathbf{b}_j$ for the aforementioned $\mathbf{U} \in K^{n\times\kappa}$. For $j > \kappa$ the value of $\mathbf{u}_\kappa[j]\cdot\mathbf{b}_j$ is zero since otherwise the corresponding $\mathbf{c}_\kappa$ would not be in $\operatorname{Span}\{\mathbf{v}_j\}_{j\leqslant\kappa}$. Thus, we can rewrite

$$\mathbf{c}_\kappa = \sum_{j=0}^{n-k+\kappa}\mathbf{u}_\kappa[j]\cdot\mathbf{b}_j \quad \text{for} \quad \mathbf{u}_\kappa[j] \in \mathfrak{b}_j\cdot\mathfrak{c}_\kappa^{-1} \quad \text{and} \quad \mathbf{u}_{n-k+\kappa}[j] \neq 0,$$

where the last condition is due to $\mathbf{c}_\kappa \notin \operatorname{Span}\{\mathbf{v}_j\}_{j<\kappa}$.

Now let us consider the projected lattice $\pi_{n-k}(\mathcal{L}')$, where π_{n-k} projects orthogonally to $\operatorname{Span}\{\mathbf{b}_i^*\}_{i\leqslant n-k-1}$. Then for $0 \leqslant \kappa < k$ the pseudobasis of $\pi_{n-k}(\mathcal{L}')$ is given by $(\mathbf{D}, (\mathfrak{c}_\kappa)) = ((\pi_{n-k}(\mathbf{c}_\kappa))_\kappa, (\mathfrak{c}_\kappa))$. Since the projections $\pi_{n-k}(\mathbf{b}_i)$ are zero for $i < n-k$ we can write $\mathbf{D}$ as $\pi_{n-k}([\mathbf{b}_{n-k}, \ldots, \mathbf{b}_{n-1}])\cdot\mathbf{U}'$ where $\mathbf{U}' \in K^{k\times k}$

consists of last k rows of $\mathbf{U}$. Hence $(\pi_{n-k}(\mathbf{c}_\kappa))_\kappa$ can be explicitly written as:

$$\begin{aligned}
&\mathbf{d}_0 = \mathbf{u}'_0[n-k] \cdot \pi_{n-k}(\mathbf{b}_{n-k}),\\
&\mathbf{d}_1 = \mathbf{u}'_1[n-k] \cdot \pi_{n-k}(\mathbf{b}_{n-k}) + \mathbf{u}_1[n-k+1] \cdot \pi_{n-k}(\mathbf{b}_{n-k+1}),\\
&\quad\vdots\\
&\mathbf{d}_{k-1} = \sum_{j=0}^{k-1} \mathbf{u}'_{k-1}[n-k+j] \cdot \pi_{n-k}(\mathbf{b}_{n-k+j})
\end{aligned}$$

for $\mathbf{d}_\kappa$ being the columns of $\mathbf{D}$ From these last equations it follows that

$$\mathbf{d}^*_\kappa \in \operatorname{Span} \pi_{n-k}\{\mathbf{b}^*_i\}_{n-k \leqslant i \leqslant n-k+\kappa} = \operatorname{Span}\{\mathbf{b}^*_{n-k+\iota}\}_{0 \leqslant \iota \leqslant \kappa} \tag{12}$$

$$\mathbf{d}^*_\kappa \notin \operatorname{Span} \pi_{n-k}\{\mathbf{b}^*_i\}_{n-k \leqslant i \leqslant n-k-1} = \operatorname{Span}\{\mathbf{b}^*_{n-k+\iota}\}_{0 \leqslant \iota \leqslant \kappa-1} \tag{13}$$

Thus, the rank of $\mathbf{U}'$ is k. Hence, we can apply Lemma 7 to $\mathbf{B} \in K^{m\times n}, \mathbf{U}' \in K^{k\times k}$, and $\mathbf{D}$, which yields

$$\mathbf{d}^*_\kappa = \mathbf{u}'_\kappa[\kappa] \cdot \mathbf{b}^*_{n-k+\kappa}. \tag{14}$$

Next we want to prove that $\forall \kappa < k : \mathcal{N}(\mathbf{c}^*_\kappa) \geqslant \mathcal{N}(\mathbf{d}^*_\kappa)$ Notice that

$$\mathbf{c}^*_\kappa, \mathbf{d}^*_\kappa \in \operatorname{Span}\{\mathbf{b}^*_j\}_{0 \leqslant j \leqslant n-k+\kappa} \ \forall \kappa < k, \tag{15}$$

where first inclusion is due to Equation (9) and the second is by Equation (12). By definition, $\mathbf{d}_\kappa = \pi_{n-k}(\mathbf{c}_\kappa)$. Due to Equations (12) to (15)

$$\mathbf{u}'_\kappa[\kappa] \cdot \mathbf{b}^*_{n-k+\kappa} = \mathbf{d}^*_\kappa = \pi_{n-k+\kappa}(\mathbf{d}_\kappa) = \pi_{n-k+\kappa}(\pi_{n-k}(\mathbf{c}_\kappa)) = \pi_{n-k+\kappa}(\mathbf{c}_\kappa).$$

The latter is $\pi_{n-k+\kappa}(\mathbf{c}^*_\kappa) + \pi_{n-k+\kappa}\left(\sum_{\iota=0}^{\kappa-1} \frac{\langle \mathbf{c}_\kappa, \mathbf{c}^*_\iota\rangle}{\langle \mathbf{c}^*_\iota, \mathbf{c}^*_\iota\rangle} \cdot \mathbf{c}^*_\iota\right)$ where the second summand is zero by Equation (15). Thus, $\mathbf{c}^*_\kappa = \pi_{n-k+\kappa}(\mathbf{c}^*_\kappa) + \mathbf{w}$, where $\mathbf{w} \in \operatorname{Span}\{\mathbf{b}^*_i\}_{0 \leqslant i < n-k+\kappa}$. Claim 1 applied to $\mathbf{c}^*_\kappa = \mathbf{d}^*_\kappa + \mathbf{w}$ gives us $\mathcal{N}(\mathbf{c}^*_\kappa) \geqslant \mathcal{N}(\mathbf{d}^*_\kappa)$.

Combining the definition of $\mathcal{L}' = \mathcal{L}(\mathbf{D}, \{\mathfrak{c}_\kappa\}_{\kappa<k})$ and Equation (14) we get

$$\det \pi_{n-k}(\mathcal{L}') = \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{d}^*_\kappa \cdot \mathfrak{c}_\kappa) = \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{b}^*_{n-k+\kappa} \cdot \mathbf{u}'_\kappa[\kappa] \cdot \mathfrak{c}_\kappa).$$

Recall that all $\mathbf{u}_\kappa[n-k+\kappa] \in \mathfrak{b}_{n-k+\kappa} \cdot \mathfrak{c}_\kappa^{-1}$ as shown in Equation (11). Then

$$\prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{b}^*_{n-k+\kappa} \cdot \mathbf{u}_\kappa[n-k+\kappa] \cdot \mathfrak{c}_\kappa) \geqslant \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{b}^*_{n-k+\kappa}) \cdot \mathcal{N}(\mathfrak{b}_{n-k+\kappa} \cdot \mathfrak{c}_\kappa^{-1} \cdot \mathfrak{c}_\kappa);$$

$$\det \pi_{n-k}(\mathcal{L}') \geqslant \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{b}^*_{n-k+\kappa}) \cdot \mathcal{N}(\mathfrak{b}_{n-k+\kappa}) \geqslant \min_{\substack{J \subset \{0,\dots,n-1\}\\ |J|=k}} \prod_{j\in J} \mathcal{N}(\mathbf{b}^*_j) \cdot \mathcal{N}(\mathfrak{b}_j).$$

This implies $\det(\mathcal{L}') = \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{c}_\kappa^*)\cdot\mathcal{N}(\mathfrak{c}_\kappa) \geqslant \prod_{\kappa=0}^{k-1} \mathcal{N}(\mathbf{d}_\kappa^*)\cdot\mathcal{N}(\mathfrak{c}_\kappa) = \det \pi_{n-k}(\mathcal{L}')$. Hence $\det \mathcal{L}' \geqslant \det \pi_{n-k}(\mathcal{L}')$. Summing up, the following chain of inequalities proves the result:

$$\det \mathcal{P} \geqslant \det \mathcal{L}' \geqslant \det(\pi_{n-k}(\mathcal{L}')) \geqslant \min_{\substack{J \subset \{0,\ldots,n-1\} \\ |J|=k}} \prod_{k\in J} \mathcal{N}(\mathbf{b}_j^*)\cdot\mathcal{N}(\mathfrak{b}_i).$$

□

3.1 Descending to Subfields

The problem of finding short vectors (in the algebraic norm) in a rank-2 module over a number field K can be reduced to the problem of finding a short vector in a rank-$2d'$ module defined over a subfield $L \subset K$ with $[K : L] = d'$ and $\zeta \in L$ such that $K = L[\zeta]$. We would make use of this method while studying the behaviour of our algebraic lattice reduction algorithm on the NTRU modules defined in Section 2.2.

Let K and $L \subset K$ be two number fields of a relative degree $d' = [K : L]$. Concretely, we have:

$$\mathcal{O}_K = \mathcal{O}_L \oplus \zeta\mathcal{O}_L \oplus \ldots \oplus \zeta^{d'-1}\mathcal{O}_L. \tag{16}$$

As a consequence, the module $M = \mathbf{b}_0 \cdot \mathcal{O}_K + \mathbf{b}_1 \cdot \mathcal{O}_K$ decomposes over $\mathcal{O}_L$ as:

$$\Big(\mathbf{b}_0 \cdot \mathcal{O}_L \oplus \zeta\mathbf{b}_0 \cdot \mathcal{O}_L \oplus \ldots \oplus \zeta^{d'-1}\mathbf{b}_0 \cdot \mathcal{O}_K\Big) \oplus$$
$$\Big(\mathbf{b}_1 \cdot \mathcal{O}_L \oplus \zeta\mathbf{b}_1 \cdot \mathcal{O}_L \oplus \ldots \oplus \zeta^{d'-1}\mathbf{b}_1 \cdot \mathcal{O}_K\Big),$$

yielding a basis of M viewed as a free $\mathcal{O}_L$-module of rank $2d'$. We refer to this process as descending (into the subfield L).

In order to descend from fields and their subfields we define the descend procedure. It represents a rank-n $\mathcal{O}_K$ module given by matrix $\mathbf{B} \in K^{m\times n}$ as a rank-$(n \cdot d')$ module over subfield L with a relative degree $[K : L]$. It returns the $md' \times nd'$ matrix over L representing the basis of the initial module over the subfield L up to a certain permutation of the coordinates. To perform the opposite operation, called ascend, we proceed as follows. We combine the coefficients of a $2d'$-dimensional vector $\mathbf{v} \in \mathcal{O}_L^{2\cdot d'}$ into a 2-dimensional vector $(\sum_{i<d'} \zeta^i \cdot (\mathbf{v})_i, (\sum_{i<d'} \zeta^i \cdot (\mathbf{v})_{d'+i})) \in M \subset K^2$.

Formally, we define the following:

1. The $\mathsf{Descend}_{\mathsf{m}}(L, \mathbf{B})$ (where the subscript m stands for "matrix") function represents rank-n $\mathcal{O}_K$ module given by matrix $\mathbf{B} \in K^{m\times n}$ as a rank $n \cdot d'$ module over subfield L with a relative degree $[K : L]$. It returns the $md' \times nd'$ matrix over L representing the basis of the initial module over the subfield L up to a certain permutation of the coordinates.
2. The $\mathsf{Ascend}_{\mathsf{v}}(K, \mathbf{v})$ (where the subscript v stands for "vector") function represents $d' \cdot n$ dimensional vector $\mathbf{v}$ over field L as n-dimensional vector over K for the small degree $[K : L]$.

3. The $\mathsf{Ascend}_{\mathsf{m}}(K, \mathbf{B})$ function receives the $m' \times nd'$ matrix B over field L (or, equivalently, a list of $n' \in \mathbb{N}$ vectors of dimension md' over L) and applies $\mathsf{Ascend}_{\mathsf{v}}(K, \mathbf{v})$ for every vector in it.

Invoking, e.g., a BKZ reduction algorithm with the same blocksize on a lattice of halved dimension is significantly faster.

However, descending into subfields has limitations. In particular, finding a small norm element in a subfield, does not guarantee that its ascend will preserve relative smallness. Concretely, let L be a subfield of K and $\mathbf{v} \in L^{[K:L]}$ be short. We ascend $\mathbf{v}$ to K obtaining an element $v \in K$. The relation between the algebraic norm (over K) of v and the algebraic norm (over L) of the corresponding vector $\mathbf{v}$ is given by the inequality from Lemma 1. Thus, minimizing $\mathcal{N}_{L/\mathbb{Q}}(\mathbf{v})$ does not necessarily entail minimizing $\mathcal{N}_{K/\mathbb{Q}}(\mathbf{v})$. Vectors $\mathbf{v}$ that are the shortest possible in the algebraic norm, will correspond to somewhat short elements v, but not necessarily *the shortest possible.*

3.2 Predicting DSD Events

All NTRU modules always admit a free basis. Let $\mathbf{B}_{\text{NTRU}} \in K^{2\times 2}$ be an $\mathcal{O}_K$-basis of an NTRU module defined over a power-of-two cyclotomic field $L \subset K$. As discussed in Section 3.1 we can descend a basis of any free module to some subfield of K. Let $\mathbf{B}$ be a basis of an NTRU module of rank $n = 2 \cdot [K : L]$ over a number field $L \subset K$ for which one has a small index $n' = [K : L]$. We would like to deduce how the log-profile $\mathbf{p} := (\log(\mathcal{N}(r_{i,i}) \cdot \mathcal{N}(\mathfrak{b}_i)))_{i<n}$ of an LLL reduced basis of $\mathcal{L}(\mathbf{B})$ will look like. We heuristically assume that the α-LLL reduced basis of $\mathcal{L}(\mathbf{B})$ admits a profile of a special form. There are two horizontal regions at the beginning and the end of the profile and a central line connecting those two flat regions. The slope of this line is controlled by α. We introduce an algebraic analogue of Heuristic 1 as follows.

Heuristic 3 (AZGSA) *Let L be a number field of degree d. Let $\mathbf{B}$ be an α-LLL reduced $\mathcal{O}_L$ basis of a rank-n NTRU module for some $\alpha > 0$. Let $\mathbf{p}$ be a log-profile of $\mathbf{B}$. Then we have:*

$$\mathbf{p}_i = \begin{cases} d \log q, & i \leqslant n/2 - n' \\ d \log q \cdot (1 - \frac{i - n/2 + n'}{2n'}), & n/2 - n' < i < n/2 + n' - 1 \\ 0, & i \geqslant n/2 + n' - 1, \end{cases} \tag{17}$$

for $n' = 1/2 + d \log d / \log \alpha$.

For fixed n, d, the larger $\log q$ is, the greater $\min_{J \subset \{0,\ldots,n-1\}:|J|=n/2} \sum_{j \in J} \mathbf{p}_j$ is. For sufficiently large $\log q$ the AZGSA would contradict Theorem 1. This observation allows us to introduce the algebraic Dense Submodule Discovery event as follows.

Definition 6 (Algebraic DSD event). *Let $B \in \mathcal{O}_L^{n\times n}$ be an $\mathcal{O}_L$-basis for some NTRU module for an even $n \in \mathbb{N}$. We define the DSD as an event* *when an algebraic lattice reduction algorithm called on* $\mathbf{B}$ *returns a pseudobasis* $([\mathbf{M}|\mathbf{B}'], \{\mathfrak{m}_i\}_{i<n/2} \cup \{\mathfrak{b}'_i\}_{n/2 \leqslant i < n})$ *for* $\mathbf{B}', \mathbf{M} \in \mathcal{O}_L^{n\times(n/2)}$ *and a module spanned by* $(\mathbf{M}, \{\mathfrak{m}_i\}_{i<n/2})$ *contains the secret vector* $(\mathbf{f}, \mathbf{g})$*.*

As in the classical case we postulate that for reasonably large modulus q the DSD event occurs meaning that we find a rank-$(n/2)$ submodule that contains $(\mathbf{f}, \mathbf{g})$. For simplicity we introduce the heuristic predicting the conditions for the algebraic LLL algorithm to trigger the DSD event assuming that the output of the algorithm is a free basis meaning all $\mathfrak{m}_i = \mathfrak{b}_j = \mathcal{O}_K$.

Heuristic 4 (Condition for algebraic DSD event) *Let* $\mathbf{B}$ *be an* α*-LLL reduced basis of a rank-*n *algebraic NTRU module over a power-of-2 cyclotomic field* L *for some* $\alpha > 0$*. Then* $\mathbf{B}$ *contains a basis of a dense rank-*$(n/2)$ *sublattice* $\mathcal{L}'$ *containing* $(\mathbf{f}, \mathbf{g})$ *as soon as:*

$$\log \mathcal{N}(\det(\mathcal{L}')) < \left(\frac{n'-1}{2}\right) \log q^{\deg L} - \frac{(n'-1)^2}{2} \log \alpha \tag{18}$$

for $n' = 1/2 + d \log d / \log \alpha$*.*

In order to make an algebraic DSD event estimator, we rewrite Equation (18) as $\log q > \frac{2\log \mathcal{N}(\det(\mathcal{L})) + (n'-1)^2 \log\alpha}{(n'-1)\deg L}$. This view enables us to say, for given NTRU parameters d, n and a parameter α, for which $\log q$ the DSD event occurs.

4 Algebraic LLL

To confirm our theoretical result on the DSD event prediction we implement a variant of an algebraic LLL from [24] restricted to the case of free bases (that is all ideals of a pseudobasis are $\mathcal{O}_K$).

4.1 Size reduction

Following [24, Definition 3.5], we define a size reduced basis for the case of power-of-2 cyclotomic fields.

Definition 7 (Size reduction). *Let* K *be a power-of-2 cyclotomic field of degree* d*. A free basis* $\mathbf{B} = [\mathbf{b}_i]_{i<n} \in K^{m\times n}$ *is said to be size reduced if for all* $i > 0$ *and* $j < i$*:*

$$||r_{i,j}/r_{j,j}|| \leqslant d/2, \tag{19}$$

where $r_{i,j}$ are defined as in Equation (2).

The size reduction process given in Algorithm 4.1 is an adaptation of the L2 reduction from [27] to the algebraic setting. At the i-th iteration, size reduction considers the i-th basis vector and subtracts $\lfloor \mu_{i,j} \rceil \cdot \mathbf{b}_j$ from it for all $j = i-1, \ldots, 0$ for $\lfloor \mu_{i,j} \rceil$ – the $\mathcal{O}_K$ rounding of $\mu_{i,j}$ as defined in Section 2. The rounding procedure is crucial because we can only subtract an $\mathcal{O}_K$-multiple of a basis vector from a given one in order to preserve the lattice. The routine is implemented in Algorithm 4.1.

Following [27], Algorithm 4.1 does not operate on the basis vectors directly. Instead, it updates a Gram matrix $\mathbf{G} = \mathbf{B}^\dagger \cdot \mathbf{B}$, which can be computed exactly, thus improving the accuracy of μ's and r's. The algorithm returns a transformation matrix $\mathbf{U}$ such that $\mathbf{B} \cdot \mathbf{U}$ is size reduced. As in Algorithm 2.1, we might be interested in size reducing not the whole basis but some if its vectors.

```
Algorithm 4.1 size_reduce
Input:  $\mathbf{U} \in K^{n \times n}$ – an unimodular matrix
        $\mathbf{G} \in K^{n \times n}$ – Gram matrix
        $\{\mu_{i,j}\}_{0 \leqslant i,j < n}, \{r_{i,j}\}_{0 \leqslant i < n, \forall j}$ – Gram Schmidt coefficients;
        $s, e$ – positions to start at and to end size reduction before.
Output: unimodular transformation $\mathbf{U}$ such that $\mathbf{B} \cdot \mathbf{U}$ is size reduced;
        $\mathbf{G} = \mathbf{B}^\dagger \cdot \mathbf{B}$ - corresponding Gram matrix;
        $\{\mu_{i,j}\}_{0 \leqslant i,j \leqslant e}, \{r_{i,j}\}_{0 \leqslant i \leqslant e, \forall j}$ – Gram Schmidt coefficients of $\mathbf{B} \cdot \mathbf{U}$
1: for $s \leqslant i < e$ do
2:     for $j := i-1, \ldots, 0$ do
3:         $\delta := \lfloor \mu_{i,j} \rceil$
4:         $\mu_{i,l} := \mu_{i,l} - \delta$ for $l \leqslant j$
5:         $r_{i,l} := r_{i,l} - \delta \cdot r_{l,l}$ for $l \leqslant j$
6:         $\mathbf{G}_{i,i} := \mathbf{G}_{i,i} - \delta \overline{\mathbf{G}}_{i,j} - \bar{\delta} \mathbf{G}_{i,j} + \delta \bar{\delta} \mathbf{G}_{j,j}$
7:         $G_{i,\ell} := \mathbf{G}_{i,\ell} - \mathbf{G}_{j,\ell} \cdot \delta$ for $\ell < i$; $\mathbf{G}_{\kappa,i} := \mathbf{G}_{\kappa,i} - \mathbf{G}_{\kappa,j} \cdot \bar{\delta}$ for $i < \kappa < n$
8:         $\mathbf{u}_i := \mathbf{u}_i - \delta \cdot \mathbf{u}_j$
9: return $\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}_{0 \leqslant i,j \leqslant e}, \{r_{i,j}\}_{0 \leqslant i \leqslant e, \forall j}$
```

Theorem 2. *On input $\mathbf{G} = \mathbf{B}^\dagger \cdot \mathbf{B}$ and $\{\mu_{i,j}\}_{0 \leqslant i,j < n}, \{r_{i,j}\}_{0 \leqslant i < n, \forall j}$, Algorithm 4.1 returns a transformation matrix $\mathbf{U}$ such that $\mathbf{B} \cdot \mathbf{U}$ is size reduced. It also returns the Gram matrix of $\mathbf{B} \cdot \mathbf{U}$, together with $\{\mu_{i,j}\}_{0 \leqslant i,j < n}, \{r_{i,j}\}_{0 \leqslant i < n, \forall j}$ in time polynomial in the bitsize of $\mathbf{B}$, rank n and $\log |\Delta_K|$.*

Proof. Let $\mathbf{C} = \mathbf{B} \cdot \mathbf{U}$ denote the basis after the execution of Algorithm 4.1. The matrix $\mathbf{U}$ is unimodular by construction: it is upper triangular with 1-s on the diagonal as its columns $\mathbf{u}_i$'s get updated only with $\mathbf{u}_j$ for $j < i$ during the execution of the algorithm. Thus, $\mathbf{C}$ is a basis of the same module.

In our considered case of power-of-2 cyclotomic fields, we have that the Euclidean norm of $\mu_{i,j} - \lfloor \mu_{i,j} \rceil$ under the coefficient embedding is no larger than $\sqrt{d}/2$, and, hence, under the canonical embedding that scales every vector by a factor of $\sqrt{d}$, it is no larger than $d/2$. This shows that the output values $\mu_{i,j}$ and $r_{i,j}$ satisfy the size reduction condition.

In Line 6 we update $r_{i,j} := r_{i,j} - \lfloor \mu_{i,j} \rceil \cdot r_{j,j}$ accordingly. Thus, each iteration of the inner loop corresponds to the following transformation on the input basis: $\mathbf{b}_i := \mathbf{b}_i - \delta \mathbf{b}_j$. What remains is to update the Gram matrix $\mathbf{G}$. Recall that $\mathbf{G}_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle$ and $\mathbf{G}_{j,i} = \overline{\mathbf{G}_{i,j}}$. After the above transformation, $\langle \mathbf{b}_i, \mathbf{b}_i \rangle$ becomes

$\langle \mathbf{b}_i , \mathbf{b}_i - \delta \mathbf{b}_j \rangle - \delta \langle \mathbf{b}_j , \mathbf{b}_i - \delta \mathbf{b}_j \rangle = \langle \mathbf{b}_i , \mathbf{b}_j \rangle - \bar{\delta} \langle \mathbf{b}_i , \mathbf{b}_j \rangle - \delta \langle \mathbf{b}_j , \mathbf{b}_i \rangle - \delta\bar{\delta} \langle \mathbf{b}_j , \mathbf{b}_j \rangle = \mathbf{G}_{i,j} - \bar{\delta} \cdot \mathbf{G}_{i,j} - \delta \cdot \overline{\mathbf{G}}_{i,j} - \delta\bar{\delta} \cdot \mathbf{G}_{j,j}$. This validates Line 7 of Algorithm 4.1. Similarly, the inner product $\langle \mathbf{b}_i , \mathbf{b}_j \rangle$ becomes $\langle \mathbf{b}_i , \mathbf{b}_j \rangle - \delta \langle \mathbf{b}_j , \mathbf{b}_j \rangle = \mathbf{G}_{i,j} - \bar{\delta} \mathbf{G}_{j,j}$ for $i \neq j$. In particular, the change of the i-th basis vector modifies $\mathbf{G}_{\kappa,i}$ and $\mathbf{G}_{\ell,i}$ for $i < \kappa < n, 0 \leq \ell < i$. These operations (Lines 7-8) as well as the update of $\mathbf{u}_i$ (Line 9) are performed in time polynomial in the bitsize of $\mathbf{B}$, n, d. □

4.2 Unit reduction

In order to shorten projected basis vectors in their Euclidean norm, e.g., reduce the diagonal elements $r_{i,i}$, we can multiply $r_{i,i}$'s by a unit. This step is specific for the algebraic lattice reduction because in the classical case for lattices over $\mathbb{Q}$, the only units are ± 1 and multiplying by any of them does not change the Euclidean norm of a vector.

Definition 8 (Unit reduction). *A free basis $\mathbf{B} = [\mathbf{b}_i]_{i<n} \in K^{m \times n}$ is said to be unit reduced if for all $0 \leqslant i < n$:*

$$||r_{i,i}||^{1/2} \leqslant 2^{O(f \log f)} \mathcal{N}(r_{i,i})^{1/d}, \tag{20}$$

where f is conductor of K and $d = \varphi(f)$ is the degree of K.

Algorithm 4.2 implements unit reduction. In particular, $r_{i,i}$ is replaced by $ur_{i,i}$ such that $||ur_{i,i}|| \leqslant ||r_{i,i}||$. This is done by mapping $r_{i,i}^{1/2}$ to the Log-unit lattice [5] and solving the CVP problem with respect to the obtained vector in that lattice. The routine implemented in Algorithm 4.2 follows the paradigm of updating the transformation matrix and the Gram matrix (instead of the basis). It returns a unimodular transformation matrix $\mathbf{U}$ and the updated Gram matrix $\mathbf{G}$. We might be interested in reducing only some of the $r_{i,i}$'s, hence we specify their indices in the input to the algorithm.

To argue on the complexity of Algorithm 4.2, we need the following lemma, which is an adaptation of [19, Theorem 1].

Lemma 8 (Adaptation of [19, Theorem 1]). *Let K be a cyclotomic field with conductor f and degree d. Let k be an $K_{\mathbb{R}}$ element. We can find an $\mathcal{O}_K$-unit u such that: $||u \cdot k|| \leqslant 2^{O(f \log f)} \mathcal{N}(k)^{1/d}$ in time polynomial in $\log |\Delta_K|$ and the bitsize of k.*

The approach proposed by Kirchner, Espitau and Fourque is similar to the one described in [12]. Using the log-embedding we can map k to the log-unit hyperplane and solve CVP on the log-unit lattice. Dividing k by the found unit gives the result. The next theorem that concludes on the runtime of Algorithm 4.2, follows from Lemma 8.

Theorem 3. *On input the Gram matrix $\mathbf{G}$ of a basis $\mathbf{B}$, and $\{r_{i,j}\}_{0 \leqslant i < n, \forall j}$, Algorithm 4.2 returns a transformation matrix $\mathbf{U}$ such that $\mathbf{BU}$ is unit reduced, and a Gram matrix of this new basis. The algorithm terminates in time polynomial in the bitsize of $\mathbf{B}$, $\log |\Delta_K|$, and n.*

[5] Note that one can map $r_{i,i}$ to the Log-unit lattice's span and then divide all coordinates by 2, which is equivalent to taking the square root.

Algorithm 4.2 unit_reduce

```
Input:  $\mathbf{U} \in K^{n\times n}$ – an unimodular matrix
        $\mathbf{G} \in K^{n\times n}$ – Gram matrix
        $\{\mu_{i,j}\}_{0\leqslant i,j\leqslant e}, \{r_{i,j}\}_{0\leqslant i<n}$ – Gram Schmidt coefficients of a lattice;
        $s, e$ – start and end indices.
Output: unimodular transformation $\mathbf{U}$ such that $\mathbf{B}\cdot\mathbf{U}$ is unit reduced;
        $\mathbf{G} = \mathbf{B}^\dagger \cdot \mathbf{B}$ – Gram matrix of returned basis
        $\{\mu_{i,j}\}_{0\leqslant i,j\leqslant e}, \{r_{i,j}\}_{0\leqslant i,j\leqslant e}$ - GSO coefficients corresponding to $\mathbf{B}\cdot\mathbf{U}$
for $s \leqslant i \leqslant e$ do
    Find unit $u$ that unit reduces $r_{i,i}^{1/2}$.                ▷ CVP on the Log-unit lattice
    $\mathbf{u}_i := \mathbf{u}_i \cdot u$
    $\mathbf{G}_{i,\ell} := \mathbf{G}_{i,\ell}\cdot u$ for $\ell \leqslant i$; $\mathbf{G}_{\kappa,i} := \mathbf{G}_{\kappa,i}\cdot \overline{u}$ for $i \leqslant \kappa < n$
    for $0 \leqslant j < i$ do
        $r_{i,j} := r_{i,j}\cdot u$; $\mu_{i,j} := \mu_{i,j}\cdot u$
    $r_{i,i} := r_{i,i}\cdot u$
return $\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}_{0\leqslant i,j\leqslant e}, \{r_{i,j}\}_{0\leqslant i,j\leqslant e}$
```

4.3 The main LLL routine

Inserting a vector in an algebraic lattice. LLL starts by considering projective lattices of the form $\pi_i([\mathbf{b}_i, \mathbf{b}_{i+1}])$. It finds a short (in the algebraic norm) vector $\mathbf{s}'$ in such 2-dimensional lattice and inserts $\mathbf{s}$ into the initial lattice basis. Such insertions take place until the basis is LLL reduced as per Definition 3. Each of these insertions is equivalent to applying a transformation given by a unimodular matrix $\mathbf{W} \in \mathcal{O}_K^{2\times 2}$ such that $\mathbf{w}_0 \cdot [\mathbf{b}_i, \mathbf{b}_{i+1}] = w_0\mathbf{b}_i + w_1 \cdot \mathbf{b}_{i+1} = \mathbf{s}$ where $\pi_i(\mathbf{s}) \in \mathcal{L}(\pi_i([\mathbf{b}_i, \mathbf{b}_{i+1}]))$, to $[\mathbf{b}_i, \mathbf{b}_{i+1}]$.

Given w_0 and w_1, the construction of a such $\mathbf{W}$ boils down to solving the Bézout equation $\mu w_0 + \nu w_1 = 1$, i.e., finding μ, ν for given w_0, w_1. Given two coprime ideals $\mathfrak{a} = w_0 \cdot \mathcal{O}_K$ and $\mathfrak{b} = w_1 \cdot \mathcal{O}_K$, there exist two elements $a \in \mathfrak{a}, b \in \mathfrak{b}$ such that $a + b = 1$ [10, Proposition 1.3.1] which gives a solution to the Bézout equation.

Recall that we define the primitivity of $\mathbf{v}$ with coefficients $(c_0, \ldots, c_{n-1})$ w.r.t. a basis of $\mathcal{L} \subset K^n$ as (see Definition 1): $\mathbf{v}$ is primitive if and only if $\sum_{i<n} c_i \cdot \mathcal{O}_K = \mathcal{O}_K$. The rationale behind this definition follows from the Laplace expansion formula which suggests that the determinant of any $n \times n$ unimodular matrix with its first row equal to $(c_0, \ldots, c_{n-1})$ lies in $\sum_{i<n} c_i \cdot \mathcal{O}_K$. If this sum is not $\mathcal{O}_K$, it is impossible to obtain a unimodular matrix with $(c_0, \ldots, c_{n-1})$ being its first row. Since we need such unimodular matrix to update a basis, this appears to be an issue. Later in Section 5 we explain how we can leverage this primitivity condition, for this section we assume that we can always find a primitive vector and, thus, to construct the corresponding Bézout equation as explained later in this section.

In order to insert a short primitive vector we use the subroutine called BezoutTransform introduced in [19] that solves Bézout equations over K.

Assume, we found a short vector with coefficient vector $\mathbf{w} = (w_0, w_1)^T$ (we will be inserting only in rank-2 modules, hence the dimension of $\mathbf{w}$). The purpose of BezTransform is to find $\nu, \mu \in \mathcal{O}_K$ such that the following matrix is unimodular:

$$\mathbf{W} = \begin{pmatrix} w_0 & \nu \\ w_1 & \mu \end{pmatrix} \in \mathcal{O}_K^{2\times 2}. \tag{21}$$

We make use of such $\mathbf{W}$ to insert a vector with a short projection onto $\pi_i(\mathbf{s}) \in \mathcal{L}(\pi_i([\mathbf{b}_i, \mathbf{b}_{i+1}]))$ into the basis of the original lattice.

Given w_0 and w_1, the construction of $\mathbf{W}$ boils down to solving the Bézout equation $\mu w_o + \nu w_1 = 1$. The difference between our approach described in Algorithm 4.3 and the one from [19] consists in the fact that we use [10, Proposition 1.3.1] to solve the Bézout equation. The algorithm from [10, Proposition 1.3.1] requires more time in practice than the one from [19], but it improves the quality of the output. Moreover, in Section 5 we show how to also improve the efficiency of this algorithm. The following lemma due to Cohen allows us to find μ, ν efficiently and is the core of Algorithm 4.3.

Lemma 9 ([10, Proposition 1.3.1]). *Given two coprime ideals $\mathfrak{a}$ and $\mathfrak{b}$, one can find in time polynomial in bitsize of $\mathfrak{a}, \mathfrak{b}$ and in $\log |\Delta_K|$ elements $a \in \mathfrak{a}, b \in \mathfrak{b}$ such that $a + b = 1$.*

Proof. Apply [10, Algorithm 1.3.2] to find such a and b in time polynomial in $\log |\Delta_K|$ and the bitsizes of $\mathfrak{a}$ and $\mathfrak{b}$. To argue on the bitsizes of a and b, we can shorten them using [10, Algorithm 1.4.13] and obtain the bound of $2^d \cdot \lambda_1(\mathfrak{a}\mathfrak{b})$ on the Euclidean norm of a. Since $b = 1 - a$, the norm of b is also bounded. The runtime of [10, Algorithm 1.4.13] is polynomial in $\log |\Delta_K|$ and the bitsize of $\mathfrak{a}$ and $\mathfrak{b}$ as it computes a $\mathbb{Z}$ basis of an ideal and then LLL reduces it. □

Suppose we found a short vector $\mathbf{v} = \mathbf{M} \cdot (w_0, w_1)^T$ in a 2-dimensional lattice with basis $\mathbf{M}$ and want to insert $\mathbf{v}$ into M. To do so we use Lemma 9 to complete $(w_0, w_1)^T$ into a unimodular matrix $\mathbf{W}$ and apply the resulting transformation to $\mathbf{M}$. It makes $\mathbf{v}$ to be the first vector in the basis. Moreover, it follows from the proof of Lemma 9 that the norm of the second new basis vector will be polynomially bounded in the bitsize of the original basis M.

Algorithm 4.3 BezoutTransform

```
Input: $(w_0, w_1)$ over $\mathcal{O}_K$ such that $w_0 \cdot \mathcal{O}_K + w_1 \cdot \mathcal{O}_K = \mathcal{O}_K$.
Output: unimodular matrix with first vector $(w_0, w_1)^T$
1: $\mathfrak{a}, \mathfrak{b} := w_0 \cdot \mathcal{O}_K, w_1\mathcal{O}_K$
2: Using Lemma 9 obtain $a \in \mathfrak{a}, b \in \mathfrak{b}$ such that $a + b = 1$.
3: return $\begin{pmatrix} w_0 & -b/w_1 \\ w_1 & a/w_0 \end{pmatrix}$
```

Guarantees on Bézout solutions The routine given in Algorithm 4.3 finds coefficients μ and ν that solve a Bézout equation Equation (21) but does not guarantee that the obtained solution is (somewhat) short. As this solution represents the coefficients of the second inserted vector in a rank-2 projective lattice, we are interested in making this solution short. To do so we introduce two improvements to Algorithm 4.3.

Let us first describe how we make a solution shorter. Let $\mathfrak{a} = w_0 \cdot \mathcal{O}_K, \mathfrak{b} = w_1 \cdot \mathcal{O}_K$ be two coprime ideals, and let $a \in \mathfrak{a}$ and $b \in \mathfrak{b}$ such that $a + b = 1$ and $a = \mu \cdot w_0,\ b = \nu \cdot w_1$ be the output of Algorithm 4.3. We then search for $t \in \mathfrak{ab}$ such that $a - t$ is small w.r.t. the Euclidean norm. Since $b = 1 - a$, we have that $b + t$ will be small as well. Notice that $a - t \in \mathfrak{a},\ b + t \in \mathfrak{b}$ and $(a - t) + (b + t) = 1$. We set $t = \lfloor \frac{a}{w_0 w_1} \rceil \cdot w_0 w_1$ and therefore bounding $||a - t||$ from above as $d/2 \cdot ||w_0 \cdot w_1||_\infty$ due to the triangular inequality.

While solving the Bézout equation can be done in time polynomial in the bitsize of the equation's coefficients, this algorithm ceases to perform in reasonable time starting at the 512-th cyclotomic field on. The second enhancement reduces the problem of solving a Bézout equation into a subfield, where the answer can be found faster. Following [19], we descend the problem into a subfield, solve it there, and reduce the answer as shown above.

For example, solving a Bézout equation over 256th cyclotomic field may require several minutes with Pari GP's idealaddtoone, our approach solves it in seconds. The runtime of our approach also scales better with the bitsize of the coefficients of a Bézout equation.

Algorithm 4.4 Improved_BezTransform

```
Input: $(w_0, w_1)$ over $\mathcal{O}_K$ such that $w_0 \cdot \mathcal{O}_K + w_1 \cdot \mathcal{O}_K = \mathcal{O}_K$
       $d'$ – threshold dimension
Output: unimodular matrix with first vector $(w_0, w_1)$
$\mathfrak{a}, \mathfrak{b} := w_0 \cdot \mathcal{O}_K, w_1 \mathcal{O}_K$
if $\deg(K) > d'$ then
    Set $L := \mathbb{Q}[\zeta^2] \subset K = \mathbb{Q}[\zeta]$ – an index-2 subfield of $K$.
    $\nu', \mu' := \left(\mathsf{Improved\_BezTransform}(\mathcal{N}_{K/L}(w_0), \mathcal{N}_{K/L}(w_1))\right)_1$   ▷ Use the first row of the output
    $\mu := w_0^{-1} \mathcal{N}_{K/L}(w_0) \cdot \mu'$; $\nu := w_1^{-1} \mathcal{N}_{K/L}(w_1) \cdot \nu'$
    $a := \mu \cdot w_0$; $b := \nu \cdot w_1$
else
    Using Lemma 9 obtain $a \in \mathfrak{a}, b \in \mathfrak{b}$ such that $a + b = 1$.
$t := \lfloor \frac{a}{w_0 \cdot w_1} \rceil \cdot w_0 \cdot w_1$
$a := a - t; b := b + t$
return $\begin{pmatrix} w_0 & -b/w_1 \\ w_1 & a/w_0 \end{pmatrix}$
```

4.4 Algebraic LLL Algorithm

Let us now summarize the steps of the algebraic LLL given in Algorithm 4.5. To keep the bitsizes of basis vectors polynomially bounded, we call size and unit reduction (Algorithm 4.1 and Algorithm 4.2). This ensures that we can efficiently compute the norms and check the condition from Definition 3. Next we consider the i-th projective sublattice $\mathbf{M}_i$ for the least i that violates Definition 3, and search for a short primitive vector in this sublattice. Once such vector is found, we insert it using Algorithm 4.3. The process continues until the basis is LLL reduced. We give more details on how to find a short vector in $\mathbf{M}_i$ later in this section, for now we assume access to algebraic approxSVP oracle that returns short primitive vectors.

Under this assumption, the complexity analysis of LLL from [24] carries over to Algorithm 4.5. The only significant difference between our Algorithm 4.5 and [24, Algorithm 3.4] is in the way we perform the insertion. While we rely on solving Bézout equation, Lee et al. [24, Lemma 2.8] follow a more general approach of converting a short generating set of a module to its pseudobasis preserving the shortness. As we chose to work with bases rather than pseudobases, Algorithm 4.3 better fits out design choice.

Algorithm 4.5 BasicLLL

```
Input:  $\mathbf{B} \in K^{m \times n}$ – basis matrix of a free module,
        $\mathbf{G}$ – Gram matrix of $\mathbf{B}$,
        $\alpha \in \mathbb{R} : \alpha > \gamma_{\mathcal{N}}^{2d} 2^d \Delta_K$ – constant defining the quality of the reduction.
Output: unimodular transform $\mathbf{U}$ such that $\mathbf{B} \cdot \mathbf{U}$ is LLL-reduced.
 1: $\mathbf{U} := \mathrm{Id}_n$
 2: while index $i$ exists in Line 5 do
 3:     $\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}, \{r_{i,j}\} :=$ size_reduce($\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}, \{r_{i,j}\}, 0, n-1$)
 4:     $\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}, \{r_{i,j}\} :=$ unit_reduce($\mathbf{U}, \mathbf{G}, \{\mu_{i,j}\}, \{r_{i,j}\}, 0, n-1$)
 5:     Find minimal $i$ such that $\alpha \cdot \mathcal{N}(r_{i+1,i+1}) \leqslant \mathcal{N}(r_{i,i})$
 6:     Compute Gram-Schmidt vectors $\{\mathbf{b}_\kappa^*\}_{\kappa \leqslant i}$
 7:     Find a short primitive vector $\mathbf{v}$ in $\mathbf{M}_i = [\mathbf{b}_i^*, \pi_i(\mathbf{b}_{i+1})]$
 8:     $(w_0, w_1)^T := \mathbf{M}^{-1} \cdot \mathbf{v}$
 9:     $\mathbf{W} :=$ BezTransform($w_0, w_1$)                          ▷ Algorithm 4.3
10:     Apply $\mathbf{W}$ to $i$-th and $(i+1)$-th columns of $\mathbf{U}$ and update $\mathbf{G}$ accordingly.
11: return $\mathbf{U}$
```

Theorem 4 (Adapted from [24, Theorem 3.4]). *Let $\mathcal{A}$ be an oracle that solves algebraic approxSVP with approximation factor $\gamma_{\mathcal{N}}$. Then, for a cyclotomic power-of-2 field K of degree d, Algorithm 4.5 given on input a real $\alpha > \gamma_{\mathcal{N}}^{2d} 2^d \Delta_K$,a basis $\mathbf{B} \in K^{m \times n}$ of a free $\mathcal{O}_K$ module, its Gram matrix $\mathbf{G} = \mathbf{B}^\dagger \cdot \mathbf{B}$, and an access to $\mathcal{A}$, returns a unimodular $\mathbf{U}$ such that $\mathbf{BU}$ is an LLL reduced* *basis of $\mathcal{L}(\mathbf{B})$. It requires*

$$\text{poly}\left(n, m, \log|\Delta_k|, \frac{1}{\log\alpha\cdot(\gamma_{\mathcal{N}}^{2d}2^d|\Delta_K|)}\right)\cdot\log\frac{\max_i \mathcal{N}(r_{i,i})}{\min_i \mathcal{N}(r_{i,i})}$$

number of calls to $\mathcal{A}$ and terminates in

$$\text{poly}\left(\log|\Delta_k|, \text{bitsize}(\mathbf{B}), \frac{1}{\log\alpha\cdot(\gamma_{\mathcal{N}}^{2d}2^d|\Delta_K|)}\right)$$

time.

Proof. The reduction process starts with the size reduction Algorithm 4.1 and the unit reduction Algorithm 4.2. As proven in Theorem 2 and Theorem 3, they preserve the module and terminate in time $\text{poly}(n, \log|\Delta_k|, \text{bitsize}(\mathbf{B}))$.

Suppose that the algebraic approxSVP oracle $\mathcal{A}$ is invoked on the projective lattice $\mathbf{M}_i$ starting at some index $i < n-1$, and it returns a primitive vector $\mathbf{v}$. Then Algorithm 4.3, given the coordinates of this short vector w.r.t. the basis $\mathbf{M}$ of the projective lattice, outputs a unimodular matrix $\mathbf{W} \in \mathcal{O}_K^{2\times 2}$ such that the first vector in $\mathbf{M}\cdot\mathbf{W}$ is $\mathbf{v}$. Applying $\mathbf{W}$ to the i-th and $(i+1)$-th vectors of $\mathbf{B}$ we obtain the vector $\mathbf{v}' \in \mathcal{L}$ such that its projection orthogonally to the $\text{span}\{\mathbf{b}_j\}_{j<i}$ is $\mathbf{v}$. This operation preserves the module and Lemma 9 guarantees that the new $(i+1)$-th basis vector is polynomially bounded in its bitsize.

What remains to show is an upper bound on the number of the while-loop iterations in Algorithm 4.5. For this we refer the reader to the proof of [24, Theorem 3.4] as it follows exactly the same arguments. □

Algebraic approxSVP. The final ingredient required to complete the algebraic LLL algorithm is the algebraic approxSVP oracle called in Line 7 of Algorithm 4.5. To our knowledge, there is only one specialized algebraic approxSVP oracle introduced in [24] (and improved in [13]). In the former work,the authors propose an explicit algorithm to instantiate such an oracle. While being applicable to any rank-2 module, it requires costly precomputations in lattices of high dimensions and class group computations, which, given the state-of-the-art implementations, are too prohibitive in practice.

As we aim for a more practical approxSVP oracle, we sacrifice on generality in the sense that we do not have full control on *algebraic* norms of the returned vectors. In our implementation, we instantiate our algebraic approxSVP oracle with a classical BKZ reduction by mapping the bases of projective rank-2 modules $\mathbf{M}_i$'s defined in Equation (4) to $\mathbb{R}^{2d}$. As this is not an *authentic* algebraic approxSVP oracle, we cannot guarantee that it finds short algebraic norm vectors in $\mathbf{M}_i$'s. However, in practice small vectors in Euclidean norms tend to have small algebraic norms as well. Furthermore, as BKZ returns many short vectors (e.g., a basis of somewhat short vectors), we have many candidates for insertion.

4.5 Output quality of algebraic LLL

Analogously to [24, Lemma 3.2], we can bound the algebraic norm of the first vector of an LLL reduced basis. For that, we need the following lemma.

Lemma 10 ([24, Lemma 2.6]). *Let M be a rank-n algebraic module and let $\mathcal{L}$ be the corresponding algebraic lattice. Consider GSO coefficients $r_{i,i}, i < n$ of M. Then $\lambda_1^{\mathcal{N}}(\mathcal{L}) \geqslant \min_{i<n}(\mathcal{N}(r_{i,i}))$.*

The next lemma is an adaptation of the similar result from [24, Lemma 3.2] to the case of the free modules over the power-of-2 cyclotomic fields.

Lemma 11 (Adapted from [24, Lemma 3.2]). *Let $\mathbf{B}$ be an algebraic LLL reduced free basis of $\mathcal{L}$ – a free $\mathcal{O}_K$ module of rank n. Then*

$$\mathcal{N}(\mathbf{b}_0) \leqslant \alpha^{n-1} \cdot \lambda_1^{\mathcal{N}}(\mathcal{L}).$$

In addition, if $\alpha = (1+\varepsilon)\,\gamma_{\mathcal{N}}^{2d} 2^d |\Delta_K|$ for some $\varepsilon > 0$, and d –the degree of K, we have that:

$$\frac{\lambda_1(\mathbf{b}_0 \cdot \mathcal{O}_K)}{\lambda_1(\mathcal{L})} \leqslant \alpha^{\frac{n-1}{d}} \cdot |\Delta_K|^{1/(2d)}.$$

Proof. Recall Lemma 10 which suggests that $\lambda_1^{\mathcal{N}}(\mathcal{L}) \geqslant \min_i \mathcal{N}(r_{i,i})$. Combining this fact with the definition of the LLL reducedness we have the first claim.

We can find a primitive vector $\mathbf{s}$ with approximation factor $\gamma_{\mathcal{N}}$ for every projective module M_i obtained during the execution of Algorithm 4.5. Such vectors can be inserted into the basis since they are primitive.

By Lemma 2 applied to the rank-1 module spanned by $\mathbf{b}_0$, we have that $\lambda_1(\mathbf{b}_0 \cdot \mathcal{O}_K) \leqslant \sqrt{d} \cdot (\lambda_1^{\mathcal{N}}(\mathbf{b}_0 \cdot \mathcal{O}_K))^{1/d} \cdot |\Delta_K|^{1/(2d)}$. As $\mathbf{b}_0 \cdot \mathcal{O}_K$ is a rank-1 module, we have that $\lambda_1^{\mathcal{N}}(\mathbf{b}_0 \cdot \mathcal{O}_K) = \mathcal{N}(\mathbf{b}_0)$ and the following holds thanks to Lemma 2: $\lambda_1(\mathbf{b}_0 \cdot \mathcal{O}_K) \leqslant \sqrt{d}\mathcal{N}(\mathbf{b}_0)^{1/d} |\Delta_K|^{1/(2d)}$. Applying the same lemma to the whole module gives the bound on its Euclidean minima $\lambda_1(\mathcal{L}) \geqslant \sqrt{d} \cdot \lambda_1^{\mathcal{N}}(\mathcal{L})^{1/d}$. Since $\mathcal{N}(\mathbf{b}_0)^{1/d} \leq \alpha^{(n-1)/d} \cdot \lambda_1^{\mathcal{N}}(\mathcal{L})$, we obtain the result:

$$\frac{\lambda_1(\mathbf{b} \cdot \mathcal{O}_K)}{\lambda_1(\mathcal{L})} \leqslant \frac{\sqrt{d}\alpha^{(n-1)/d} \cdot \lambda_1^{\mathcal{N}}(\mathcal{L}) |\Delta_K|^{1/(2d)}}{\sqrt{d} \cdot \lambda_1^{\mathcal{N}}(\mathcal{L})^{1/d}} = \alpha^{(n-1)/d} \cdot |\Delta_K|^{1/(2d)}.$$

□

5 Implementation of Algebraic LLL

5.1 Fast arithmetic

In order to make our algebraic LLL efficient, we make use of the fact that the canonical embedding is a linear map $\mathcal{F} : K^n \rightarrow \mathbb{C}^{n \cdot d}$. If we have two vectors $\mathbf{v}, \mathbf{u} \in K^n$, then $\mathcal{F}(\mathbf{v} + \mathbf{u}) = \mathcal{F}(\mathbf{u}) + \mathcal{F}(\mathbf{v})$. The canonical embedding is essentially the Fourier transformation that can be efficiently computed in $O(d \log d)$. That fact helps us to speed up the computations as follows. The equality $\mathcal{F}(\langle \mathbf{v} , \mathbf{u} \rangle) = \langle \mathcal{F}(\mathbf{v}) , \mathcal{F}(\mathbf{u}) \rangle$ also holds, which lowers the complexity of scalar product computations from $O(nd^2)$ to $O(nd \log d)$.

Similarly, we can speed up norm computations. Let $k \in K$ where K a power-of-2 cyclotomic field $K = \mathbb{Q}(\zeta)$. Then $\mathcal{F}(k) = (l_0, l_1, \ldots, l_{d/2-1}, \bar{l}_{d/2-1}, \ldots, \bar{l}_1, \bar{l}_0)$ for some $l_i \in \mathbb{C}, i < d$ and by keeping only $d/2$ first coefficients we can always reconstruct the whole vector. We denote $(l_0, l_1, \ldots, l_{d/2-1})$ as $\mathcal{F}'(k)$. The algebraic norm of k is then $\prod_{i<d/2} |(\mathcal{F}'(k))_i|^2$ and it can be computed in $d/2+1$ complex multiplications. Thus we store all the data: bases, GSO-coefficients, Gram matrices, and transformation matrices in the canonical embedding.

5.2 Precomputations and PIP

Log-unit lattice. For unit reduction (Algorithm 4.2) we require a good basis of the log-unit lattice. We precompute and classically LLL reduce log-unit lattices for 32-nd, 64-, 128-, 256-, 512-th cyclotomic fields. This allows us to efficiently solve approxCVP, i.e., for a given number field element, find a close unit that potentially reduces its Euclidean norm, which is the core of Algorithm 4.2.

In order to find a close unit we first need to construct a basis of a log unit lattice that allows us to solve approxCVP with good approximation factor. Following [32,12], we construct a basis of the log-unit lattice from the generators of the cyclotomic unit group $\frac{\zeta^i - 1}{\zeta - 1}$ for all i coprime to d. We then classically LLL reduce this basis and store it with the corresponding cyclotimic units. Denote by $\mathbf{B}_{\log}$ this LLL reduced basis consisting of vectors $\mathbf{v}_i$ and the corresponding units $v_i \in K$.

Upon receiving a target $k \in K$, we map it onto H and call the Babai's Nearest Planes algorithm [2] on $\mathbf{B}_{\log}$. Let $(c_0, \ldots, c_{d/2-1}) \in \mathbb{Z}^{d/2-1}$ be the coefficients of the returned solution w.r.t. $\mathbf{B}_{\log}$. A close to k unit u is obtained as $\prod_{i \leqslant d/2-2} v_i^{c_i}$. The unit that unit reduces k is then $u^{-1} = \prod_{i \leqslant d/2-2} v_i^{-c_i}$. The routine is shown in Algorithm 5.1. Essentially, it follows the approaches from [32,12].

Algorithm 5.1 LogCVP

```
Input:  $k \in K$ for a cyclotomic field $K$.
        An LLL reduced basis $\mathbf{B}_{\log} = \{\mathbf{v}_i\}_{0 \leqslant i < d/2-1}$ of the log unit lattice
        Cyclotomic units $v_i$ for $0 \leqslant i < d/2-1$.
Output: A cyclotomic unit $u$ such that $\mathrm{Log}(u^{-1})$ is close to $\mathrm{Log}(k)$.
Compute $\mathbf{t} := \mathrm{Log}(k) = \pi_{\mathbf{1}}(\mathrm{Log}(k))$ for $\mathbf{1}$ – the all-ones vector.
$(c_0, \ldots, c_{d/2-2}) \leftarrow \mathsf{BabaiNearestPlanes}(\mathbf{B}_{\log}, \mathbf{t})$    ▷ Use [2]
return $\prod_{i \leqslant d/2-1} v_i^{-c_i}$
```

5.3 Finding and inserting a short vector in M

In this section we present a complete LLL algorithm that includes all the enhancements described above.

We search for a short vector in a projective rank-2 modules defined in Equation (4) using classical BKZ reduction implemented in [31]. Depending on the degree d of the number field that M is defined over, we either directly map (the appropriate scaling of) M to $\mathbb{Z}^{2d}$, or we perform the descend procedure from Section 3.1.

The base case. When the number field K has a relatively small degree d, we can launch BKZ reduction algorithm on a basis of the canonical embedding [6] of rank-2 projective submodules $\mathbf{M}_i$ for $i \in \{0, \ldots, n-2\}$. It shortens the $2 \cdot d$ vectors of a basis of $\mathbf{M}_i$ over $\mathbb{Q}$ in their Euclidean norm. The algebraic-geometric inequality from Lemma 1 suggests that $\mathcal{N}(k) \leqslant ||k||^d \cdot d^{-d/2}$, and we hope that small elements in the Euclidean norm also have small algebraic norms. Since our algorithm works with transformation matrices, we focus on finding coefficients $\mathbf{w}' = (w'_0, \ldots, w'_{2d-1})^T \in \mathbb{Z}^{2d}$ w.r.t. the input basis of M_i that give us a short vector. Then we set $\mathbf{w} = (w_0, w_1)^T = \mathrm{Ascend}_{\mathbf{v}}(K, \mathbf{w}')$.

In practice sometimes a short vector cannot be inserted into the basis because it is not primitive. In order to lower the probability of such situation, we return a list of short vectors. The whole subroutine is presented in Algorithm 5.2.

Algorithm 5.2 SVP

```
Input: $\mathbf{B} \in \mathcal{O}_K^{n \times n}$ - basis matrix of the lattice
Output: coefficients $\mathbf{u}_k$ of short vectors of the lattice defined by $\mathbf{B}$
$\mathbf{M} := \mathsf{Descend}_{\mathsf{m}}(\mathbb{Q}, \mathbf{B}) = \{\mathbf{m}_i\}_{0 \leqslant i < n \cdot d}$
BKZ reduce $\mathbf{M}$ obtaining $\{\mathbf{m}'_i\}_{0 \leqslant i < n \cdot d}$
return $U := \{B^{-1} \cdot \mathsf{Ascend}_{\mathsf{v}}(K, \mathbf{m}'_i) \mid 0 \leqslant i < n \cdot d\}$
```

The recursive approach. Runtime of strong BKZ reduction becomes prohibitive in practice for modules of high dimensions over $\mathbb{Q}$. In this case we descend basis vectors of M given in Equation (4) into a subfield L of index 2 obtaining a basis for a rank-4 module over L.[7] We treat this new module as a new input to algebraic LLL reduction. Now in this new call to LLL, we shall be considering modules M_i's whose dimensions over $\mathbb{Q}$ are twice as small.

Complete algebraic LLL algorithm. The version of algebraic LLL with recursive descend to subfields is given in Algorithm 5.3. The main differences from Algorithm 4.5 are the following ones. First, rather than reducing rank-2 submodules of K, we descend into a subfield L and call approxSVP on rank-2 modules over L. Second, we control the lengths of the inserted basis vectors by solving Bézout

[6] We can launch it on the coefficient embedding of M_i as well, but this approach did not give us any practical advantage.

[7] Other powers-of-two are also possible, but in practice we chose to descend to as large subfields as possible.

equations more efficiently with Algorithm 4.4. At last, we update the Gram matrix coefficients of the input basis on-the-fly rather than recomputing them from an updated basis. Concretely, let the basis transformation $\mathbf{W}$ obtained in Line 24 of Algorithm 5.3 be of the form

$$\mathbf{W} = \begin{pmatrix} w_{0,0} & w_{0,1} \\ w_{1,0} & w_{1,1} \end{pmatrix}.$$

Let $\mathbf{G}$ be a non-updated Gram matrix and let $\mathbf{G}'$ denote the updated one. It will be easier to look at the three zones of the elements of $\mathbf{G}$ as in Figure 1. The following are the results of straightforward computations. In region I, we have:

$$\mathbf{G}'_{\kappa,\ell} = w_{\kappa-i,0}\mathbf{G}_{\kappa,\ell} + w_{\kappa-i,1}\mathbf{G}_{\kappa+1,\ell}, \quad \text{for } \kappa,\ell \in \{i,i+1\}. \tag{22}$$

Analogously, for region III we have:

$$\mathbf{G}'_{\kappa,\ell} = \overline{w}_{\ell-j,0}\mathbf{G}_{\kappa,\ell-j} + \overline{w}_{\ell-j,1}\mathbf{G}_{\kappa,\ell+1}, \quad \text{for } \kappa,\ell \in \{i,i+1\}. \tag{23}$$

The explicit formulas for the three elements from zone II are:

$$\begin{aligned}
\mathbf{G}'_{i,i} &= w_{0,0}\overline{w}_{0,0}\mathbf{G}_{i,i} + w_{0,0}\overline{w}_{0,1}\overline{\mathbf{G}}_{i+1,i} + w_{0,1}\overline{w}_{0,0}\mathbf{G}_{i+1,i} + w_{0,1}\overline{w}_{0,1}\mathbf{G}_{i+1,i+1} \\
\mathbf{G}'_{i+1,i+1} &= w_{1,0}\overline{w}_{1,0}\mathbf{G}_{i,i} + w_{1,0}\overline{w}_{1,1}\overline{\mathbf{G}}_{i+1,i} + w_{1,1}\overline{w}_{1,0}\mathbf{G}_{i+1,i} + w_{1,1}\overline{w}_{1,1}\mathbf{G}_{i+1,i+1} \\
\mathbf{G}'_{i+1,i} &= u_{1,0}\overline{w}_{0,0}\mathbf{G}_{i,i} + w_{1,0}\overline{w}_{0,1}\overline{\mathbf{G}}_{i+1,i} + w_{1,1}\overline{w}_{0,0}\mathbf{G}_{i+1,i} + w_{1,1}\overline{w}_{0,1}\mathbf{G}_{i+1,i+1}
\end{aligned} \tag{24}$$

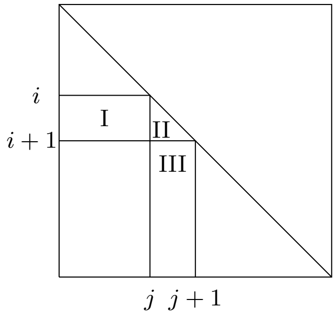

Fig. 1: Regions of the Gram matrix to be updated

6 Experiments

We implemented a version of algebraic LLL in SageMath 9.8. For the algebraic approxSVP oracle we call the FPYLLL [31] library. The implementation accompanying our work and the experimential data can be found at `https://github.com/mooninjune/AlgebraicLLL`. In all our experiments we use 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 used to run multiple parallel experiments.

Algorithm 5.3 BasicLLLWithDescend

```
Input:  $\mathbf{B} \in \mathcal{O}_K^{m \times n}$ – basis of a module lattice defined over $K = \mathbb{Q}(\zeta_d)$;
        $\alpha > 1 \in \mathbb{R}$ – constant defining the quality of the reduction;
        $d'$ – threshold dimension.
Output: unimodular transform $\mathbf{U}$ such that $\mathbf{B} \cdot \mathbf{U}$ is an LLL reduced
$\mathbf{U} := \mathrm{Id}_n$; $\mathbf{G} := \mathbf{B}^\dagger \cdot \mathbf{B}$
while $\exists i : \alpha^2 \mathcal{N}(r_{i+1,i+1}) > \mathcal{N}(r_{i,i})$ do
    $\mathbf{W}, \mathbf{G}, \{\mu_{i,\ell}\}, \{r_{i,\ell}\} :=$ unit_reduce$(\mathbf{G}, \mu, r, 0, n-1)$, $\ell \leqslant n-1$
    $\mathbf{U} := \mathbf{U} \cdot \mathbf{W}$
    $\mathbf{B} := \mathbf{B} \cdot \mathbf{W}$
    $\mathbf{W}, \mathbf{G}, \{\mu_{i,\ell}\}, \{r_{\ell,\ell}\} :=$ size_reduce$(\mathbf{G}, \mu, r, 0, n-1)$, $\ell \leqslant n-1$
    $\mathbf{U} := \mathbf{U} \cdot \mathbf{W}$
    $\mathbf{B} := \mathbf{B} \cdot \mathbf{W}$
    Compute $\{\mathbf{b}_\kappa^*\}_{\kappa \leqslant i}$ using Equation (1)
    $\pi_i(\mathbf{b}_{i+1}) = \mathbf{b}_{i+1} - \sum_{\kappa=0}^{i-1} \mu_{i+1,\kappa} \cdot \mathbf{b}_\kappa^*$
    $\mathbf{B}' := [\mathbf{b}_i^*, \pi_i(\mathbf{b}_{i+1})]$                                ▷ $2 \times m$ matrix
    if $\deg(K) > d'$ then
        Set $L = \mathbb{Q}[\zeta_{d/2}]$ for $\zeta_{d/2}$ – primitive $(d/2)$-th root of unity.
        $T :=$ Ascend$_\mathsf{m}$(kefll(Descend$_\mathsf{m}$$(L, \mathbf{B}')$, $\alpha \cdot |\Delta_L / (2 \cdot \Delta_K)|, d'$))
    else                                                   ▷ If degree of $K$ makes SVP feasible
        $T := \mathrm{SVP}(\mathbf{B}')$                          ▷ Use SVP oracle on rank-2 submodule.
    $\mathbf{t} := \emptyset$
    for $\mathbf{t} = (t_0, t_1) \in T$ do
        if $t_0 \cdot \mathcal{O}_K + t_1 \cdot \mathcal{O}_K = \tau \cdot \mathcal{O}_K$ then
            $\mathbf{t} := \mathbf{t}/\tau$
            break
    if $\mathbf{t} = \emptyset$ then
        continue
    $\mathbf{W} :=$ Improved_BezTransform$(\mathbf{t}/\tau, d')$;
    $\mathbf{U} := \mathbf{U} \cdot \mathbf{W}$
    $\mathbf{B} := \mathbf{B} \cdot \mathbf{W}$
    Update $\mathbf{G}$ according to Equations (22) to (24)
return $\mathbf{U}$
```

6.1 ZGSA accuracy

One of the most prominent example of algebraic lattices in cryptography are modules that come from the module-LWE problem [23]. Modern lattice-based post-quantum signature and encryption standards [15,17,7] rely on the hardness of this problem. Given $m > k \geq 0$ and a modulus $q > 1$, sample $k \cdot m$ elements $(a_{0,0}, \ldots, a_{0,k-1}, \ldots, a_{m-1,0}, \ldots, a_{m-1,k-1})$ uniformly at random from $\mathcal{O}_K / q\mathcal{O}_K$. Consider a rank-$(m+k)$ module with an $\mathcal{O}_K$-basis given by the columns of the following matrix $\mathbf{B} \in \mathcal{O}_K^{(m+k)\times(m+k)}$:

$$\mathbf{B} = \begin{pmatrix} q & \dots & 0 & a_{0,0} & \dots & a_{k-1,0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & q & a_{m-1,0} & \dots & a_{k-1,m-1} \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}. \tag{25}$$

For $\mathbf{B}$ as above, the module-LWE problem can be formulated as the closest vector problem on the module lattice generated by $\mathbf{B}$ with the guarantee that the solution is unique. Hence, we are interested in the behaviour of lattice reduction algorithms on such bases.

We run our algebraic LLL on modules defined by bases $\mathbf{B}$ (with $k = m$) and monitor how the profile, that is the sequence $\{\mathcal{N}(r_{i,i})\}_i$ as per Definition 4 with the ideals of the pseudobasis being $\mathcal{O}_K$, changes. We plot these profiles in Figure 2 and compare the profiles to Heuristic 3. By the shape of $\mathbf{B}$ given in Equation (25), the input profile consists of the first k q-ary vectors and the last $r_{i,i}$ for $k \leq i < 2k$ are 1. As expected, after the LLL reduction, the profiles becomes 'flatter' and resembles the predicted profiles quite accurately. This situation is analogous to the behaviour of the classical lattice profiles after the execution of a classical LLL algorithm.

6.2 NTRU modules

Concrete estimations. To compare the predictions of Heuristic 4 with practice, we consider NTRU modules M over a cyclotomic field K of a conductor f as in Equation (6). We descend the bases of these modules twice obtaining corresponding modules of rank 8 over a filed $L \subset K$ for a degree $\deg L = (\deg K)/4$. We launch our LLL algorithm on such bases and detect the recovery the dense sublattice containing $(\mathbf{f}, \mathbf{g})$.

Our estimator predicts that algebraic LLL recovers such dense sublattice at $\log_2 q = 12.8$ for $f = 32$, $\log_2 q = 16.1$ for $f = 64$, and $\log_2 q = 20.1$ for the conductor $f = 128$. The comparison between our predictions with the experimental data is given in Table 1. Our experiments confirm the predictions: the success rate for the predicted $\log q$ is close to 1 for cases $f = 32, 64$.

A minor but still visible discrepancy with the predictions appears in the case of $f = 128$. This phenomenon occurs due to the following issue. For 128-th cyclotomic field we are forced to launch an algebraicSVP oracle on submodules of dimension 128 over $\mathbb{Z}$. Using the BKZ algorithm with large block sizes becomes rather expensive, so we use block size at most 50 which affected the quality of the oracle, but made our extensive experiments feasible. A minor adjustment $\gamma_{\mathcal{N}}$ to 0.55 gives a prediction $\log q = 21.2$ which resembles the actual situation more precisely.

In order to estimate for which q the DSD event occurs for the *classical LLL algorithm*, we ran the DSD estimator from [14] for the block size set to 2. The comparison of the least moduli sufficient to trigger the DSD events both in

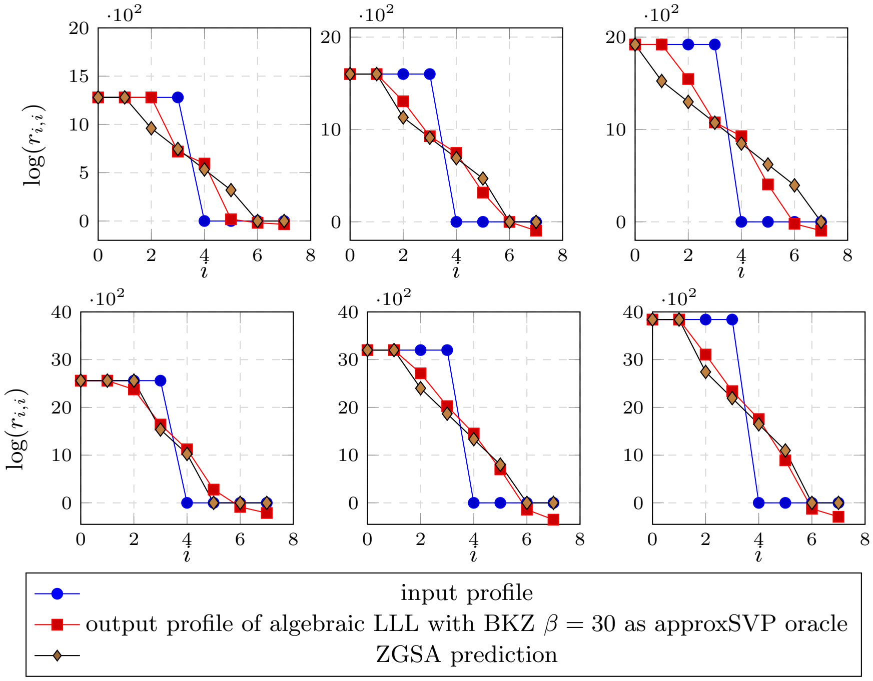

Fig. 2: Algebraic profiles for q-ary modules defined in Equation (25) for $k = m = 4$. We take cyclotomic fields of conductor $f = 64$ for the top figures, and $f = 128$ for the bottom figures. Profiles are averaged over 20 different modules. From left to right: we choose q to be primes of 20-, 25-, 30-bits.

classical and the algebraic settings are presented in Table 2. The data illustrates that the algebraic DSD events occur for larger moduli than in the classical case.

We used the FPyLLL implementation of BKZ-50 algorithm [31] for an algebraic approxSVP oracle inside the algebraic LLL. This block-size is significantly greater than the block sizes required to detect dense sublattice using classical lattice reduction.

For the NTRU modules over the 512-th cyclotomic field we ran classical BKZ reduction and compared the average block sizes required to trigger the DSD event. The results are presented in Table 3. It shows that, while we relied on the BKZ with the block size 50 to reduce the algebraic modules, the reduction could be performed by means of the classical algorithms with significantly smaller blocksizes. In order to make algebraic LLL competitive with the classical lattice reduction, a faster algebraic approxSVP oracle is needed. If such an oracle exists and can be efficiently implemented in polynomial time, our proposed algorithm can compete with the classical LLL at triggering the DSD events at least for mentioned parameters. We leave the question of existence of such oracle open.

$f = 32$		$f = 64$		$f = 128$	
$\log_2 q$	Success rate, %	$\log_2 q$	Success rate, %	$\log_2 q$	Success rate, %
12.5	80	16.0	90	20.0	0
13.0	100	16.5	100	20.5	30
13.5	100	17.0	100	21.0	75
14.0	100	17.5	100	21.5	95
14.5	100	18.0	100	22.0	100

Table 1: Percentage of DSD events on various algebraic LLL reduced NTRU lattices. For $f \in \{32, 64\}$ the BKZ block size is 25 and 50 for $f = 128$.

Field conductor f	32	64	128
$\log q$ for classic LLL	5.95	10.4	14.85
$\log q$ for algebraic LLL	12.8	16.1	20.1

Table 2: Predicted $\log q$ sufficient to trigger a DSD event on NTRU modules: our LLL vs. classical one.

6.3 Insertion Failures

A short vector with coefficients (c_0, c_1) will be inserted into a dimension-2 lattice basis if $c_0 \cdot \mathcal{O}_K + c_1 \cdot \mathcal{O}_K = \mathcal{O}_K$. Let us now discuss how restrictive this condition is. In our implementation of the algebraic LLL algorithm we 1)call BKZ reduction which returns many short vectors, and 2) try to leverage the condition using a PIP solver. However, there exist rank-2 $\mathcal{O}_K$-modules with the property that none of their free bases contains the shortest (again, in the Euclidean norm) vector. We now explicitly describe such modules.

Let M be given by

$$M = \begin{pmatrix} r_{0,0} \\ 0 \end{pmatrix} \cdot \mathcal{O}_K \oplus \begin{pmatrix} r_{1,0} \\ r_{1,1} \end{pmatrix} \cdot \mathcal{O}_K, \tag{26}$$

for some $r_{0,0}, r_{1,0}, r_{1,1} \in K_{\mathbb{R}}$. Such modules arise, for example, when we look at R-factors of algebraic rank-2 modules.

Suppose that a shortest vector in the module defined by Equation (26), is $\mathbf{v} = (s, 0)^T$. As the second coordinate of $\mathbf{v}$ is zero, we have that the intersection $M \cap (K_{\mathbb{R}} \times \{0\})$ is $\begin{pmatrix} r_{0,0} \\ 0 \end{pmatrix} \cdot \mathcal{O}_K$. It may, however, happen that $s \cdot \mathcal{O}_K \neq r_{0,0} \cdot \mathcal{O}_K$. In this case $(s, 0)$ cannot be inserted as a basis vector of M while keeping it a basis (not pseudobasis).

This example relies on the fact that the principal ideals of number fields might not have their shortest element in the Euclidean norm being a generator. We perform some experiments to see how often this situation arises in practice.

$\log_2 q$	20.0	20.5	21.0	21.5	22.0
β_{BKZ}	2.5	2.3	2.2	2.0	2.0

Table 3: Average β_{BKZ} that triggers a DSD event on NTRU modules over 512-th cyclotomic field with parameter q.

For that we generate “random” principal ideals in cyclotomic fields of conductors 32 and 64. The concept of “randomness” in the ideal class group is given in [6], where the authors describe a random walk in the so-called Arakelov class group. We do not give the details of this random walk here, but refer the reader to [6] and to our implementation `https://github.com/mooninjune/AlgebraicLLL`.

The result of a walk is an ideal $\mathcal{I}$. We check if it is principal (this step makes the experiments hard to extend to large fields). If the ideal turns out to be principal, we run classical enumeration on its (scaled) basis in order to find its shortest nonzero vector $\mathbf{s}$. We then compute how often $\mathcal{I} \neq s\mathcal{O}_K$, where s is such that its coefficient embedding is $\mathbf{s}$. The results of the experiments are given in Table 4. These results suggest that an Euclidean SVP oracle does not guarantee a solution to SVP in the algebraic norm.

Field conductor	% ideals not generated by their shortest element
32	69.4
64	70.2

Table 4: Percentage of principal ideals not generated by their shortest (in Euclidean norm) element. We run 500 experiments per field.

Yet not all is lost when the candidate for insertion is not primitive. In case $c_0 \cdot \mathcal{O}_K + c_1 \cdot \mathcal{O}_K = \tau \cdot \mathcal{O}_K$ for some $\tau \in \mathcal{O}_K$, we can insert $\mathbf{v}/\tau$ into the basis since now $\mathbf{v}/\tau$ is primitive. Finding τ is known as the Principal Ideal Problem (PIP) that asks to find, for a given principal ideal (in an $\mathcal{O}_K$-basis), its generator. Known algorithms for this problem are subexponential in Δ_K assuming Generalized Riemann Hypothesis [3,4]. However, for some number fields of not so large degree, solving PIP is efficient in practice.

Of course, we do not know a priori that $c_0 \cdot \mathcal{O}_K + c_1 \cdot \mathcal{O}_K$ is principal. However, as we have precomputed the class group for fields of conductor up to 128, this check is efficient in practice. On fields of conductor 256 and higher, the algorithm from [4] fails on non-principal inputs, which can be detected.

References

1. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes. In: Annual International Cryptology Conference. pp. 153–178. Springer (2016) 2, 10
2. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Comb. **6**(1), 1–13 (1986) 8, 28
3. Biasse, J.F.: Subexponential time relations in the class group of large degree number fields. Adv. Math. Commun. **8**(4), 407–425 (2014) 4, 35
4. Biasse, J.F., Espitau, T., Fouque, P.A., Gélin, A., Kirchner, P.: Computing generator in cyclotomic integer rings: A subfield algorithm for the principal ideal problem in and application to the cryptanalysis of a FHE scheme. In: Advances in Cryptology–EUROCRYPT 2017. pp. 60–88 (2017) 35
5. Biasse, J.F., Song, F.: Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In: Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms. pp. 893–902. SIAM (2016) 4
6. de Boer, K., Ducas, L., Pellet-Mary, A., Wesolowski, B.: Random self-reducibility of ideal-SVP via Arakelov random walks. In: Advances in Cryptology - CRYPTO 2020. pp. 243–273 (2020) 35
7. Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In: 2018 IEEE EuroS&P. pp. 353–367 (2018) 1, 31
8. Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J.M., Saito, T., Schwabe, P., Whyte, W., Xagawa, K., Yamakawa, T., Zhang, Z.: PQC round-3 candidate: NTRU. Technical report (2019), `https://ntru.org/f/ntru-20190330.pdf` 10
9. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Advances in Cryptology – ASIACRYPT 2011. pp. 1–20 (2011) 8
10. Cohen, H.: Advanced topics in computational number theory, vol. 193. Springer Science & Business Media (2012) 5, 13, 22, 23
11. Coppersmith, D., Shamir, A.: Lattice attacks on NTRU. In: International conference on the theory and applications of cryptographic techniques. pp. 52–61. Springer (1997) 2
12. Cramer, R., Ducas, L., Peikert, C., Regev, O.: Recovering short generators of principal ideals in cyclotomic rings. In: Advances in Cryptology–EUROCRYPT 2016. pp. 559–585. Springer (2016) 21, 28
13. De Micheli, G., Micciancio, D.: A fully classical LLL algorithm for modules. Cryptology ePrint Archive (2022) 26
14. Ducas, L., van Woerden, W.P.J.: NTRU fatigue: How stretched is overstretched? In: Advances in Cryptology - ASIACRYPT 2021. pp. 3–32. Lecture Notes in Computer Science (2021) 2, 9, 10, 11, 32
15. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (Feb 2018) 1, 31
16. Fieker, C., Stehlé, D.: Short bases of lattices over number fields. In: Algorithmic Number Theory. pp. 157–173 (2010) 13

17. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhan, Z.: FALCON: Fast-fourier lattice-based compact signatures over NTRU (2018), `https://www.di.ens.fr/~prest/Publications/falcon.pdf` 1, 31
18. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) ANTS-III. pp. 267–288 (1998) 1
19. Kirchner, P., Espitau, T., Fouque, P.A.: Algebraic and Euclidean lattices: optimal lattice reduction and beyond. Cryptology ePrint Archive, Paper 2019/1436 (2019), `https://eprint.iacr.org/2019/1436` 3, 21, 22, 23, 24
20. Kirchner, P., Espitau, T., Fouque, P.: Fast reduction of algebraic lattices over cyclotomic fields. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020. pp. 155–185 (2020) 3
21. Kirchner, P., Fouque, P.: Revisiting lattice attacks on overstretched NTRU parameters. In: Advances in Cryptology - EUROCRYPT 2017. pp. 3–26 (2017) 2, 10
22. Kirshanova, E., May, A., Nowakowski, J.: New NTRU Records with Improved Lattice Bases. In: Post-Quantum Cryptography. pp. 167–195. Springer Nature Switzerland, Cham (2023) 10
23. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75**(3), 565–599 (2015) 31
24. Lee, C., Pellet-Mary, A., Stehlé, D., Wallet, A.: An LLL algorithm for module lattices. In: Advances in Cryptology – ASIACRYPT 2019. pp. 59–90 (2019) 3, 6, 9, 19, 25, 26, 27
25. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische annalen **261**(ARTICLE), 515–534 (1982) 8
26. Micciancio, D.: CS 206A: Lattice algorithms and applications. Minkowski's theorem. `https://cseweb.ucsd.edu/classes/wi16/cse206A-a/lec2.pdf` (2016), accessed: 2024-05-27 8
27. Nguyen, P., Stehlé, D.: An LLL algorithm with quadratic complexity. SIAM J. Comput. **39**, 874–903 (01 2009) 6, 7, 19, 20
28. Pataki, G., Tural, M.: On sublattice determinants in reduced bases. arXiv preprint arXiv:0804.4014 (2008) 2, 10, 14
29. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**, 201–224 (1987) 2, 3, 8
30. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In: STACS 2003: 20th Annual Symposium on Theoretical Aspects of Computer Science Berlin, Germany, February 27–March 1, 2003 Proceedings 20. pp. 145–156. Springer (2003) 8
31. The FPLLL development team: fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.1 (2023), available at `https://github.com/fplll/fpylll` 8, 29, 30, 33
32. Washington, L.C.: Introduction to cyclotomic fields, Graduate Texts in Mathematics, vol. 83. Springer-Verlag, New York (1982) 5, 6, 28