THESIS

GR-NTRU: UNDERSTANDING THE SECURITY OF LATTICE-BASED CRYPTOSYSTEMS THROUGH GROUP RINGS

Submitted by

Kirk L. Bonney

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2022

Master's Committee:

Advisor: Dr. Rachel Pries

Dr. Jamie Juul
Dr. Indrajit Ray

ABSTRACT

GR-NTRU: UNDERSTANDING THE SECURITY OF LATTICE-BASED CRYPTOSYSTEMS THROUGH GROUP RINGS

Originally conceived in 1996 by authors Hoffstein, Pipher, and Silverstein, the N^{th}-degree Truncated Ring Unit (NTRU) cryptosystem rivals common cryptosystems such as RSA in terms of speed and security. In pursuit of a deeper understanding of NTRU, we explore a generalization of the cryptosystem using group rings, known as GR-NTRU. This perspective allows for the formulation of a new kind of attack on NTRU-like cryptosystems. In particular, via representation theory, one can decompose a group ring into smaller matrix rings. This decomposition can greatly impact the computational complexity of lattice-based attacks on NTRU-like cryptosystems. We present a summary of how this attack affects GR-NTRU for certain classes of groups, and we end with a detailed example for the group S_3.

DEDICATION

To my partner, Romana, and my cat, Muffin.

TABLE OF CONTENTS

Chapter 1

Introduction

In 1994, Peter Shor shook the world of computer security with the publication of *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer* [22]. This seminal paper implied that when quantum computing becomes more feasible, the computationally hard problems that underpinned common cryptosystems, such as RSA, elliptic curve cryptography (ECC), and Diffie-Helman key exchange (DHKE), would become trivial. This called attention to quantum-resilient cryptosystems.

A cryptosystem fails to be quantum-resilient once we find a quantum algorithm that trivializes its security. As mentioned above, RSA is no longer quantum-resilient as there exist theoretical algorithms which can factor integers with incredible speed, rendering RSA obsolete in a world with robust quantum computers. Fortunately for now, working quantum computers are still limited in their computational capacity. Two of the largest quantum computers as of 2021 are Google's Sycamore with 53 qubits and Zuchongzi of the University of Science and Technology in China with 56 qubits, while one of the most efficient proposed circuits for Shor's algorithm requires $2n + 3$ qubits to factor an n-bit number [3]. Common encryption key lengths for RSA are 2048 and 4096 bits, so quantum computers cannot threaten their security yet.

We introduce the NTRU cryptosystem as one of the most promising, quantum-resilient cryptosystems. Authors Hoffstein, Pipher, and Silverstein published *NTRU: A Ring-Based Public Cryptosystem* in 1998 [7]. The acronym stands for N^{th}-degree Truncated Polynomial ring Unit, referencing the underlying algebraic stucture, $\mathbb{Z}[x]/(x^N - 1)$. Within the past decade, the cybersecurity community has recognized NTRU as a cryptosystem fit for practical application. The cryptosystem has also been selected as a finalist for NIST's post-quantum cryptography standardization effort in 2020 [1]. However, NTRU is not only an excellent post-quantum candidate for computer security, it also outperforms standard cryptosystems in the present digital realm. For example, a message of

block length N takes $\mathcal{O}(N^2)$ operations to ecrypt or decrypt in the NTRU cryptosystem, which is a significant improvement over the $\mathcal{O}(N^3)$ operations needed in RSA [7].

We will establish the necessary background for understanding NTRU in the context of convolution polynomials as well as lattices. After this, we will establish the procedures of the NTRU cryptosystem and look at some of the basic attacks against it. Afterwards, we will turn our attention towards variants of NTRU. We will see that NTRU is algebraically flexible; the basic formulation of the cryptosystem still works when we make alterations to the underlying ring.

Our focus is the GR-NTRU variant, which generalizes NTRU in terms of group rings. One can realize the convolution polynomials, $\mathbb{Z}[x]/(x^N - 1)$, as the group ring $\mathbb{Z}[C_N]$, where C_N is the cyclic group of size N. GR-NTRU provides the framework to develop a cryptosystem for $\mathbb{Z}[G]$, for any group G. While this perspective alone is a great contribution to the overall understanding of NTRU-like cryptosystems, it also reveals a new kind of attack on NTRU-like cryptosystems that leverages the theory of representations of the group G. We will conclude with a discussion of examples of this attack on specific groups.

Chapter 2

Background

2.1 Convolution Polynomial Rings

The core algebraic structures of the NTRU cryptosystem are convolution polynomial rings.

Definition 2.1.1. Let N be a positive integer. The *ring of convolution polynomials of rank* N is the quotient ring

$$R = \frac{\mathbb{Z}[x]}{(x^N - 1)}.$$

Further, the *ring of convolution polynomials of rank* N *modulo* p is the quotient ring

$$R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[x]}{(x^N - 1)}.$$

It is helpful to view these rings as $\mathbb{Z}$-modules where we may identify an element

$$a(x) = \sum_{i=0}^{N-1} a_i x^i \in R \text{ or } R_p,$$

with the vector of coefficients $(a_0, \ldots, a_{N-1})$. When we intend to view a convolution polynomial $a(x)$ as a vector, we write $\mathbf{a}$.

The name *convolution* comes from the ring multiplication, which is commonly refered to as the convolution of polynomials and denoted by $\star$.

Proposition 2.1.1. *The product of two polynomials* $\mathbf{a}(x)$, $\mathbf{b}(x) \in R$ *is given by the formula*

$$a(x) \star b(x) = c(x) \textit{ with } c_k = \sum_{i+j \equiv k \bmod N} a_i b_{k-i}.$$

In vector form we can write this as

$$\mathbf{a} \star \mathbf{b} = (a_0, \ldots, a_{N-1}) \star (b_0, \ldots, b_{N-1}) = (c_0, \ldots, c_{N-1}),$$

with coefficients c_k given by the above proposition.

Example 2.1.1. Set $N = 11$ and let

$$f(x) = 1 + 2x^1 + 3x^4 + 5x^5 + 9x^6 + 6x^7 + 6x^8,$$

and

$$g(x) = 3x^1 + 2x^2 + 2x^4 + 2x^5 + 5x^8 + x^9.$$

Then,

$$f(x) \star g(x) = 30 + 42x^1 + 48x^2 + 54x^3 + 41x^4 + 51x^5 + 31x^6 + 37x^7 + 47x^8 + 57x^9 + 42x^{10}.$$

We have a natural homomorphism from R to R_p by reducing the coefficients modulo p.

$$\phi : R \to R_p \text{ defined by } \phi\left((a_0, \ldots, a_{N-1})\right) = (\bar{a}_0, \ldots, \bar{a}_{N-1}).$$

Where $\bar{a_i}$ denotes reduction of an integer modulo p.

We may invert this process via a lifting map $\psi : R_p \to R$. This map, however, is not natural and involves a choice of how we bring the coefficients back. In the decryption process of NTRU, it is desirable to lift an element of R_p to R by taking coefficients to have smallest absolute value.

Definition 2.1.2. Let $a(x) \in R_p$. The *center-lift* of $\mathbf{a}(x)$ to R is the unique polynomial $a'(x) \in R$ satisfying

$$a'(x) \bmod p = a(x)$$

whose coefficients are chosen in the interval

$$-\frac{p}{2} < a_i' \leq \frac{p}{2}.$$

Example 2.1.2. Set $N = 5$ and let

$$f(x) = 8 - 2x^1 - 5x^2 + 6x^3 + 11x^4.$$

We reduce f modulo $p = 3$,

$$\phi(f(x)) \equiv \bar{f}(x) \equiv 2 + x^1 + x^2 + 2x^4 \bmod 3 \in R_3.$$

Lifting $\bar{f}$ back to R,

$$\psi(\bar{f}(x)) = -1 + x^1 + x^2 - x^4$$

It is worth noting that the lifting map does not act as an inverse. However, if the coefficients of the polynomial were to already be in the interval $(-p/2, p/2]$, it would return the polynomial one began with.

Inverting polynomials modulo p is important for generating public/private key pairs in NTRU. The following gives necessary and sufficient conditions for an element of R_p to have a multiplicative inverse.

Proposition 2.1.2. *Let q be prime. Then $a(x) \in R_q$ has a multiplicative inverse if and only if*

$$\gcd(a(x), x^N - 1) = 1 \textit{ in } (\mathbb{Z}/q\mathbb{Z})[x].$$

When it exists, we can find it using the Extended Euclidean algorithm for polynomials.

Example 2.1.3. Let $N = 7$ and $p = 3$. We compute the inverse of $a(x) = x^3 + 2x^2 + 1$ in R_3. First, compute $\gcd(a(x), x^3 - 1)$ modulo 3. Proceeding with the Euclidean algorithm,

$$x^7 + 2 \equiv a(x) \cdot \left(x^4 + x^3 + x^2 + 2\right) + x^2 \bmod 3$$
$$a(x) \equiv \left(x^2\right) \cdot (x + 2) + 1 \bmod 3$$

so $\gcd(a(x), x^7 - 1) = 1$. This allows us to conclude that $a(x)$ is invertible in R_2. Via the substitution method of the extended Eucldiean algorithm we obtain,

$$1 \equiv (2x + 1) \cdot \left(x^7 + 2\right) + a(x) \cdot \left(x^5 + 2x^2 + 2x + 2\right) \bmod 3.$$

So the inverse of $a(x)$ in R_3 is $x^5 + 2x^2 + 2x + 2$.

In the NTRU cryptosystem, it is useful to have convolution polynomials with coefficients as small as possible.

Definition 2.1.3. Let d_1 and d_2 be positive integers, we let

$$\mathcal{T}(d_1, d_2) = \left\{ \begin{array}{rl} & a(x) \text{ has } d_1 \text{ coefficients equal to } 1 \\ a(x) \in R: & a(x) \text{ has } d_2 \text{ coefficients equal to } -1 \\ & a(x) \text{ has all other coefficients equal to } 0 \end{array} \right\}.$$

This set is referred to as the *ternary* polynomials, in reference to the three possible choices for each coefficient.

Example 2.1.4. Let $d_1 = d_2 = 2$. Then

$$a(x) = 1 - x - x^3 + x^5$$

is an element of $\mathcal{T}(d_1, d_2)$ as it has two coefficients equal to 1, two coefficients equal to -1, and the rest are 0.

The following gives a formula for the size of $\mathcal{T}(d_1, d_2)$ in terms of d_1, d_2, and N.

$$|\mathcal{T}(d_1, d_2)| = \binom{N}{d_1}\binom{N-d_1}{d_2} = \frac{N!}{d_1!d_2!(N-d_1-d_2)!}$$

2.2 Lattices

Lattices share many of the definitions and properties that one encounters in studying finite-dimensional vector spaces. However, the aspects in which lattices depart from vector spaces are exactly those that NTRU-like cryptosystems can leverage for hard cryptographical problems. For example, there are no shortest non-zero vectors in a vector space, whereas in lattices there are such objects.

Definition 2.2.1. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ be a set of linearly independent vectors. The lattice L generated by $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is the set of linear combinations of $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with coefficients in $\mathbb{Z}$,

$$L = \{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_n\mathbf{v}_n \mid a_i \in \mathbb{Z}\}.$$

As is the case for vector spaces, a *basis* for a lattice L is any set of $\mathbb{Z}$-linearly independent vectors that generate L. All possible bases for L have the same number of elements, and we call this number the *dimension*[1] of L and denoted by $\dim L$.

Proposition 2.2.1. *Any two bases for a lattice L are related by a matrix having integer coefficients and determinant equal to ± 1.*

Definition 2.2.2. An *integral lattice* is a lattice whose vectors all have integer coordinates. Equivalently, an integral lattice is an additive subgroup of $\mathbb{Z}^n$ for some positive integer n.

For the purposes of our discussion, we only need to consider integral lattices where $\dim L = n$. From this point onwards, we assume our lattices have these properties.

[1] Also known as *rank*.

Given an ordered basis $\mathcal{B} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of an integral lattice L, we can construct the $n \times n$ matrix M whose i^{th} row is $\mathbf{v}_i$. In this way, a different basis for L can be realized by left multiplication by an $n \times n$ matrix such as those in proposition 2.2.1.

Example 2.2.1. Let L be the lattice generated by the basis

$$\mathcal{B}_1 = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

The lattice L also has basis

$$\mathcal{B}_2 = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right\}.$$

One can calculate that these two matrices are related by

$$M = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

More precisely, $\mathcal{B}_2 = M\mathcal{B}_1$.

A lattice $L \subset \mathbb{R}^n$ inherits the usual Euclidean norm from $\mathbb{R}^n$. That is, if $\mathbf{v} = (v_1, \ldots, v_n) \in L$, where $\mathbf{v}$ is written with respect to the standard basis,

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \cdots + v_n^2}.$$

The following gives an alternate characterization of a lattice.

Definition 2.2.3. A non-empty subset L of $\mathbb{R}^n$ is an *additive subgroup* if it is closed under addition and subtraction. It is called a *discrete additive subgroup* if there is a positive constant $\epsilon > 0$ with the following property: for each $\mathbf{v} \in L$,

$$L \cap \{\mathbf{w} \in \mathbb{R}^n \ : \ \|\mathbf{v} - \mathbf{w}\| < \epsilon\} = \{\mathbf{v}\}.$$

Theorem 2.2.1. *A subset of $\mathbb{R}^m$ is a lattice if and only if it is a discrete additive subgroup.*

The next definition introduces an important tool for understanding a lattice with respect to a particular basis.

Definition 2.2.4. Let L be a lattice of dimension n and let $\mathcal{B} = \mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for L. The *fundamental domain* for L corresponding to $\mathcal{B}$ is the set

$$\mathcal{F}(\mathcal{B}) = \{t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n \ : \ 0 \leq t_i < 1\}$$

One of the valuable properties of the fundamental domain is the following.

Proposition 2.2.2. *Let $L \subset \mathbb{R}^n$ be a lattice of dimension n and let $\mathcal{F}$ be a fundamental domain for L. Then every vector $\mathbf{w} \in \mathbb{R}^n$ can be written in the form*

$$\mathbf{w} = \mathbf{t} + \mathbf{v} \text{ for a unique } \mathbf{t} \in \mathcal{F} \text{ and a unique } \mathbf{v} \in L.$$

Equivalently, the union of the translated fundamental domains

$$\bigcup_{\mathbf{v} \in L} (\mathcal{F} + \mathbf{v}),$$

where

$$\mathcal{F} + \mathbf{v} = \{\mathbf{t} + \mathbf{v} \ : \ \mathbf{t} \in \mathcal{F}\},$$

covers $\mathbb{R}^n$,

A useful invariant of the fundamental domain over varying bases is volume. That is, given two bases $\mathcal{B}$ and $\mathcal{P}$ for a lattice L, the volumes of the two parallelepipeds $\mathcal{F}(\mathcal{B})$ and $\mathcal{F}(\mathcal{P})$ are equal. This can be seen as a consequence of proposition 2.2.1, as the determinant of any change of basis matrix will be ± 1.

Definition 2.2.5. Let L be a lattice of dimension n and let $\mathcal{F}$ be a fundamental domain for L. Then the n-dimensional volume of $\mathcal{F}$ is called the *determinant of* L. It is denoted by $\det(L)$.

The following result gives a useful upper bound for the volume of a fundamental domain.

Proposition 2.2.3 (Hadamard's Inequality). *Let L be a lattice, let $\mathcal{B} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ be a basis for L, and let $\mathcal{F}$ be a fundamental domain for L. Then*

$$\det L = \mathit{Vol}(\mathcal{F}) \leq \|\mathbf{v}_1\| \cdots \|\mathbf{v}_n\|.$$

The key fact is that the closer the basis $\mathcal{B}$ is to being orthogonal, the closer the inequality is to being an equality.

Proposition 2.2.4. *Let $L \subset \mathbb{R}^n$ be a lattice of dimension n, let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be a basis for L. Let M be the matrix whose rows are the basis vectors. Then the volume of $\mathcal{F}$ is given by*

$$\mathit{Vol}(\mathcal{F}(\mathcal{B})) = |\det(M)|.$$

2.3 Introduction to SVP and CVP

In this section, we introduce and analyze two fundamental problems of lattices.

- Shortest Vector Problem (SVP): Given a lattice L, find a nonzero $\mathbf{v} \in L$ that minimizes $\|\mathbf{v}\|$.
- Closest Vector Problem (CVP): Given a lattice L and an element $\mathbf{w} \in \mathbb{R}^n$, find $\mathbf{v} \in L$ that minimizes $\|\mathbf{v} - \mathbf{w}\|$ with $\mathbf{v} \neq \mathbf{w}$.

Note that SVP is just a special case of CVP with $\mathbf{w} = \mathbf{0}$. The following theorem of Hermite provides an upper bound for the smallest vector in a lattice in terms of its dimension and determinant.

Theorem 2.3.1 (Hermite's Theorem). *Every lattice $L \subset \mathbb{R}^n$ of dimension n contains a nonzero vector $\mathbf{v} \in L$ satisfying*

$$\|\mathbf{v}\| \leq \sqrt{n} \det(L)^{\frac{1}{n}}.$$

Hermite's constant γ_n is the smallest value such that every lattice L with dimension n contains a nonzero vector $\mathbf{v} \in L$ where

$$\|\mathbf{v}\|^2 \leq \gamma_n \det(L)^{2/n}.$$

So by Hermite's theorem, we have that $\gamma_n \leq n$. Surprisingly, this constant is only known for $1 \leq n \leq 8$ and $n = 24$. For example, in the case where $n = 4$, we have that $\gamma_n = \sqrt{2}$. This useful bound given by Hermite's theorem is a corollary of the following important result in the theory of lattices.

Theorem 2.3.2 (Minkowski's Theorem). *Let $L \subset \mathbb{R}^n$ be a lattice of dimension n and let $S \subset \mathbb{R}^n$ be a bounded symmetric convex set whose volume satisfies*

$$\mathit{Vol}(S) > 2^n \det(L).$$

Then S contains a nonzero lattice vector. If S is also closed, it suffices to take $\mathit{Vol}(S) \geq 2^n \det(L)$.

Improvements on this bound can be made by involving the gamma function $\Gamma(s)$, which we define by

$$\Gamma(s) = \int_0^\infty x^{s-1} e^{-x} dx.$$

We also have the following useful estimate on $\Gamma(s)$ due to Stirling.

$$\ln \Gamma(1+s) = \ln(s/e)^s + \frac{1}{2}\ln(2\pi s) + \mathcal{O}(1) \text{ as } s \to \infty.$$

Exact bounds for a shortest vector $\mathbf{v}_{\text{shortest}}$ of L for large n are difficult to obtain, but the following heuristic gives us a reasonable way to estimate them: the number of lattice points in a ball centered at $\mathbf{0}$ should approximately be the volume of the ball divided by the volume of a fundamental domain.

Theorem 2.3.3. *Let $\mathbb{B}_R(\mathbf{a})$ be the ball of radius R centered at $\mathbf{a} \in \mathbb{R}^n$. Then the volume of $\mathbb{B}_R(\mathbf{a})$ is*

$$\mathit{Vol}(\mathbb{B}_R(\mathbf{a})) = \frac{\pi^{n/2} R^n}{\Gamma(1 + n/2)}.$$

Via Stirling's formula, when n is large, the volume of the ball $\mathbb{B}_R(\mathbf{a})$ is approximated by

$$\mathit{Vol}(\mathbb{B}_R(\mathbf{a})) \approx \sqrt{\frac{2\pi e}{n}} R.$$

Using this formula for the volume of an n-ball, we introduce the Gaussian expected shortest length.

Definition 2.3.1. Let L be a lattice of dimension n. The *Gaussian expected shortest length* is

$$\sigma(L) = \sqrt{\frac{n}{2\pi e}} (\det L)^{\frac{1}{n}}.$$

The *Gaussian heuristic* states that a shortest nonzero vector in a random lattice will satisfy

$$\|\mathbf{v}_{\text{shortest}}\| \approx \sigma(L).$$

In practice, the Gaussian heuristic is useful for understanding the difficulty of finding a shortest vector. If a true shortest vector v in L is much smaller than $\sigma(L)$, then it turns out that lattice reduction algorithms are more typically more successful in finding v.

Babai's Algorithm

This section introduces Babai's Algorithm. We will indicate the motivation for the algorithm here, but details are omitted.

Babai's algorithm is based on the relatively simple idea that the elements of a lattice L that are closest to a given vector $\mathbf{w} \in \mathbb{R}^n$ are likely to be vertices of the translate of the fundamental region which contains $\mathbf{w}$. This idea is a fact when the fundamental region is built from orthogonal basis

vectors. For example, consider the case of $\mathbb{R}^2$ and L given as the square integer lattice. To find the closest vector to $\mathbf{w} \in \mathbb{R}^2$, simply find which grid square $\mathbf{w}$ lies in, and test the four corners.

Given an orthogonal basis $\mathcal{B} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ for the lattice $L \subset \mathbb{R}^n$, we may calculate the length of a vector in L via

$$\|a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n\|^2 = a_1^2\|\mathbf{v}_1\| + \cdots + a_n^2\|\mathbf{v}_n\|.$$

Thus, in the case where $\mathbf{w} = \mathbf{0}$, we may find the solution to the SVP by taking the shortest vectors of the set $\{\pm\mathbf{v}_1, \ldots, \pm\mathbf{v}_n\}$. However, the reliability of the statement decreases as the given basis becomes less orthogonal. For this reason, we recall Hadamard's inequality from proposition 2.2.3. From this inequality, we define the *Hadamard ratio*[2] of the basis $\mathcal{B}$ to be

$$\mathcal{H}(\mathcal{B}) = \left(\frac{\det L}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|\cdots\|\mathbf{v}_n\|}\right)^{1/n}.$$

The possible values of $\mathcal{H}(\mathcal{B})$ lie between 0 and 1, where 0 indicates linear dependence, and 1 indicates orthogonality. Through this ratio, we can decide on a threshold of orthogonality. For example, we could say that a basis $\mathcal{B}$ is sufficiently orthogonal if $0.9 \leq \mathcal{H}(\mathcal{B}) \leq 1$.

The idea of Babai's algorithm is to first find a sufficiently orthogonal basis for L, and then search among the vertices $\{\pm\mathbf{v}_1, \ldots, \pm\mathbf{v}_n\}$ of $\mathcal{F}(\mathcal{B})$ for a shortest vector.

2.4 Group Ring Background

Group rings are a natural generalization of a vector space. Our definition of group ring will generalize in two ways: we allow the scalars to be a ring and the 'vectors' will be the elements of some chosen group.

Definition 2.4.1. Let R be a ring and G be a group. We define the *group ring* $R[G]$ to be the free R-module with basis G where multiplication is defined distributively in accordance with the

[2]The reciprocal is known as the *orthogonality defect*.

inherent multiplication of G. That is,

$$\left(\sum_{g\in G} a_g \cdot g\right) \cdot \left(\sum_{h\in G} b_h \cdot h\right) = \sum_{g,h\in G} (a_g b_h) \cdot (gh) = \sum_{f\in G} c_f \cdot f,$$

where

$$c_f = \sum_{gh=f} a_g b_h = \sum_{g\in G} a_g b_{g^{-1}f}.$$

Addition is likewise defined in the natural way,

$$\left(\sum_{g\in G} a_g \cdot g\right) + \left(\sum_{g\in G} b_g \cdot h\right) = \sum_{g\in G} a_g b_g.$$

We can also write elements of a group ring as a tuple when the ordering on the elements of G is clear. For example, if $G = C_N$, the cyclic group of size N with generator λ, then we may identify $a = \sum_k^N a_k \cdot \lambda^k$ with the tuple $\mathbf{a} = (a_1, \ldots, a_N)$.

In this paper we will be primarily concerned with the case where $R = \mathbb{Z}$. This tethers GR-NTRU to the original NTRU, as it relates the cryptosystem to the realm of integral lattices.

Group rings play a central role in the theory of group representations. This connection is important for analyzing the security of GR-NTRU later on. We present a summary of group representations and the relationship of the theory to group rings.

Definition 2.4.2. Let G be a group and K be a field. A *representation* of G over K of degree n is a group homomorphism

$$\sigma : G \to \mathrm{GL}(K, n).$$

Not surprisingly, representations are the key object of study in representation theory. A representation σ defines an action of G on the vector space $V = K^n$ via

$$g \cdot v = \sigma(g)v$$

where $g \in G$ and $v \in V$. Also, given an action of G on a vector space V we may obtain a representation of G. When the context is clear, we may refer to such a representation as V. What follows are some key definitions relating to a represntation V.

Definition 2.4.3. Let V be a representation of G of degree n.

- If the action of G on V fixes a subspace W of V, we call W a *subrepresentation* of V.
- If V only has itself and 0 as representations, we call V *irreducible*.
- Given representations V and W of G, if there exists a vector space isomorphism $\psi : V \to W$ such that

$$\psi(g \cdot v) = g \cdot \psi(v)$$

for all $g \in G$ and $v \in V$, we say that V and W are *isomorphic* as representations of G.

Now we use these definitions to state two key facts about representations. First we give a relationship between irreducible representations of G and conjugacy classes of G.

Proposition 2.4.1. *Let G be a finite group. Then the number of irreducible representations of G is equal to the number of conjugacy classes in G.*

Next, we describe the correspondence between representations of a group G and group rings.

Proposition 2.4.2. *A group representation $\sigma : G \to GL(K, n)$ can be naturally extended to a ring homomorphism of group rings $\bar{\sigma} : K[G] \to \mathbf{M}_n(K)$. Likewise, a ring homomorphism $\bar{\sigma} : K[G] \to \mathbf{M}_n(K)$ provides a group representation $\sigma : G \to GL(K, n)$ via restriction.*

Chapter 3

NTRU

3.1 The encryption/decryption process

The NTRU public key cryptosystem uses convolution polynomial rings to securely transfer ciphertexts. In this section we describe the mathematics of this scheme.

First, recall that for a fixed N, we defined R to be the ring of convolution polynomials $\mathbb{Z}[x]/(x^N - 1)$ and defined R_p as $(\mathbb{Z}/p\mathbb{Z})[x]/(x^N - 1)$. We also defined the set $\mathcal{T}(d_1, d_2)$ by,

$$\mathcal{T}(d_1, d_2) = \left\{ a(x) \in R : \begin{array}{r} a(x) \text{ has } d_1 \text{ coefficients equal to } 1 \\ a(x) \text{ has } d_2 \text{ coefficients equal to } -1 \\ a(x) \text{ has all other coefficients equal to } 0 \end{array} \right\}.$$

Elements of $\mathcal{T}(d_1, d_2)$ are important for the efficiency of NTRU. Their simple structure allows for quick computations of the convolution product. The small coefficients also allow us to control the size of the convolution polynomials computed throughout encryption and decryption, which is important for avoiding decryption failure.

Set-up: To begin, public parameters (N, p, q, d) are chosen so that N and p are prime, $\gcd(N, q) = 1$, and $q > (6d + 1)p$.

Key Creation: Alice does the following

- chooses a private $\mathbf{f} \in \mathcal{T}(d + 1, d)$, which is invertible in both R_q and R_p.
- computes inverses $\mathbf{f}_q$ and $\mathbf{f}_p$ in R_q and R_p respectively.
- chooses a private $\mathbf{g} \in \mathcal{T}(d, d)$.
- Publishes public key $\mathbf{h} = \mathbf{f}_q \star \mathbf{g}$.

Encryption: Bob does the following

- Chooses plaintext $\mathbf{m} \in R_q$ so that its coefficients satisfy $-\frac{p}{2} < m_i \leq \frac{p}{2}$.
- Chooses a random $\mathbf{r} \in \mathcal{T}(d, d)$.
- Uses Alice's public key $\mathbf{h}$ to compute $\mathbf{e} \equiv p\mathbf{h} \star \mathbf{r} + \mathbf{m} \bmod q$.
- Sends ciphertext $\mathbf{e}$ to Alice.

Decryption: Alice does the following

- Computes $\mathbf{f} \star \mathbf{e} \equiv p\mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m} \bmod q$.
- Takes the center-lift of $\mathbf{f} \star \mathbf{e}$, call it $\mathbf{a}$, and computes $\mathbf{m} \equiv \mathbf{f}_p \star \mathbf{a} \bmod p$.

Next we follow an example from *Introduction to Mathematical Cryptography* [8].

Example 3.1.1. Choose public parameters $(N, p, q, d) = (7, 3, 41, 2)$, which satisfy $q > (6d+1)p$. Then, following the above process

First the key creation,

- Alice chooses $f(x) = x^6 - x^4 + x^3 + x^2 - 1 \in \mathcal{T}(3, 2)$ and $g(x) = x^6 + x^4 - x^2 - x \in \mathcal{T}(2, 2)$.
- Then Alice finds the inverses

$$f_q(x) = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37 \in R_q,$$
$$f_p(x) = x^6 + 2x^5 + x^3 + x^2 + x + 1 \in R_p.$$

- Alice keeps $(f(x), f_p(x))$ as the private key.
- Alice publishes the public key

$$h(x) = f_q(x) \star g(x) = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30 \in R_q.$$

Next, Bob uses the public key to encrypt the plaintext

- Bob wants to send Alice the plaintext

$$m(x) = -x^5 + x^3 + x^2 - x + 1.$$

- Bob chooses the random element

$$r(x) = x^6 - x^5 + x - 1.$$

- Using **r**, Bob sends Alice the ciphertext

$$e(x) \equiv pr(x) \star h(x) + m(x) \equiv 31x^6 + 19x^5 + 4x^4 + 2x^3 + 40x^2 + 3x + 25 \bmod q.$$

Finally, Alice decrypts the ciphertext.

- Alice computes

$$f(x) \star e(x) \equiv x^6 + 10x^5 + 33x^4 + 40x^3 + 40x^2 + x + 40 \bmod q.$$

- Alice takes the center lift of $f(x) \star e(x)$ to obtain

$$a(x) = x^6 + 10x^5 - 8x^4 - x^3 - x^2 + x - 1 \in R.$$

- To finish the decryption, Alice reduces $a(x)$ modulo p and computes

$$f_p(x) \star a(x) \equiv 2x^5 + x^3 + x^2 + 2x + 1 \bmod p.$$

Taking the center lift of the above modulo p will return the plaintext $m(x)$.

3.2 Proof of Decryption

We outlined a summary of the mathematical procedure of NTRU, but we still need to verify that Alice's decryption process really gives back the orginal plaintext.

Theorem 3.2.1. *Given NTRU parameters* (N, p, q, d) *where*

$$q > (6d + 1)p, \tag{3.1}$$

then, in the language of the procedure above, $\mathbf{f}_p \star \mathbf{a} \equiv \mathbf{m} \bmod p$.

Proof. We start by unpacking the calculation of $\mathbf{a}$,

$$\begin{aligned}
\mathbf{a} &\equiv \mathbf{f} \star \mathbf{e} \bmod q \\
&\equiv \mathbf{f} \star (p\mathbf{h} \star \mathbf{r} + \mathbf{m}) \bmod q \\
&\equiv p\mathbf{f} \star \mathbf{f}_q \star \mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m} \bmod q \\
&\equiv p\mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m} \bmod q.
\end{aligned}$$

Next, in R we examine

$$p\mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m}. \tag{3.2}$$

Note first that $\mathbf{g}, \mathbf{r} \in \mathcal{T}(d, d)$, so that the maximum coefficient of their convolution product is $2d$. Likewise, $\mathbf{f} \in \mathcal{T}(d+1, d)$ and $\mathbf{m}$ was chosen to be a center lift of a polynomial - i.e. its coefficients satisfy $-\frac{p}{2} < m_i \leq \frac{p}{2}$ - so that the largest possible coefficient of $\mathbf{f} \star \mathbf{m}$ is $(2d + 1) \cdot \frac{p}{2}$. Then the largest coefficient of (3.2) has magnitude at most

$$p \cdot 2d + (2d + 1) \cdot \frac{p}{2} = \left(3d + \frac{1}{2}\right) p.$$

Now (3.1) becomes relevant. This bound ensures that the coefficients of (3.2) have magnitude strictly smaller than $q/2$, so that computing modulo q, rather than in R, and then taking the center

lift will result in the same thing,

$$\mathbf{a} = p\mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m}. \tag{3.3}$$

The rest simply follows from unfolding the calculation of $\mathbf{F}_p \star \mathbf{a}$ modulo p,

$$\begin{aligned}
\mathbf{f}_p \star \mathbf{a} &\equiv \mathbf{f}_p \star (p\mathbf{g} \star \mathbf{r} + \mathbf{f} \star \mathbf{m}) \bmod p \\
&\equiv \mathbf{f}_p \star \mathbf{f} \star \mathbf{m} \\
&\equiv \mathbf{m}.
\end{aligned}$$

□

This proof enlightens us as to why certain choices are made in the selection of various pieces of NTRU. We involve the ternary polynomials $\mathcal{T}(d_1, d_2)$, because their coefficients let us effectively control the size of the coefficients, and by choosing q large enough we ensure that (3.3) holds in the proof.

3.3 Attacks on NTRU

In this section, we state the main mathematical problem posed for an attacker against an NTRU cryptosystem. We then explore some of the basic methods of attack, and describe the ideas behind lattice-based attacks.

Recall that the public key of an NTRU cryptosystem is $\mathbf{h} = \mathbf{f}_q \star \mathbf{g}$.

Definition 3.3.1. The *NTRU Key Recovery Problem* is as follows. Given the convolution polynomial $\mathbf{h}$, find ternary polynomials $\mathbf{f}$ and $\mathbf{g}$ such that $\mathbf{f} \star \mathbf{h} \equiv \mathbf{g} \bmod q$.

When we examine NTRU in the context of lattices, we will see that solving the NTRU Key Recovery Problem is equivalent to solving the SVP in a particular lattice. This equivalence is what guarantees that the hard mathematical problem for NTRU is sufficiently difficult for a practical cryptosystem.

Brute Force Attack

The most rudimentary attack on any cryptosystem is a brute force attack. In a private key attack on NTRU, one can recover the private key by calculating $\mathbf{f} \star \mathbf{h} \bmod q$ for all $\mathbf{f} \in \mathcal{T}(d+1, d)$ and checking if it has small entries or by trying all $\mathbf{g} \in \mathcal{T}(d, d)$ and checking if $\mathbf{g} \star \mathbf{h}^{-1} \bmod q$ has small entries. Since $|\mathcal{T}(d, d)|$ is typically smaller than $|\mathcal{T}(d+1, d)|$, the security of private keys under brute force attacks are determined by $|\mathcal{T}(d, d)|$.

Likewise, a brute force attack on a particular plaintext m can be made by checking if $\mathbf{e} - \mathbf{r} \star \mathbf{h} \bmod q$ has small entries for all $\mathbf{r} \in \mathcal{T}(d, d)$. The security of a particular message is also given by $|\mathcal{T}(d, d)|$.

There is an improvement to the brute force search that are worth mentioning. To describe it, we first define rotations of a convolution polynomial.

Definition 3.3.2. Given a convolution polynomial $\mathbf{f} = (f_1, \ldots, f_N) \in R$, where the rank of R is N, a *rotation* of $\mathbf{f}$ is $x^k \star \mathbf{f}$ where k is any integer.

We remark that for a given $\mathbf{f}$, there are N distinct rotations of $\mathbf{f}$. It follows that if $\mathbf{f}$ and $\mathbf{g}$ give a solution to the problem, then so do $x^k \star \mathbf{f}$ and $x^k \star \mathbf{g}$ for $1 \leq k < N$. However, if $\mathbf{m}$ is a plaintext, then if we use $x^k \star \mathbf{f}$ and $x^k \star \mathbf{g}$ to decrypt, we obtain the rotated plaintext $x^k \star \mathbf{m}$. Thus it suffices to search for solutions upto rotation, and once a solution is found one must decide which rotation of the plaintext is valid. This reduces computations by a factor of $1/N$.

3.4 Lattice-based attacks

A variety of attacks exist for NTRU, however we are only concerned with lattice-based attacks for our discussion. We begin by understanding an NTRU cryptosystem in the context of lattices, where we may carry out an attack on the cryptosystem by solving the SVP for a certain lattice. Once we establish this connection, we can formulate a lattice based attack by finding a sufficiently orthogonal basis and applying Babai's algorithm.

Realizing NTRU with lattices

Begin with an NTRU cryptosystem with parameters (N, p, q, d) and a public key

$$h(x) = h_0 + h_1 x + \cdots + h_{N-1} x^{N-1}.$$

We define the *NTRU lattice* L_h^{NTRU} associated to $h(x)$ to be the $2N-$dimensional lattice spanned by the rows of the following matrix

$$M_h^{\text{NTRU}} = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right).$$

The matrix M_h^{NTRU} can be decomposed into four $N \times N$ blocks

1. The upper left block is the identity matrix, written as I_N.
2. The lower left block is the zero matrix, written as 0_N.
3. The lower right block is q times the identity, written as qI_N.
4. The upper right block are the cyclic shifts of the coefficients of $h(x)$, written as H.

Hence we can compactly write

$$M_h^{\text{NTRU}} = \begin{bmatrix} I_N & H \\ 0_N & qI_N \end{bmatrix}.$$

To accompany this setup, we will identify a pair of polynomials $(a(x), b(x))$ with the $2N$-dimensional vector

$$(a(x), b(x)) = (a_0, \ldots, a_{N-1}, b_0, \ldots, b_{N-1}),$$

where $a(x) = a_0 + a_1 x + \ldots + a_{N-1}x^{N-1}$ and $b(x) = b_0 + b_1 x + \ldots + b_{N-1}x^{N-1}$.

As in the original setup of NTRU, assume that $h(x)$ is constructed using private polynomials $f(x)$ and $g(x)$. We will show that the vector $(f(x), g(x))$ is in the NTRU lattice L_h^{NTRU}.

Proposition 3.4.1. *Assuming that $f(x) \star h(x) \equiv g(x) \bmod q$, let $u(x) \in R$ be such that*

$$f(x) \star h(x) = g(x) + qu(x).$$

Then

$$(f(x), -u(x)) \cdot M_h^{NTRU} = (f, g).$$

Proof. Recall the block form

$$M_h^{\text{NTRU}} = \begin{bmatrix} I & H \\ 0 & qI \end{bmatrix}.$$

Clearly the first N entries of $(f(x), -u(x))M_h^{\text{NTRU}}$ will be f. For the next N entries, when $(f(x), g(x))$ is multiplied by the column which starts with h_k, the result is

$$h_k f_0 + h_{k-1} f_1 + \cdots + h_{k+1} f_{N-1} - qu_k.$$

This is the k^{th} entry of $f(x) \star h(x) - qu(x)$, which we also recognize as $g(x)$. In this way, we have described a linear combination of basis vectors of L_h^{NTRU} which results in $(f(x), g(x))$. □

The following result gives a summary of key information about the lattice formulation of an NTRU cryptosystem.

Proposition 3.4.2. *Let (N, p, q, d) be parameters for an NTRU cryptosystem with the following simplifying assumptions*

$$p = 3,\ d \approx N/3,\ \textit{and}\ q \approx 6pd \approx 2pN.$$

Let L_h^{NTRU} be the NTRU lattice associated to the private key (f, g). Then the following are true

a) $\det(L_h^{NTRU}) = q^N$.

b) $\|f, g\| \approx \sqrt{4d} \approx \sqrt{4N/3} \approx 1.155\sqrt{N}$.

c) The Gaussian heruistic predicts that the shortest nonzero vector in the $NTRU$ lattice has length

$$\sigma(L_h^{NTRU}) \approx \sqrt{Nq/\pi e} \approx 0.838N.$$

Proof. a) This follows from the fact that $M_{\mathbf{h}}^{\text{NTRU}}$ is upper triangular with N entries equal to q and the rest equal to 1. By proposition 2.2.4, we have $\det(M_h^{\text{NTRU}}) = \det(L_h^{\text{NTRU}})$.

b) Both f and g have approximately d coefficients equal to 1, d coefficients equal to -1, and the rest equal to zero, as they are ternary polynomials in the sets $\mathcal{T}(d+1, d)$ and $\mathcal{T}(d, d)$ respectively.

c) This follows immediately from the Gaussian heuristic where the dimension is $2N$.

□

The important consequence of this result is that when the dimension N is large, it is highly likely that the solution to the SVP for the lattice L_h^{NTRU} are $(f(x), g(x))$ and its rotations. We saw that a brute force attack already forces a large choice of N for a cryptosystem to be secure, so in practice the SVP for the lattice associated to an NTRU cryptosystem is solved by $(f(x), g(x))$.

3.5 Solving the SVP for Lattices

Babai's algorithm gave us a strategy for solving the SVP for a lattice L: find a sufficiently orthogonal basis using Hadamard's ratio and search among the the basis vectors for a solution. Now

that we have situated NTRU in terms of lattices, if an attacker can find a sufficiently orthogonal basis for the NTRU lattice, they can carry out an effective attack. One of the fundamental algorithms for finding such a basis is known as LLL Latice reduction, where LLL stands for the authors Lenstra,Lenstra, and Lovász [11]. The algorithm is based off of the method of Gram-Schmidt for finding orthogonal basis in vector spaces. Many improvements of this algorithm exist and we refer the reader to [7] for additional details.

Chapter 4

NTRU Variants

Since 1996, mathematicians and cryptographers have been excited to understand the ideas in NTRU. One aspect of this understanding is accomplished by publishing a great breadth of variants on the scheme. This section seeks to give a condensed survey of some of the most interesting and important NTRU variants that have arised over the past few decades.

4.1 CTRU

Authors Gaborit, Ohler, and Sole published one of the earlier generalizations of NTRU in 2002, replacing the ring of integers $\mathbb{Z}$ with the ring of polynomials in one variable over a finite field $\mathbb{F}_2[T]$ [26]. This main advantage of this approach is avoiding attacks based on the LLL algorithm and the Chinese remainder theorem. While CTRU is an important piece of progress in the study of NTRU-like cryptosystems, unfortunately in avoiding the classic attacks on NTRU, this cryptosystem is highly vulnerable to more elementary attacks based on linear algebra, such as one utilizing the Popov normal form. This cryptosystem also provides no speed advantages over NTRU.

4.2 MaTRU and NNRU

In 2005, Coglianse and Goi suggested MaTRU, a variant of NTRU, with the goal of improving the performance of the algorithm [4]. Here, the algebra of the system is based on $k \times k$ matrices with entries in the ring $R = \mathbb{Z}[X]/(X^n - 1)$. MaTRU shares the same sort of vulnerabilities as NTRU, namely the brute force and lattice-based attacks. When $nk^2 = N$, one can compare MaTRU to NTRU directly, and in this case MaTRU is k times faster in speed.

A few years later in 2009, Nitin Vats would improve on this cryptosystem with NNRU [27]. This crptosystem uses the same algebraic structure as MaTRU, however it changes the decryption to be non-commutative as well, as it is only the encryption of MaTRU which is non-commutative.

This change, among other improvements, makes NNRU not only more resilient to lattice-based attacks, but also makes it significantly faster than NTRU.

4.3 Matrix NTRU

Not to be confused with MaTRU, Matrix NTRU was proposed by Nayak *et al.* in 2008 [14]. This cryptosystem utilizes matrices in the integers, rather than polynomials. It comes with advantages when sending large messages, as well as the benefits of security that come with a non-commutative underlying algebraic structure. In particular, some lattice based attacks cannot be leveraged against non-commutative systems. However, this system is vulnerable to a reaction attack, where an attack can gain information about the private key by sending an purposefully altered ciphertext.

We will see later in the discussion that this cryptosystem plays a key role in a unique attack on GR-NTRU based cryptosystems.

4.4 QTRU

Malekian *et al.* presented a probabalistic and multi-dimensional version of NTRU utilizing quaternion algebras [2]. Similar to MaTRU, the non-commutative nature of this cryptosystem means that lattice attacks such as that of Coppersmith and Shamir do not work against it. Additionally, the more complicated multiplication of the quaternions causes other lattice attacks to work much more slowly. Unfortunately, this also applies to the speed of encrytion as well, so QTRU operates four times slower than NTRU under the same parameters.

4.5 ETRU

In 2009, an NTRU-like cryptosystem utilizing the ring of integers for a number field was introduced by Nevins *et al.* as ETRU [9]. This cryptosystem uses the Eisenstein integers $\mathbb{Z}[\omega]$ in place of the integers. The Eisenstein integers are still an Euclidean domain, allowing for the formulation of geometric attacks analogous to the SVP and CVP. It is slightly faster than NTRU, and has

smaller key sizes for the same or better level of security. ETRU is vulnerable to the usual lattice attacks.

4.6 OTRU

The first non-associative approach to an NTRU variant, OTRU, was proposed by Malekian and Zakerolhosseini in 2010 [13]. This high speed, probablistic, and multi-dimensional cryptosystem is based on the octonion algebra structure. Its standout feature is the ability to ecrypt eight data vectors at each encryption step. It shares the same lattice attack vulnerabilities as NTRU, however, under equal sizes of public and private keys as well as message length, the associated lattice to a OTRU system is eight times larger than NTRU - increasing its security significantly.

4.7 ILTRU

Motivated by ETRU, Atani proposed ILTRU in 2015 [10]. This cryptosystem sought to generalize the success of ETRU by examing NTRU-like systems that use the algebraic structure of $R = \mathbb{Z}[\omega]/\Phi$ with $\Phi = x^n + x^{n-1} + \ldots + x + 1$ where $n+1$ is prime so that Φ is an irreducible cyclotomic polynomial. This ring relates to a variant of the ring learning with errors problem where one can show that ILTRU is CPA-secure. This is a kind of provable security as opposed to the typically heuristic security available for NTRU and other variants.

4.8 Summary

The above collection of variants demonstrates the capacity for many algebraic structures in NTRU-like cryptosystems. Aside from providing alternatives with improved security, as in the case of OTRU, or improved speed, as in the case of MaTRU, these variants also give insight into the security of other cryptosystems. However, when it comes to practical computer security, having one very well-studied cryptosystem is much more valuable than having many less-studied cryptosystems. For this reason, it is valuable to find ways to unify the many cryptosystems we have described. The GR-NTRU cryptosystem offers a generalization of NTRU and many of its

variants. In essence, GR-NTRU provides the framework to form a cryptosystem out of any group ring over the integers. We will see that many NTRU-like cryptosystems can be realized through GR-NTRU for a particular choice of group. From this perspective, analysis on GR-NTRU carries over to many other cryptosystems. In particular, we will describe an attack on GR-NTRU which has the potential to reduce the computational complexity of lattice-based attacks.

Chapter 5

GR-NTRU

In this chapter, we describe the design of the GR-NTRU cryptosystem and look at a special kind of attack carried out via the Matrix NTRU cryptosystem. We also examine which of the variants we discussed can be formulated in the context of GR-NTRU.

In 2015, authors Yasuda, Dahan, and Sakurai published *Characterizing NTRU-Variants Using Group Ring and Evaluating their Lattice Security*, where the GR-NTRU cryptosystem is established and analyzed [25]. The key insight of the GR-NTRU cryptosystem is recognizing that the underlying structure of the NTRU cryptosystem is a group ring, and that for each choice of group one can obtain an analogue of NTRU. For example, the N^{th} degree truncated polyhnomial ring utilized in NTRU is isomorphic to the group ring formed by choosing $G = C_N$, the cyclic group of order N. We will see that the encryption and decryption processes of NTRU do not need the specific choice of C_N to be carried out.

The value of this perspective is that many of the NTRU variants that have been investigated over the past couple of decades can be realized as a GR-NTRU cryptosystem for a particular choice of group. When we allow for twisted group rings, we can describe even more cryptosystems under the GR-NTRU paradigm. This allows us to translate results concerning the GR-NTRU cryptosystem to many other proposed systems with relative ease. Moreover, group rings are intimately tied to the theory of representations, which provides a rich framework for analyzing the security of group ring compatible cryptosystems.

Name	Algebraic Structure	GR-NTRU compatibility
NTRU	$\mathbb{Z}[x]/(x^N - 1)$	GR
CTRU	$\mathbb{F}_2[t][x]/(x^N - 1)$	-
QTRU	Quaternion algebras	TGR
ETRU	Ring of Eisenstein integers $\mathbb{Z}[\omega]$	GR*
OTRU	Octonion algebra	
ILTRU	Ideal lattices	
NTWO	$\mathbb{Z}[x, y]/(x^N, -1, y^N - 1)$	GR
Non-commutative NTRU	$\mathbb{Z}[D_n][x]/(x^N - 1)$	GR

Table 5.1: Summary of NTRU variants.

The labels for GR-NTRU compatibility are as follows: GR indicates that the system can be realized via GR-NTRU, TGR indicates that the system can be realized via GR-NTRU using a twisted group ring, GR* indicates that the system can be realized as a subring of a group ring, and the label - indicates that the system cannot be realized in GR-NTRU.

5.1 Set up

Given a finite group G of size N, we define the subsets $\mathcal{M}$ and $\mathcal{L}(d_1, d_2)$ of the group ring $\mathbb{Z}[G]$. The elements of the set $\mathcal{M}_p$ are the possible messages we can encrypt. We refer to $\mathcal{M}$ as the space of messsages and define it as

$$\mathcal{M}_p = \left\{ m = \sum_{g \in G} a_g \cdot g \in \mathbb{Z}/p\mathbb{Z}[G] \mid -\frac{p}{2} < a_g \leq \frac{p}{2}, \ \forall g \in G \right\}.$$

These are the elements of $\mathbb{Z}/p\mathbb{Z}[G]$ with coefficients centered around 0. For positive integers d_1, d_2,

$$\mathcal{L}(d_1, d_2) = \left\{ f = \sum_{g \in G} a_g \cdot g \in \mathbb{Z}/p\mathbb{Z}[G] \mid \begin{array}{l} f \text{ has } d_1 \text{ coefficients equal to 1,} \\ f \text{ has } d_2 \text{ coefficients equal to -1,} \\ \text{the rest are 0.} \end{array} \right\}.$$

It is useful to view this set as a generalization of the ternary polynomials for convolution polynomial rings.

5.2 GR-NTRU Cryptosystem

A GR-NTRU cryptosystem involes the following parameters: a finite group G, two primes p and q, positive integers d_1, d_2, d_3, an element $f \in \mathcal{L}(d_1, d_1 - 1)$ such that f is a unit when considered as an element of $\mathbb{Z}/p\mathbb{Z}[G]$ and as an element of $\mathbb{Z}/q\mathbb{Z}[G]$, and an element $g \in \mathcal{L}(d_2, d_2)$. We can refer to such a system as a tuple of these parameters: (G, p, q, f, g). This information determines all of the other relevant particulars of a GR-NTRU cryptosystem, such as the public key and private key, aside from the random choice of $r \in \mathcal{L}(d_3, d_3)$.

Key Creation

- Choose $f \in \mathcal{L}(d_1, d_1 - 1)$, $g \in \mathcal{L}(d_2, d_2)$ such that there exists $f_q, f_p \in \mathbb{Z}[G]$ satisfying $f \star f_q \equiv 1 \bmod q$ and $f \cdot f_p \equiv 1 \bmod p$.

- Compute $h = f_q \cdot g \bmod q$.

- The **Public Key** is the tuple (h, p, q).
- The **Private Key** is the pair (f, f_p).

Encryption

Choose a plaintext $m \in \mathcal{M}_p$ for encryption. Next choose a random $r \in \mathcal{L}(d_3, d_3)$. As before, r plays the important role of involving the multiplication operation of the group ring in the encryption, distinguishing the cryptosystem from group-based cryptosystems. Compute the ciphertext

$$c \equiv ph \cdot r + m \bmod q.$$

Decryption

Given a ciphertext c, we first compute

$$a \equiv f \cdot c \bmod q.$$

Next we shift the coefficients of a modulo q so that they lie in the interval $(-q/2, q/2)$. Then the plaintext m can be recovered by computing $f_p \star a \bmod p$.

Example 5.2.1. Consider the group ring $\mathbb{Z}[C_5]$ where C_5 is the cyclic group of size 5 with generator λ. Set $p = 3$ and $q = 73$. Also set $d_1 = 3$, $d_2 = d_3 = 2$. We will carry out an example of the GR-NTRU Cryptosystem using this group ring with the help of Alice and Bob.

First Alice chooses

$$f = \lambda + \lambda^1 + \lambda^2 - \lambda^9 - \lambda^{10} \in \mathcal{L}(3, 2),$$

and

$$g = \lambda^3 + \lambda^5 - \lambda^6 - \lambda^8 \in \mathcal{L}(2, 2).$$

The element f in invertible modulo $p = 3$ and $q = 73$ with inverses

$$f_p = \lambda^0 + 2\lambda^1 + \lambda^2 + \lambda^4 + 2\lambda^5 + 2\lambda^9 + \lambda^{10},$$

and

$$f_q = 10\lambda^0 + 22\lambda^1 + 19\lambda^2 + 32\lambda^3 + 19\lambda^4 + 54\lambda^5 + 16\lambda^6 + 16\lambda^7 + 13\lambda^9 + 19\lambda^{10}.$$

Alice computes

$$h \equiv f \star g \bmod 73 \equiv 72\lambda^1 + 72\lambda^2 + \lambda^4 + 3\lambda^5 + \lambda^6 + \lambda^7 + 71\lambda^8 + 72\lambda^9 + 72\lambda^{10} \bmod 73,$$

and posts (h, p, q) as the public key.

Now Bob wishes to securely send the plaintext

$$m = \lambda^0 + \lambda^2 - \lambda^4 - \lambda^8 + \lambda^9.$$

Bob randomly chooses r,

$$r = -\lambda^1 + \lambda^3 - \lambda^7 + \lambda^{10} \in \mathcal{L}(2, 2),$$

then computes the ciphertext

$$c \equiv ph \cdot r + m \bmod q = 63 + 58\lambda^1 + 69\lambda^2 + 3\lambda^3 + 13\lambda^4 + 70\lambda^6 + 67\lambda^7 + 7\lambda^8 + 6\lambda^{10} \bmod 73.$$

Bob sends the ciphertext to Alice, who decrypts c using the private key.

$$a \equiv -4 + -3\lambda^1 + 3\lambda^2 + -2\lambda^3 + 3\lambda^4 + -2\lambda^5 + 5\lambda^8 + \lambda^9 + -2\lambda^{10} \bmod 73.$$

Then, adjusting the coefficients to lie in the appropriate range, Alice finds

$$a \equiv -4 + -3\lambda^1 + 3\lambda^2 + -2\lambda^3 + 3\lambda^4 + -2\lambda^5 + 5\lambda^8 + \lambda^9 + -2\lambda^{10} \bmod 73.$$

Finally Alice decrypts using,

$$f_p \cdot a \equiv \lambda^0 + \lambda^2 - \lambda^4 - \lambda^8 + \lambda^9 \bmod 3,$$

which matches the original message m Bob intended to send.

5.3 Attacks

Lattice Attacks

An important quality of GR-NTRU is that the lattice formulation of NTRU generalizes as well. Given a group G of order N and a public key $h = (h_{g_1}, \ldots, h_{g_N})$ associated to some $\mathbb{Z}[G]$ cryptosystem (G, p, q, f, g), we can formulate the matrix

$$M(h) = \begin{bmatrix} I_N & \mathcal{C}(h) \\ 0_N & qI_N \end{bmatrix}$$

where we define $\mathcal{C}(h)$[3] by

$$\mathcal{C}(h) = \begin{pmatrix} h_1 & h_{g_1^{-1}g_2} & \cdots & h_{g_1^{-1}g_N} \\ h_{g_2^{-1}g_1} & h_1 & \cdots & h_{g_2^{-1}g_N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{g_N^{-1}g_1} & h_{g_N^{-1}g_2} & \cdots & h_1 \end{pmatrix}.$$

As before, the lattice $L(h)$ generated by the rows of $M(h)$ contains the vector $(f, g) \in \mathbb{Z}^{2n}$, and this vector can be found by solving the SVP for $L(h)$. Thus the lattice-based attacks described for NTRU may also be levied against a generic GR-NTRU cryptosystem.

[3]This is a kind of *circulent* matrix.

Matrix NTRU Attack

To further analyze the security of GR-NTRU cryptosystems, we will expand on the NTRU-like cryptosystem called Matrix NTRU (M-NTRU), which makes use of matrix rings. We then show that GR-NTRU cryptosystems may be reduced to a series of smaller M-NTRU cryptosystems. This decomposition has the potential to drastically reduce the dimensionality involved in attacking GR-NTRU. Essentially, an attack carried out on each of the lower-dimensional cryptosystems can be lifted to an attack on the whole cryptosystem. The computational complexity of attacks on NTRU-like cryptosystems are typically polynomial functions of dimension N, which in the case of GR-NTRU is the size of the group G. Understanding how M-NTRU can break down dimensionality is key to controlling the lower bound of computational intensity.

What follows is an outline of how this works:

- Find an injective ring homomorphism

$$\tau : \mathbb{Z}[G] \to \mathbf{M}_{n_1}(\mathbb{Z}) \bigoplus \mathbf{M}_{n_2}(\mathbb{Z}) \bigoplus \cdots \bigoplus \mathbf{M}_{n_k}(\mathbb{Z}),$$

- Solve the smaller lattice problem for each $1, \ldots, k$ by passing through τ.
- Once the smaller problems are all solved, pull everything back through τ to get a solution to the original cryptosystem.

We will see that the choice of group G greatly impacts the shape of possible injections τ, namely the sizes of the n_i. This amounts to studying the representations of the group G, which we will carry out for a number of examples in the next chapter.

Example 5.3.1. To further understand how this impacts security, consider the situation where $G = C_{10}$. We have the existence of an embedding,

$$\tau : \mathbb{Z}[G] \to \mathbf{M}_1(\mathbb{Z}) \bigoplus \mathbf{M}_1(\mathbb{Z}) \bigoplus \mathbf{M}_4(\mathbb{Z}) \bigoplus \mathbf{M}_4(\mathbb{Z}).$$

As a simplification, we will assume that the LLL algorithm for solving SVP in a lattice of size N is $\mathcal{O}(N^2)$. Then if one were to use LLL to crack a GR-NTRU cryptosystem built with G directly, one expects it to take $c \cdot 20^2$ operations to carry this out, for some positive constant c. Here the 20 comes from 2 times the size of the group G. However, if one first passes through to the smaller matrix rings using τ we see that the small problems in each would take $c \cdot 2^2$, $c \cdot 2^2$, $c \cdot 8^2$, and $c \cdot 8^2$ respectively, for a total of $c \cdot 136$ operations, less than half of the computations needed before the decomposition.

5.4 Matrix NTRU

Let $R = \mathbb{M}_n(\mathbb{Z})$. We define the following subsets of $\mathbb{Z}^n$ similar to those defined in the GR-NTRU cryptosystem. For a chosen positive integer p, define $\mathcal{M}_p$ by

$$\mathcal{M}_p = \left\{ M = [m_{i,j}] \in \mathbb{M}_n(\mathbb{Z}/p\mathbb{Z}) \mid -\frac{p}{2} < m_{i,j} \leq \frac{p}{2}, \forall\, 1 \leq i, j \leq n \right\}.$$

For positive integers d_1, d_2 we also define

$$\mathcal{D}(d_1, d_2) = \left\{ A = (a_{ij}) \in R \mid \begin{array}{c} \text{Each row of } A \text{ has } d_1 \text{ coefficients equal } 1 \\ \text{Each row of } A \text{ has } d_2 \text{ coefficients equal } -1 \\ \text{the rest are } 0. \end{array} \right\}.$$

Key Creation

- Choose positive integers d_1, d_2, d_3.
- Choose $F \in \mathcal{D}(d_1, d_1 - 1),\ G \in \mathcal{D}(d_2, d_2)$ such that there exists $F_q, F_p \in R$ satisfying $F \cdot F_q = 1 \bmod q$ and $F \cdot F_p = 1 \bmod p$.
- Compute $H = F_q \cdot G \bmod q$.
- The **Public Key** is the matrix H.

- The **Private Key** is the pair (F, F_p).

Encryption

To encrypt, first a message M is selected from the set $\mathcal{M}_p$. Then a random $R \in \mathcal{D}(d_3, d_3)$ is also chosen, and the cipher text is computed as

$$C \equiv pH \cdot R + M \bmod q.$$

Decryption

To decrypt the ciphertext c, first compute

$$A \equiv F * C \bmod q.$$

Next we adjust the coefficients of A so that they lie in the interval $(-q/2, q/2]$. The original message M is recovered by computing

$$M \equiv F_p * A \bmod p.$$

5.5 Lattice-Based Attack on M-NTRU

The key fact concerning attacks on M-NTRU is that the usual lattice attacks still apply. In *A General NTRU-Like Framework for Constructing Lattice-Based Public-Key Cryptosystems* [16], the authors provide a general framework for NTRU-like cryptosystems that describes a generic procedure for situating NTRU-like cryptosystems in the context of lattices. Summarizing for the case of M-NTRU, we consider the matrix

$$B = \begin{bmatrix} I_n & 0_n \\ H^T & q \cdot I_n \end{bmatrix}.$$

By the Gaussian Heuristic, it follows that solving the SVP for B when n is large will recover F and G with high likelihood. The presence of lattice-based attacks for M-NTRU will be the basis for the general attack we give for GR-NTRU.

5.6 Attack on GR-NTRU using M-NTRU

Previously, we described an attack of GR-NTRU which translates the structure to a $2 \cdot |G|$-dimensional lattice, and solves either the SVP or CVP for that lattice. This section describes a generalization of this lattice attack on GR-NTRU which utilizes M-NTRU. In essence, we will show how to reduce a GR-NTRU cryptosystem to a direct product of M-NTRU cryptosystem.

Let $\mathbb{G} = (G, p, q, f, g)$ be a GR-NTRU cryptosystem, with $h = f_q \cdot g \bmod q$ where $f_q \cdot f = 1 \bmod q$. Given an embedding

$$\tau : \mathbb{Z}[G] \hookrightarrow \mathbf{M}_{n_1}(\mathbb{Z}) \oplus \mathbf{M}_{n_2}(\mathbb{Z}) \oplus \cdots \oplus \mathbf{M}_{n_k}(\mathbb{Z}),$$

the cryptosystem $\mathbb{G}$ corresponds to k M-NTRU cryptosystems $\mathbb{M}_i$ over $\mathbf{M}_{n_i}(\mathbb{Z})$ as follows. For $1 \leq k \leq l$, define $F_i, H_i \in \mathbf{M}_{n_i}(\mathbb{Z})$ to be the images of f and h under the i^{th} projection map for τ:

$$F_i = \tau_i(f) \text{ and } H_i = \tau_i(h),$$

and let F_i and H_i be the secret and public key, respectively, for the M-NTRU cryptosystem $\mathbb{M}_i$. Since τ is injective, we may recover F and H from the collections $\{F_i\}$ and $\{H_i\}$.

Furthermore, let $m \in \mathcal{M}_p$ be a message and let c be the corresponding ciphertext for the $\mathbb{G}$ cryptosystem. For $1 \leq i \leq k$, let $M_i = \tau_i(m) \in \mathbf{M}_{n_i}(\mathbb{Z})$ and let $C_i = \tau_i(c) \in \mathbf{M}_{n_i}(\mathbb{Z})$, respectively. In this way, M_i can be interpreted as a message for the cryptosystem $\mathbb{M}_i$, and C_i can be interpreted as the corresponding ciphertext.

In summary, for each $1 \leq i \leq k$ we have a dictionary:

Ring	$\mathbb{Z}[G]$	$\tau(\mathbb{Z}[G]) \subset \bigoplus \mathbf{M}_{n_i}(\mathbb{Z})$
Private Key	f	$\{F_i\}_{i=1}^k$
Public Key	h	$\{H_i\}_{i=1}^k$
Message	m	$\{M_i\}_{i=1}^k$
Ciphertext	c	$\{C_i\}_{i=1}^k$

Table 5.2: Dictionary of cryptosystem objects between GR-NTRU and a product of M-NTRU systems.

Given a GR-NTRU cryptosystem $\mathbb{G}$, the strategy of attack works like this:

1. Find an injective ring homomorphism

$$\tau : \mathbb{Z}[G] \to \mathbf{M}_{n_1}(\mathbb{Z}) \bigoplus \mathbf{M}_{n_2}(\mathbb{Z}) \bigoplus \cdots \bigoplus \mathbf{M}_{n_k}(\mathbb{Z}).$$

2. Create the corresponding M-NTRU cryptosystems $\mathbb{M}_i$ for each $\leq i \leq k$. Use lattice attacks on each of these cryptosystems to obtain the private keys $\{F_i\}$.

3. Using the correspondence between the collection $\{\mathbb{M}_i\}$ and $\mathbb{G}$, obtain the private key f for $\mathbb{G}$ from the collection $\{F_i\}$.

The lattice attack for $\mathbb{G}$ takes $\mathcal{O}(N^2)$ computations, where $N = |G|$. However, the attack described above takes $\mathcal{O}(n_l^2)$ computations, where $n_l = \max\{n_1, \ldots, n_k\}$. So a homomorphism τ which minimizes n_l is the best choice for an attacker.

Chapter 6

GR-NTRU for Certain Groups

In the GR-NTRU paper, the security of GR-NTRU for four classes of groups are analyzed via the M-NTRU attack described above. We summarize the results for these groups and present an example of our own. First, we detail the process taken to analyze these groups.

The general approach taken to find an injective ring homomorphism τ for the M-NTRU attack relies primarily on group representation theory. Recall that if $\sigma : G \mapsto GL(n, K)$ is a group representation of G over the field K, then σ extends in a natural way to a homomorphism of group rings $\bar{\sigma} : K[G] \to \mathbf{M}_n(K)$. Vice-a-versa, a homormorphism of group rings may be restricted to give back a group representation.

Given a group G, suppose that we have a (not necessarily injective) homomorphism

$$\tau : \mathbb{Z}[G] \to \mathbf{M}_{n_1}(\mathbb{Z}) \bigoplus \mathbf{M}_{n_2}(\mathbb{Z}) \bigoplus \cdots \bigoplus \mathbf{M}_{n_k}(\mathbb{Z}). \tag{6.1}$$

By scalar extension with $\otimes_{\mathbb{Z}}\mathbb{C}$, we may extend τ to a map

$$\bar{\tau} : \mathbb{C}[G] \to \mathbf{M}_{n_1}(\mathbb{C}) \bigoplus \mathbf{M}_{n_2}(\mathbb{C}) \bigoplus \cdots \bigoplus \mathbf{M}_{n_k}(\mathbb{C}). \tag{6.2}$$

Via diagonalization, we may write this as a group ring homomorphism

$$\bar{\tau} : \mathbb{C}[G] \to \mathbf{M}_n(\mathbb{C})$$

where $n = \sum_{i=1}^{k} n_i$. By the correspondence given in proposition 2.4.2, this map provides a group representation $\sigma : G \to \mathbb{C}^n$. Thus, to find the desired map τ, it suffices to search among the representations of G. This would be a difficult task in general if not for the following additional result.

Proposition 6.0.1. *The homomorphism τ given in 6.1 is injective if and only if the group representation afforded by $\bar{\tau}$ in 6.2 includes all irreducible representations of G as subrepresentations.*

This fact is very important as it further restricts our attention to a very specific kind of representation, those that contain all irreducible representations as subrepresentations.

An attacker seeks to find a map τ such that the maximum of the n_i is as small as possible so that the computation time is minimized. Therefore, to find a suitable τ for an attack, one looks for an injective τ as in 6.1 where $\max\{n_i\}$ is minimized. By the previous string of ideas, we may obtain a τ by first finding all irreducible representations of the relevant group G and then building a representation out of all the irreducbible representations.

We end by summarizing the results of this approach as analyzed by the authors in the GR-NTRU paper, and afterwards we explore an example in detail for S_3.

Group	Order	$\max n_i$
$\bigoplus C_{n_i}$: Product of cyclic groups	$n_1 \times \cdots \times n_k$	$\phi(n_1) \times \ldots \times \phi(n_k)$
D_n: Dihedral group	$2n$	n
F_p: Frobenius group	$p(p-1)$	p
S_n: Symmetric group on n letters	$n!$	Cannot be estimated in general

Table 6.1: Summary of M-NTRU security analysis for various groups

The ϕ in the description of the product of cyclic groups is Euler's totient function. In the case of the symmetric group, the dimension of the lattice attack cannot be given explicitly in terms of n, so it must be investigated for a specific choice of n. In what follows, we look at how this works for S_3.

6.1 GR-NTRU for S_3

Let $G = S_3$ the symmetric group on 3 letters. To analyze the GR-NTRU cryptosystem for this group, we first describe key pieces of the representation theory for S_3 and use these to describe an injective τ for use by the M-NTRU attack.

To efficiently describe the desired decomposition of $\mathbb{Z}[S_3]$ into matrix rings, we rely on some of the powerful theorems of representation theory.

Theorem 6.1.1 (Maschke). *Given a finite group G, the group ring $\mathbb{Q}[G]$ is semi-simple.*

Theorem 6.1.2 (Wedderburn-Artin). *If A is a semi-simple algebra over a field K, then A is uniquely decomposed into a direct sum of matrix rings each over a skew field:*

$$A \simeq \bigoplus_{i=1}^{k} \mathbf{M}_{n_i}(D_i).$$

Thus we have a decomposition of $\mathbb{Q}[G]$ into matrix rings, however we can say even more in the case of $K = \mathbb{C}$.

Theorem 6.1.3.

$$\mathbb{C}[G] \simeq \bigoplus_{\sigma \in \hat{G}} \mathbf{M}_{n_\sigma}(\mathbb{C}).$$

Here, $\hat{G}$ is the set of isomorphism classes of irreducible representations of G, and n_σ is the degree of an irreducible representations σ.

In the case of S_n, passing from $\mathbb{C}$ to $\mathbb{Q}$ does not change the decomposition given above, so we have

$$\mathbb{Q}[G] \simeq \bigoplus_{\sigma \in \hat{G}} \mathbf{M}_{n_\sigma}(\mathbb{Q}).$$

Thus to obtain this decomposition for S_3 we must describe the irreducible representations of S_3. Recall proposition 2.4.1 which states that the number of distinct irreducible representations for a group G is equal to the number of conjugacy classes of G. In general, there is no canonical correspondence between these two objects. However, in the case of the symmetric group the theory of Young tableaux gives a very elegant mapping between the two in terms of partitions of n. We do not exposit the details here, but the reader may refer to Harris & Fulton chapter 4 [6] for how this works.

For each partition λ of the integer 3, we have a corresponding irreducible representation σ_λ. The way σ_λ is constructed is not unique, however the degree of the representation is invariant of the construction. The degree can be calculated using the hook formula for Young tableaux

$$\deg \sigma_\lambda = \frac{n!}{\Pi(\lambda)}$$

where $\Pi(\lambda)$ gives the product of hook lengths for the tableaux for λ. The partition (3) corresponds to a representation of size $\frac{3!}{6} = 1$, the partition $(2,1)$ corresponds to a representation of size $\frac{3!}{3} = 2$, and the partition $(1,1,1)$ corresponds to a representation of size $\frac{3!}{6} = 1$. Therefore,

$$\mathbb{Q}[S_3] \simeq \mathbf{M}_1(\mathbb{Q}) \oplus \mathbf{M}_1(\mathbb{Q}) \oplus \mathbf{M}_2(\mathbb{Q}).$$

So in an M-NTRU attack on an S_3-based group ring cryptosystem, the maximal dimension of a lattice problem an attacker must solve is 4. Without the decomposition the size of the lattice problem would be $2 \cdot |S_3| = 12$. For larger n, we can see greater reductions in dimension. What follows is a reproduction of a table of calculations from the GR-NTRU paper.

n	$\min \deg(\sigma_\lambda)$	$\min \deg(\sigma_\lambda)/n!$
8	90	$1/448$
9	216	$1/1680$
10	768	$1/4725$
11	2310	$1/17290$
⋮	⋮	⋮
19	64664600	$1/1881169920$
20	249420600	$1/975421440$
21	1118939184	$1/204838502400$

Table 6.2: List of M-NTRU attack dimensions for various S_n.

Chapter 7

Closing Remarks

We started by establishing the mathematics to understand the NTRU cryptosystem. The algebra of convolution polynomials enjoys a rich relationship to integer lattices, which lead to numerous ways one can analyze the security of NTRU. We saw that NTRU lends itself to a vast algebraic generalization as described by GR-NTRU. A closer look at GR-NTRU showed us that many algebraic structures we encounter in NTRU-like cryptosystems can be classified as group rings. We then described an improvement on the lattice-based attacks for GR-NTRU using M-NTRU and representation theory. Looking at a collection of examples, we saw explicitly how a matrix ring decomposition of the group ring $\mathbb{Z}[G]$ can reduce the computational complexity of lattice-based attacks.

The new attack on NTRU and its variants demonstrates that group rings and representation theory are an important perspective for further analyzing these cryptosystems. Interesting questions are left unanswered: which groups G provide practical cryptosystems through GR-NTRU, what more can GR-NTRU tell us about NTRU-like cryptosystems that already exist, and can representation theory be further leveraged for a deeper understanding of NTRU-like cryptosystems?

Bibliography

[1] PQC Standardization Process: Third Round Candidate Announcement.

[2] Khadijeh Bagheri, Mohammad-Reza Sadeghi, and Daniel Panario. A Non-commutative Cryptosystem Based on Quaternion Algebras. *arXiv:1709.02079 [cs, math]*, September 2017. arXiv: 1709.02079.

[3] Stephane Beauregard. Circuit for Shor's algorithm using 2n+3 qubits. *arXiv:quant-ph/0205095*, February 2003. arXiv: quant-ph/0205095.

[4] Michael Coglianese and Bok-Min Goi. MaTRU: A New NTRU-Based Cryptosystem. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan, editors, *Progress in Cryptology - INDOCRYPT 2005*, volume 3797, pages 232–243. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.

[5] Don Coppersmith and Adi Shamir. Lattice Attacks on NTRU. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Walter Fumy, editors, *Advances in Cryptology — EUROCRYPT '97*, volume 1233, pages 52–61. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. Series Title: Lecture Notes in Computer Science.

[6] William Fulton and Joe Harris. *Representation theory: a first course*. Number 129 in Graduate texts in mathematics ; Readings in mathematics. Springer, New York, corr. 3rd print edition, 1996.

[7] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Joe P. Buhler, edi-

tors, *Algorithmic Number Theory*, volume 1423, pages 267–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. Series Title: Lecture Notes in Computer Science.

[8] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An introduction to mathematical cryptography*. Undergraduate texts in mathematics. Springer, New York Heidelberg, 2. ed edition, 2014.

[9] Katherine Jarvis and Monica Nevins. ETRU: NTRU over the Eisenstein integers. *Designs, Codes and Cryptography*, 74(1):219–242, January 2015.

[10] Amir Hassani Karbasi and Reza Ebrahimi Atani. Iltru: An ntru-like public key cryptosystem over ideal lattices. *IACR Cryptol. ePrint Arch.*, 2015:549, 2015.

[11] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

[12] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

[13] Ehsan Malekian and Ali Zakerolhosseini. OTRU: A non-associative and high speed public key cryptosystem. In *2010 15th CSI International Symposium on Computer Architecture and Digital Systems*, pages 83–90, Tehran, Iran, September 2010. IEEE.

[14] Rakesh Nayak, C.V. Sastry, and Jayaram Pradhan. A matrix formulation for NTRU cryptosystem. In *2008 16th IEEE International Conference on Networks*, pages 1–5, New Delhi, India, 2008. IEEE.

[15] Gaithuru Juliet Nyokabi, Mazleena Salleh, and Ismail Mohamad. NTRU inverse polynomial algorithm based on circulant matrices using gauss-jordan elimination. In *2017 6th ICT International Student Project Conference (ICT-ISPC)*, pages 1–5, Johor, Malaysia, May 2017. IEEE.

[16] Yanbin Pan and Yingpu Deng. A General NTRU-Like Framework for Constructing Lattice-Based Public-Key Cryptosystems. In Souhwan Jung and Moti Yung, editors, *Information Security Applications*, volume 7115, pages 109–120. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[17] Donald S. Passman. *The algebraic structure of group rings*. Pure and applied mathematics. Wiley, New York, 1977.

[18] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *arXiv:quant-ph/0301141*, January 2004. arXiv: quant-ph/0301141.

[19] Hashim H. R., Molnár A., and Tengely Sz. Cryptanalysis of ITRU. *arXiv:2005.09258 [cs]*, May 2020. arXiv: 2005.09258.

[20] J. E. Roseblade. THE ALGEBRAIC STRUCTURE OF GROUP RINGS. *Bulletin of the London Mathematical Society*, 11(3):362–362, October 1979.

[21] Mehmet Sever and Ahmet Şükrü Özdemir. NTRU Over Galois Rings. *Applied Mathematics and Nonlinear Sciences*, 6(1):499–506, January 2021.

[22] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. arXiv: quant-ph/9508027.

[23] Joseph H Silverman. A meet-in-the-middle attack on an ntru private key.

[24] Sonika Singh and Sahadeo Padhye. Generalisations of NTRU cryptosystem: Generalisations of NTRU cryptosystem. *Security and Communication Networks*, 9(18):6315–6334, December 2016.

[25] Xavier Dahan Takanori Yasuda and Kouichi Sakurai. Characterizing NTRU-Variants Using Group Ring and Evaluating their Lattice Security.

[26] Nitin Vats. Algebraic Cryptanalysis of CTRU Cryptosystem. In Xiaodong Hu and Jie Wang, editors, *Computing and Combinatorics*, volume 5092, pages 235–244. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISSN: 0302-9743, 1611-3349 Series Title: Lecture Notes in Computer Science.

[27] Nitin Vats. NNRU, a noncommutative analogue of NTRU. *arXiv:0902.1891 [cs]*, February 2009. arXiv: 0902.1891.

[28] Hassan R. Yassein, Nadia M. G. Al-Saidi, and Alaa K. Farhan. A new NTRU cryptosystem outperforms three highly secured NTRU-analog systems through an innovational algebraic structure. *Journal of Discrete Mathematical Sciences and Cryptography*, pages 1–20, May 2020.

[29] Na Zhao and Shenghui Su. An Improvement and a New Design of Algorithms for Seeking the Inverse of an NTRU Polynomial. In *2011 Seventh International Conference on Computational Intelligence and Security*, pages 891–895, Sanya, Hainan, China, December 2011. IEEE.