UNIVERSITY
OF OSLO

Master's thesis

S-unit Attacks for Lattice-Based Cryptography

The Mathematical Theory

Ingeborg Nedkvitne

Mathematics
60 ECTS credits

Department of Mathematics
Faculty of Mathematics and Natural Sciences

Spring 2023

Ingeborg Nedkvitne

S-unit Attacks for Lattice-Based Cryptography

The Mathematical Theory

Supervisors:
Kristian Ranestad
Thomas Gregersen
Martin Strand

Abstract

Lattice-based cryptography derived from cylotomic rings, base their security on two hard mathematical problems, finding the shortest and the closest vector in a lattice. Even though there are many factors to consider when analyzing attacks against cryptosystems, this thesis focus on the mathematical theory. We study the properties of cyclotomic fields, S-units and how they relate to lattices. We present a detailed step by step approach on how to preform an S-unit attack and highlight the most important aspects to it. In addition, we develop new and comprehensive examples that demonstrates the different sides to such attacks.

Contents

Acknowledgements

I want to thank my supervisor, Kristian Ranestad, for introducing me to algebra and for the great guidance throughout this thesis. I also want to thank my supervisors, Thomas Gregersen and Matrin Strand, for introducing me to lattice-based cryptography and S-unit attacks, and for the additional help.

A special thank you to Leandros and Herman for the support during the two years. To "løpegruppa" and everyone else in the study halls, thank you for the great company.

Chapter 1

Introduction

Public key cryptography was first discovered in 1970's by James H. Ellis [5] and was an important development in cryptographic history. It consists of a secret and a public key, where the encryption is held secure by hard mathematical problems. Until recently these mathematical problems consisted of finding prime factorization of large numbers and the discrete log problem, including elliptic curves. Both of these problems are considered secure on classical computers, but with the threat of quantum computers becoming more powerful in the future, these problems will no longer be hard. It was already showed by Peter W. Shor in 1994 that such problems could easily be solved by quantum algorithms [13]. Therefore, in 2016 the National Institute of Standards and Technology (NIST) proposed to start a process for finding the next standardized public key cryptosystem that is both secure against quantum and classical computers.

Most of the entries to this process were lattice-based and in July of 2022, they announced that CRYSTALS-Kyber [11] was one of the most promising candidates for the new standardization in public key cryptography. CRYSTALS-Kyber is a lattice-based cryptosystem inspired by LWE and NTRU, both defined by cyclotomic rings. With lattice-based cryptography as the new standard it is natural to ask the following questions: are these schemes as secure as they seems? What if there is some not yet discovered hidden mathematical theory that could be exploited?

Lattice-based cryptography was first introduced by Miklós Ajtai in 1996, and the first lattice-based public key cryptosystem was introduced in 1998 by Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. There have been several improvements since then, which has produced many different variations of the original schemes, including CRYSTALS-Kyber. Lattice-based cryptography depends on either of two hard mathematical problems, the short vector problem and the closest vector problem. Today, there exists well known algorithms that attempt to solve these problems, but they depend on the dimension of the lattice, which makes them uncertain of how accurate their findings actually are. This means that there has not yet been found any attacks proving lattice-based cryptography insecure. Inspired by work of Daniel J. Bernstein and Tanja Lange [4], [2] we will look at the mathematical theory of lattice-based cryptography and explore a potential attack against them, namely *S-unit attacks.* This attack uses a very different approach than the best algorithms we have today. Instead of looking at the properties of the cryptosystem as a lattice, it rather attempts to exploits the properties of the cyclotomic ring itself. For the mathematical theory we primarily rely on Lawrence C. Washington's book *Introduction to Cyclotomic Fields* [15] and Gerald J. Janusz's book *Algebraic Number Fields* [6].

Our contribution is to present the mathematical theory behind cyclotomic fields and S-units. We provide a detailed step by step approach on how to preform an S-unit attack, declare mathematically the necessary choices done, along with simple examples for each concept. Most importantly, we apply the theory through new and comprehensive examples, where we implement all the techniques and display different outcomes of an S-unit attack.

1.1 Outline

In Chapter 2 we start by introducing lattices and some of their properties. We look at the basic concepts for lattice-based cryptography by describing the algorithms for NTRU and LWE. Then we present the hard mathematical problems for lattices and give a brief overview of why the existing algorithms for lattice attacks may not have the most accurate estimations to these hard problems. We end the chapter with the LLL-algorithm, including an example of a key recovery attack against NTRU.

The main mathematical theory is presented in Chapter 3. This is where we get an understanding of cyclotomic rings, highlight their most important properties for S-unit attacks and give some explanations for the choices we make. We introduce the concept of S-units and how they are related to lattices by using well-known concepts from algebraic number theory.

In the fourth chapter we describe the algorithms for attacks against lattice-based cryptography. We begin with outlining the steps for a unit attack. Then we construct an algorithm for S-unit attacks, where we also give some additional justification for the choices we made, such as how to choose the prime ideals for the S-units and which element we should attempt to reduce.

In Chapter 5 we present the main results through new and complex examples. These examples will display how the algorithms work for concrete values, both for general ideals and for specific attacks against NTRU. Most importantly, we construct new examples showcasing the different outcomes when preforming an S-unit attack.

In the sixth and final chapter we conclude on what we have found, look at the assumptions we have made for the S-unit attack and provide different areas that could be interesting to explored further or done differently.

Most of the computations are done using SageMath [14], which is a free open-source mathematics software written for calculations in algebra, number theory, calculus and statistics. See Appendix B for SageMath documentation.

1.2 Notation

We write polynomials with bold font, such as $\boldsymbol{a}(x) = a_0 + a_1x + a_2x^2 + \cdots a_{m-1}x^{m-1}$, to emphasize that they correspond to vectors. When we refer to the *size* of an element $v = (v_1, ..., v_n)$, it is the length of a vector with respect to the Euclidean norm on $\mathbb{R}^n$,

$$\|v\| = \sqrt{\sum_{i=1}^{n} |v_i|^2}.$$

The notion of a *good basis* for a lattice means a basis where the vectors are as orthogonal as possible to each other and with the smallest size as possible. The notation $\lfloor \cdot \rceil$ denotes the round of to closest integer.

We denote sets without zero by $\mathbb{F}^*$ and the residue class of integers co-prime to m modulo m by $(\mathbb{Z}/m\mathbb{Z})^*$. The sets on the form $\mathbb{Z}_p$ are $\mathbb{Z}$ modulo p. We use the letter K for the cyclotomic field and the letter R for the cyclotomic ring. For the the m'th root of unity we use x, ζ_m and $e^{2\pi i/m}$ interchangeably.

Lower case p is reserved as prime elements in $\mathbb{Z}$, capital P as prime ideals in the cyclotomic ring R and $\mathfrak{p}$ as prime ideals in a general ring. Unless otherwise, *norm* is referred to the algebraic norm $\mathcal{N}_{K/\mathbb{Q}}(a)$ for the field extension $K/\mathbb{Q}$. The *infinite norm* and *finite norm* are absolute values, where we use *finite norm* and *p-adic norm* interchangeably.

Chapter 2

Lattice-Based Cryptography

We begin by introducing lattices and some of their most important properties for lattice-based cryptography. Then, we give a brief overview of two classical cryptosystems, NTRU and LWE. We present the hard mathematical problems for lattices, including the bounds and heuristics used for solving these. To end the chapter, we describe the LLL algorithm, followed up with an example.

2.1 Lattice Theory

Definition 2.1. A *lattice* $\mathcal{L}$ is a finitely generated abelian subgroup of a real vector space $V \subseteq \mathbb{R}^n$ on the form

$$\mathcal{L} = \{a_1 v_1 + ... + a_d v_d \mid a_i \in \mathbb{Z}\}$$

where $v_1, ..., v_d \in \mathbb{R}^n$ are linearly independent vectors. If $\mathcal{L}$ has the same dimension as V $\mathcal{L}$ is a *full lattice* and the basis $v_1, ..., v_n$ of V is the basis of $\mathcal{L}$.

The basis vectors $v_1, ..., v_n$ can be written as the $n \times n$-matrix $M_{\mathcal{L}}$ with $v_1, ..., v_n$ as the columns. The span of a lattice $\mathcal{L}$ is the linear space spanned by its vectors,

$$\text{span}(\mathcal{L}) = \text{ span}(v_1, ..., v_n) = \{My \,|\, y \in \mathbb{R}^n\}.$$

Let $v_1, ..., v_n$ be a basis of $\mathcal{L}$ and let $w_1, ..., w_n \in \mathcal{L}$ be another set of vectors, then each w_i can be written as a linear combination of the basis vectors with integer coefficients a_{ij}, such as

$$\begin{gathered} w_1 = a_{11}v_1 + a_{12}v_2 + \ldots + a_{1n}v_n \\ w_2 = a_{21}v_1 + a_{22}v_2 + \ldots + a_{2n}v_n \\ \vdots \\ w_n = a_{n1}v_1 + a_{n2}v_2 + \ldots + a_{nn}v_n. \end{gathered}$$

The w_i's are also a basis for $\mathcal{L}$ if and only if the integer coefficients has a matrix

$$M_{\mathcal{L}} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$$

with determinant different from zero, i.e. $\det(M_{\mathcal{L}}) = \pm 1$. This follows from the fact that the coefficients are integers, hence $\det(M_{\mathcal{L}})$ must be an integer and we have that

$$1 = \det(I) = \det\Big(M_{\mathcal{L}} M_{\mathcal{L}}^{-1}\Big) = \det(M_{\mathcal{L}}) \det\Big(M_{\mathcal{L}}^{-1}\Big),$$

where both $\det(M_{\mathcal{L}})$ and $\det\left(M_{\mathcal{L}}^{-1}\right)$ are integers. Otherwise, if any of them were bigger than 1 the other one must be less than 1, hence not contain integer coefficients for a lattice.

Definition 2.2. For a full lattice $\mathcal{L}$ with the basis vector $\mathcal{B} = \{v_1, ..., v_n\}$, the set

$$\mathcal{F} = \{a_1 v_1 + ... + a_n v_n \,|\, 0 \leq a_i < 1, 1 \leq i \leq n\}$$

is the *fundamental domain* for $\mathcal{L}$.

Definition 2.3. Every vector $w \in \mathbb{R}^n$ can be written on the form $w = t + v$, called the *translates* of $\mathcal{F}$. The set

$$\mathcal{F} + v = \{w = t + v \mid t \in \mathcal{F}, v \in \mathcal{L}\}$$

is the union of all the translates which covers all of $\mathbb{R}^n$ as the v ranges over all the vectors in $\mathcal{L}$.

Definition 2.4. The *volume* of $\mathcal{F}$ is the determinant of $\mathcal{L}$ given by

$$\text{Vol}(\mathcal{F}) = |\det(\mathcal{L})|,$$

where the basis vectors $\mathcal{B} = \{v_1, ..., v_n\}$ of $\mathcal{L}$ have the coordinates $v_i = (a_{i1}, ..., a_{in})$ corresponding to the matrix

$$M_{\mathcal{L}} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

The volume of $\mathcal{F}$ has an upper bound called *Hadamard's inequality*

$$\det(\mathcal{L}) = \text{vol}(\mathcal{F}) \leq \|v_1\|\|v_2\| \dots \|v_n\|,$$

and the closer the basis is to being orthogonal the closer this inequality is an equality. We define the *Hadamard ratio* to be

$$\mathcal{H}(\mathcal{B}) = \left(\frac{\det(\mathcal{L})}{\|v_1\| \dots \|v_n\|}\right)^{1/n},$$

for the basis $\mathcal{B} = \{v_1, ..., v_n\}$ of $\mathcal{L}$. This has the ratio $0 < \mathcal{H}(\mathcal{B}) \leq 1$, and the closer it is to be 1, the more orthogonal are the basis vectors.

Now, we present some important results that are useful when estimating the size of a vector in a lattice, including Minkowski and Hermite's Theorem.

Definition 2.5. Define the *closed ball* with center $a \in \mathbb{R}^n$ and radius r to be

$$B_r(a) = \{x \in \mathbb{R}^n \mid \|x - a\| \leq r\}.$$

Theorem 2.6 (Janusz [6]). *An additive subgroup $\mathcal{L}$ of V is a lattice if an only if every ball B_r contains a finite number of points of $\mathcal{L}$.*

Definition 2.7. Let $S \subseteq \mathbb{R}^n$ be a subset.

- S is *bounded* if the lengths of the vectors in S are bounded.
- S is *symmetric* if for every point $x \in S$, then $-x \in S$.
- S is *convex* if whenever two points $x, y \in S$, then $\frac{x+y}{2} \in S$.
- S is a centrally symmetric, convex set if for every $x, y \in S$, we have that $\frac{x-y}{2} \in S$.

Theorem 2.8 (Minkowski [6]). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a full lattice and S a bounded, centrally symmetric, convex subset of $\mathbb{R}^n$. If $\mathrm{Vol}(S) > 2^n\ \mathrm{Vol}(\mathcal{L})$, then S contains a nonzero point of $\mathcal{L}$.*

Proof. Let $\mathcal{F}$ be the fundamental domain for a lattice $\mathcal{L}$. By definition we know that every vector $a \in S$ can be written as $a = v_a + w_a$ for $v_a \in \mathcal{L}$ and $w_a \in \mathcal{F}$, which means elements from $\mathcal{L}$ and $\mathcal{F}$ will cover all of S . Now, consider the set

$$\frac{1}{2}S = \left\{\frac{1}{2}a \mid a \in S\right\},$$

then

$$\mathrm{Vol}\left(\frac{1}{2}S\right) = 2^{-n}\mathrm{Vol}(S) > \det(\mathcal{L}) = \mathrm{Vol}(\mathcal{F}).$$

So, for $\mathcal{L}$ and $\mathcal{F}$ to cover all of S there must exists two distinct points $a_1, a_2 \in S$ such that for $\frac{1}{2}a_1 = v_{\frac{1}{2}a_1} + w_{\frac{1}{2}a_1}$ and $\frac{1}{2}a_2 = v_{\frac{1}{2}a_2} + w_{\frac{1}{2}a_2}$, we have that $v_{\frac{1}{2}a_1} \neq v_{\frac{1}{2}a_2}$, but $w_{\frac{1}{2}a_1} = w_{\frac{1}{2}a_2}$. This gives us the following

$$\begin{gathered}\tfrac{1}{2}a_1 = v_{\frac{1}{2}a_1} + w \text{ and } \tfrac{1}{2}a_2 = v_{\frac{1}{2}a_2} + w \text{ with } v_1, v_2 \in \mathcal{L} \text{ and } w \in \mathcal{F} \\ \Rightarrow \tfrac{1}{2}a_1 - \tfrac{1}{2}a_2 = v_{\frac{1}{2}a_1} - v_{\frac{1}{2}a_2} \in \mathcal{L}.\end{gathered}$$

Hence, we have a nonzero point $v_{\frac{1}{2}a_1} - v_{\frac{1}{2}a_2} \in \mathcal{L}$ and since S is symmetric and convex we also have that $v_{\frac{1}{2}a_1} - v_{\frac{1}{2}a_1} = \frac{a_1 - a_2}{2} \in S$. □

Theorem 2.9 (Hermite [8]). *Every lattice $\mathcal{L}$ of dimension n consists of a nonzero vector $v \in \mathcal{L}$ such that*

$$\|v\| \leq \sqrt{n} \cdot \det(\mathcal{L})^{1/n}.$$

Proof. Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a lattice and $S \subseteq \mathbb{R}^n$ the hypercube centered at 0 with sides of length $2s$, such that

$$S = \{(x_1, ..., x_n) \in \mathbb{R}^n \mid -s \leq x_i \leq s,\ \ \forall\ \ 1 \leq i \leq n\}.$$

The set S is closed, bounded and symmetric, with $\mathrm{Vol}(S) = (2s)^n$. If we set $s = \det(\mathcal{L})^{1/n}$, then $\mathrm{Vol}(S) = (2s)^n = 2^n \det(\mathcal{L})$ and by Minkowski's theorem there exists a nonzero vector $a = (a_1, ..., a_n) \in S \cap \mathcal{L}$ with norm,

$$\|a\| = \sqrt{a_1^2 + \ldots + a_n^2} \leq \sqrt{s^2 + \ldots + s^2} = \sqrt{n} \cdot s = \sqrt{n} \cdot \det(\mathcal{L})^{1/n}.$$ □

2.2 Lattice-Based Cryptosystems

To understand how to construct an attack against lattice-based cryptography, we need some knowledge about the actual cryptosystems. By introducing the well known cryptosystems, NTRU and LWE, we get an idea of the basic concepts of the algorithms within lattice-based cryptography and how to potentially break them. Today, these cryptosystems are know to be insecure, but most of the new post-quantum algorithms builds on the same theory and attain their security from the same hard mathematical problems.

2.2.1 NTRU

We start by looking at the basic concept for the cryptosystem NTRU. Consider the cyclotomic ring over the polynomial $x^m - 1$ (for more details see Chapter 3)

$$R = \mathbb{Z}[\zeta_m],$$

and the two finite polynomial rings

$$R_p = R/pR = \mathbb{Z}_p[\zeta_m] \quad \text{and} \quad R_q = R/qR = \mathbb{Z}_q[\zeta_m],$$

where p, q are two primes, ζ_m is the m'th root of unity and $\gcd(m,p) = \gcd(m,q) = 1$. The coefficients for an element in the ring R

$$\boldsymbol{a}(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_{m-1} x^{m-1} \in R$$

correspond to a vector on the form

$$(a_0, a_1, a_2, ..., a_{m-1}) \in \mathbb{Z}^m.$$

The coefficients in the rings R_p and R_q lies between $\{0, 1, ..., p-1\}$ and $\{0, 1, ..., q-1\}$ respectively. Further, we need the following definitions and results from Silverman [8].

Definition 2.10. For a polynomial $\bar{\boldsymbol{a}}(x) \in R_q$, the *center lift* of $\bar{\boldsymbol{a}}(x)$ to R is the unique polynomial $\mathbf{a}(x) \in R$ such that

$$\mathbf{a}(x) \pmod q = \bar{\boldsymbol{a}}(x)$$

where the coefficients are in the interval $(-q/2, q/2)$.

Definition 2.11. For any positive integers d_1 and d_2 we let

$$\mathcal{T}(d_1, d_2) = \left\{ \boldsymbol{a}(x) \in R \;\middle|\; \begin{array}{c} d_1 \text{ coefficients are equal to } 1 \\ d_2 \text{ coefficients are equal to } -1 \\ \text{all others equal to } 0 \end{array} \right\}.$$

Polynomials in $\mathcal{T}(d_1, d_2)$ are called *ternary polynomials.*

Proposition 2.12. *Let q be a prime. Then $\boldsymbol{a}(x) \in R_q$ has a multiplicative inverse if and only if*

$$\gcd(\boldsymbol{a}(x), x^m - 1) = 1 \quad \textit{in} \quad R_q.$$

If this is true, then the inverse $\boldsymbol{a}(x)^{-1} \in R_q$ can be computed using the extended Euclidean algorithm to find polynomials $\boldsymbol{f}(x), \boldsymbol{g}(x) \in R_q$ satisfying

$$\boldsymbol{a}(x)\boldsymbol{f}(x) + (x^m - 1)\boldsymbol{g}(x) = 1.$$

Then $\boldsymbol{a}(x)^{-1} = \boldsymbol{f}(x)$ in R_q.

Proof. Assume we have found two polynomials $\mathbf{f}(x), \mathbf{g}(x) \in R_q$ such that

$$\mathbf{a}(x)\mathbf{f}(x) + (x^m - 1)\mathbf{g}(x) = \gcd(\mathbf{a}(x), x^m - 1).$$

If $\gcd(\mathbf{a}(x), x^m - 1) = 1$ we get that $\mathbf{a}(x)\mathbf{f}(x) + (x^m - 1)\mathbf{g}(x) = 1$ and by reducing modulo $(x^m - 1)$ we get $\mathbf{a}(x)\mathbf{f}(x) = 1 \in R_q$. Hence, $\mathbf{f}(x)$ is the inverse of $\mathbf{a}(x)$. Conversely, if $\mathbf{a}(x) \in R_q$ has a multiplicative inverse in R_q, then $\mathbf{a}(x)$ is a unit and there exists a polynomial $\mathbf{f}(x) \in R_q$ such that $\mathbf{a}(x)\mathbf{f}(x) = 1 \in R_q$. This implies that

$$\mathbf{a}(x)\mathbf{f}(x) \equiv 1 \pmod{x^m - 1},$$

and so there is a polynomial $\mathbf{g}(x) \in R_q$ satisfying

$$\mathbf{a}(x)\mathbf{f}(x) = 1 + (x^m - 1)\mathbf{g}(x) \text{ in } R_q. \qquad \square$$

The following example shows how to find such multiplicative inverse to a polynomial. For later, we use Sage when constructing the keys for NTRU. See Appendix B.2 for the Sage code.

Example 2.13. Let $m = 16$ and $q = 3$, such that we have the cyclotomic rings $R = \mathbb{Z}[\zeta_{16}]$ and $R_3 = \mathbb{Z}_3[\zeta_{16}]$. Since the cyclotomic ring is over the polynomial $x^{16} - 1$, we have that $(-1)^2 = 1 = x^{16} = (x^8)^2$ and so $x^8 + 1 = 0$ is the minimal polynomial for R. Let $\mathbf{a}(x) = x^6 - x^5 - x^3 + 1$ be a polynomial in R_3, we want to find $\mathbf{a}(x)^{-1} \pmod 3$. First use the Euclidean algorithm to compute the greatest common divisor of $x^6 - x^5 - x^3 + 1$ and $x^8 + 1$ in $\mathbb{Z}_3$. Then, reverse the algorithm to find the inverse polynomial. The Euclidean algorithm gives us

$$\begin{aligned}
x^8 + 1 &= (x^6 - x^5 - x^3 + 1)(x^2 + x + 1) + (2x^5 + x^4 + x^3 - x^2 - x) \\
x^6 - x^5 - x^3 + 1 &= (2x^5 + x^4 + x^3 - x^2 - x)(2x) + (x^4 + x^3 - x^2 + 1) \\
2x^5 + x^4 + x^3 - x^2 - x &= (x^4 + x^3 - x^2 + 1)(2x - 1) + (x^3 - 2x^2 + 1) \\
x^4 + x^3 - x^2 + 1 &= (x^3 - 2x^2 + 1)(x) + (2x^2 - x - 2) \\
x^3 - 2x^2 + 1 &= (2x^2 - x - 2)(2x) + (x - 2) \\
2x^2 - x - 2 &= (x - 2)(2x) + 1.
\end{aligned}$$

Then $\gcd(x^8 + 1, x^6 - x^5 - x^3 + 1) = 1$, and the criteria for having an inverse is satisfied. Now, reverse the Euclidean algorithm, which yields the following inverse polynomial

$$\begin{aligned}
1 &= (2x^2 - x - 2) - (x - 2)(2x) \\
&= (2x^2 - x - 2) - (x^3 - 2x^2 + 1 - (2x^2 - x - 2)(2x))(2x) \\
&= (2x^2 - x - 2)(x^2 + 1) - (2x)(x^3 - 2x^2 + 1) \\
&= (x^4 + x^3 - x^2 + 1 - (x^3 - 2x^2 + 1)(x))(x^2 + 1) - (2x)(x^3 - 2x^2 + 1) \\
&= (x^2 + 1)(x^4 + x^3 - x^2 + 1) - (x^3 - 2x^2 + 1)(x^3) \\
&= (x^2 + 1)(x^4 + x^3 - x^2 + 1) - (2x^5 + x^4 + x^3 - x^2 - x - (x^4 + x^3 - x^2 + 1)(2x - 1))(x^3) \\
&= (x^4 + x^3 - x^2 + 1)(2x^4 - x^3 + x^2 + 1) - (x^3)(2x^5 + x^4 + x^3 - x^2 - x) \\
&= (x^6 - x^5 - x^3 + 1 - (2x^5 + x^4 + x^3 - x^2 - x)(2x))(2x^4 - x^3 + x^2 + 1) - (x^3)(2x^5 + x^4 + x^3 - x^2 - x) \\
&= (2x^4 - x^3 + x^2 + 1)(x^6 - x^5 - x^3 + 1) - (2x^5 + x^4 + x^3 - x^2 - x)(x^5 - 2x^4 + 2x) \\
&= (2x^4 - x^3 + x^2 + 1)(x^6 - x^5 - x^3 + 1) - (x^8 + 1 - (x^6 - x^5 - x^3 + 1)(x^2 + x + 1))(x^5 - 2x^4 + 2x) \\
&= (x^6 - x^5 - x^3 + 1)(x^7 - x^6 - x^5 + x^3 + 2x + 1) - (x^8 + 1)(x^5 - 2x^4 + 2x)
\end{aligned}$$

By taking the last equation modulo $(x^8 + 1)$ we have

$$1 = (x^6 - x^5 - x^3 + 1)(x^7 - x^6 - x^5 + x^3 + 2x + 1).$$

This means that $\mathbf{f}(x) = x^7 - x^6 - x^5 + x^3 + 2x + 1$ is the inverse of $\mathbf{a}(x) = x^6 - x^5 - x^3 + 1$ in R_3.

We can now describe the NTRU public key cryptosystem. Let Alice be the receiver and let Bob be the one sending Alice a message. Alice start by choosing some public parameters (n, p, q, d), where n is the degree of the minimal polynomial of the field extension, p and q are two distinct primes and d is the number of coefficients for the ternary polynomials. NTRU has three algorithms, key generator (KeyGen), encryption (Enc) and decryption (Dec). Alice starts by computing a key set using the following algorithm,

Algorithm 1 KeyGen

```
Input the public parameters $(n, p, q, d)$.
Choose a polynomial $\mathbf{f}(x) \in \mathcal{T}(d+1, d)$ that is invertible in $R_q$ and $R_p$.
Choose a polynomial $\mathbf{g}(x) \in \mathcal{T}(d, d)$.
Compute $\mathbf{f}_q(x) \equiv \mathbf{f}(x)^{-1} \pmod{q} \in R_q$.
Compute $\mathbf{f}_p(x) \equiv \mathbf{f}(x)^{-1} \pmod{p} \in R_p$.
Compute $\mathbf{h}(x) \equiv \mathbf{f}_q(x) \cdot \mathbf{g}(x) \in R_q$
Return the secret key $sk = (\mathbf{f}(x), \mathbf{g}(x))$ and the public key $pk = \mathbf{h}(x)$
```

For Bob to encrypt a message he uses,

Algorithm 2 Enc

```
Input a message $\mathbf{m}(x) \in R_p$.
Choose a random polynomial $\mathbf{r}(x) \in \mathcal{T}(d, d)$.
Compute $\mathbf{c}(x) \equiv p\mathbf{h}(x) \cdot \mathbf{r}(x) + \mathbf{m}(x) \pmod{q} \in R_q$.
Return $\mathbf{c}(x)$.
```

The last algorithm for Alice to decrypt the message is

Algorithm 3 Dec

```
Input an encrypted message $\mathbf{c}(x) \in R_q$.
Compute $\mathbf{a}(x) = \mathbf{f}(x) \cdot \mathbf{c}(x) = p\mathbf{f}(x) \cdot \mathbf{h}(x) \cdot \mathbf{r}(x) + \mathbf{f}(x) \cdot \mathbf{m}(x) = p\mathbf{f}(x) \cdot \mathbf{f}_q(x) \cdot \mathbf{g}(x) \cdot \mathbf{r}(x) + \mathbf{f}(x) \cdot \mathbf{m}(x) = p\mathbf{g}(x) \cdot \mathbf{r}(x) + \mathbf{f}(x) \cdot \mathbf{m}(x) \pmod{q}$.
Center lift $\mathbf{a}(x) \in R_q$ to $R$.
Compute $\mathbf{f}_p(x) \cdot \mathbf{a}(x) = p\mathbf{f}_p(x) \cdot \mathbf{g}(x) \cdot \mathbf{r}(x) + \mathbf{f}_p(x) \cdot \mathbf{f}(x) \cdot \mathbf{m}(x) \equiv \mathbf{m}(x) \pmod{q}$
Return the message $\mathbf{m}(x)$.
```

The decryption works since $\mathbf{f}_p(x), \mathbf{g}(x)$ and $\mathbf{r}(x)$ are all polynomials with small coefficients, hence zero or close to zero when computing modulo p. We have the following proposition which makes sure the polynomials will disappear.

Proposition 2.14. *If the public parameters (n, p, q, d) are chosen to satisfy*

$$q > (6d + 1)p,$$

then the decrypted polynomial $\boldsymbol{b}(x)$ is equal to the plaintext $\boldsymbol{m}(x)$.

To connect this cryptosystem to lattice theory we have that the public key

$$\mathbf{h}(x) = h_0 + h_1 x + h_2 x^2 + \cdots + h_{n-1} x^{n-1}$$

corresponds to the vector $\mathbf{h} = (h_0, h_1, ..., h_{n-1})$. This can be used to generate a lattice $\mathcal{L}_{\mathbf{h}}$, which gives the following $2n$-dimensional NTRU lattice spanned by the rows of the matrix,

$$M_{\mathbf{h}} = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n-1} \\ 0 & 1 & \cdots & 0 & h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right) = \begin{pmatrix} I & \mathbf{h} \\ 0 & qI \end{pmatrix},$$

and we have the following result from [8].

Proposition 2.15. *Assuming $\boldsymbol{f}(x) \cdot \boldsymbol{h}(x) \equiv \boldsymbol{g}(x) \pmod q$, let $\boldsymbol{u}(x) \in R$ be the polynomial satisfying*

$$\boldsymbol{f}(x) \cdot \boldsymbol{h}(x) = \boldsymbol{g}(x) + q\boldsymbol{u}(x).$$

Then

$$(\boldsymbol{f}(x), -\boldsymbol{u}(x))M_{\boldsymbol{h}} = (\boldsymbol{f}(x), \boldsymbol{g}(x)),$$

so the vector $(\boldsymbol{f}(x), \boldsymbol{g}(x))$ is in the NTRU lattice $\mathcal{L}_{\boldsymbol{h}}$.

2.2.2 LWE

The other well know lattice-based cryptosystem is *Learning with errors* (LWE). From the description by Peikert [12], this cryptosystem was originally based on integer vectors and classic linear algebra. There are now many improved versions, including a ring based (RLWE), which is the adaptation we will use. Let $m = 2n$, for $n \in \mathbb{Z}$, such that we have the cyclotomic ring

$$R = \mathbb{Z}[x]/(x^n + 1),$$

and for a positive integer q let

$$R_q = R/qR = \mathbb{Z}_q[x]/(x^n + 1).$$

RLWE is based on the *Ring Short Integer Solution* (ring-SIS) problem which is defined as follows,

Definition 2.16. Given m uniformly random elements $a_i \in R_q$ defining a vector $\mathbf{a} \in R_q^m$, the *ring short integer solution problem* is to find a nonzero vector $\mathbf{z} \in R^m$ of norm $\|\mathbf{z}\| \leq \beta$, such that

$$\langle \mathbf{a}, \mathbf{z} \rangle = \sum_i a_i \cdot z_i = 0 \in R_q.$$

Remark 2.17. There are some important assumption when choosing the parameters. We have that $\beta < q$, otherwise $\mathbf{z} = (q, 0, ..., 0) \in R^m$ would be a trivial solution. Next, β and m must be large enough such that a solution is guaranteed, i.e. $\beta \geq \sqrt{m_0}$ and $m \geq m_0$ where $m_0 \geq \log(q)$.

To define the RLWE cryptosystem, let χ be the Gaussian error distribution over R of width αq for an $\alpha < 1$.

Definition 2.18. Let $s \in R_q$ be an element defined as the *secret.* Let $A_{s,\chi}$ be the *ring LWE distribution* over $R_q \times R_q$ defined by

$$(a, b = a \cdot s + e \pmod q)$$

where $a \in R_q$ is chosen randomly form a uniform distribution and e is chosen from χ.

The RLWE cyptosystem has the secret $s \in R_q$ as the secret key and $(a, b) \in R_q \times R_q$ as the public key. To encrypt a message $m \in R_2$, corresponding to a bit string with n elements, we compute

$$c = (u, v) \approx \left(a \cdot r, b \cdot r + m \cdot \lfloor \frac{q}{2} \rceil \right) \in R_q \times R_q,$$

where $r \in R$ is chosen uniformly at random.

To decrypt the message we use the secret $s \in R_q$ and compute

$$\begin{aligned} v - s \cdot u &= b \cdot r + m \cdot \lfloor \frac{q}{2} \rceil - s \cdot a \cdot r \\ &= s \cdot a \cdot r + e \cdot r + m \cdot \lfloor \frac{q}{2} \rceil - s \cdot a \cdot r \\ &= e \cdot r + m \cdot \lfloor \frac{q}{2} \rceil \\ &\approx m \cdot \lfloor \frac{q}{2} \rceil \pmod q. \end{aligned}$$

We assume $e \cdot r \approx 0 \pmod q$ since it is a polynomial with small coefficients. Furthermore, we check each coefficient of $m \cdot \lfloor \frac{q}{2} \rceil$. If it is close to 0 we assume the bit is 0, if it is closer to $\lfloor \frac{q}{2} \rceil$ we assume it is 1.

In the same way as NTRU, RLWE can be interpreted as a lattice by embedding an ideal $I \subseteq R$ corresponding to a cyclic lattice in $\mathbb{Z}^n$. Let $a \in R_q$ be a polynomial generating an ideal $I \subset R$. Then by letting the coefficients of a correspond to a vector in $\mathbb{Z}^n$, and to the first column of the circular matrix $\mathbf{A}_a \in \mathbb{Z}_q^{n \times n}$, we have a lattice $\mathcal{L}_a \in \mathbb{Z}^n$ generated by the columns of $\mathbf{A}_a$. We draw m random elements $a_i \in R_q$, defining a vector $\mathbf{a} \in \mathbb{Z}^m$, and construct the matrix $\mathbf{A}_{a_i} \in \mathbb{Z}_q^{n \times n}$ for each a_i. Define the matrix $\mathbf{A} = [\mathbf{A}_{a_1} \mid \cdots \mid \mathbf{A}_{a_m}] \in \mathbb{Z}_q^{n \times nm}$ and the vector $b^t = s^t \mathbf{A} + e^t \pmod q$. Now, let $b^t = [b_1^t \mid \cdots \mid b_m^t]$ for each $b_i \in \mathbb{Z}_q^n$. Then from the relation $b_i^t = s^t \mathbf{A}_{a_i} + e_i^t$, we get

$$\begin{aligned} \bar{b}_i^t &= b_i^t \mathbf{A}_{a_i}^{-1} \\ &= s^t + e_i^t \mathbf{A}_{a_i}^{-1} \end{aligned}$$

for each $i = 1, ..., m$. We use this to find an error e_i by combining different relations on the form $\bar{b}_i^t - \bar{b}_j^t = e_i^t \mathbf{A}_{a_i}^{-1} - e_j^t \mathbf{A}_{a_i}^{-1}$ for $i \neq j$. Hence, finding one error e_i yields the secret s by computing

$$(b_i^t - e_i^t) \cdot \mathbf{A}_{a_i}^{-1} = s^t \mathbf{A}_{a_i} \cdot \mathbf{A}_{a_i}^{-1} + e_i^t \mathbf{A}_{a_i}^{-1} - e_i^t \mathbf{A}_{a_i}^{-1} = s^t.$$

2.3 The Hard Mathematical Problems

In cryptography a *Hard mathematical problem* is defined as a mathematical problem that cannot be solved in polynomial time, i.e. there is no algorithm that solves the problem with a running time upper bounded by a polynomial expression. In lattice-based cryptography there are two such problems to analyze. Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice and $v = (v_1, ..., v_n)$ an element in $\mathcal{L}$. We define $\lambda_1(\mathcal{L}) = \min_{w \in \mathcal{L}} \|w\|$ to be the smallest vector in the lattice. Then we have the following two problems:

- **The Shortest Vector Problem** (SVP): Find a nonzero vector $v \in \mathcal{L}$ that satisfies $\|v\| = \lambda_1(\mathcal{L})$.
- **The Closest Vector Problem** (CVP): For a given vector $w \in \mathbb{R}^n$, find a vector $v \in \mathcal{L}$ that minimizes $||w - v||$.

Since these are hard mathematical problems and we are working in high dimensional lattices, it is most likely not possible to find the exact values as described above. Therefore, we often rewrite these problems with an approximation factor. Let γ be the approximation factor depending on the dimension n of the lattice. Then we have the following problems.

- **The Approximate Shortest Vector Problem** (apprSVP$_\gamma$): Find a nonzero vector $v \in \mathcal{L}$ satisfying $\|v\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.
- **The Approximate Closest Vector Problem** (apprCVP$_\gamma$): For a given vector $w \in \mathbb{R}^n$, find a nonzero vector $v \in \mathcal{L}$ such that $\|v - w\| \leq \gamma \cdot \text{dist}(w, \mathcal{L})$.

For NTRU the hard mathematical problem is to recover the secret key, by only knowing the public parameters (n, p, q, d) and the public key $\mathbf{h}(x)$. From the hidden relationship

$$\mathbf{f}(x) \cdot \mathbf{h}(x) \equiv \mathbf{g}(x) \pmod{q},$$

where $\mathbf{f}(x)$ and $\mathbf{g}(x)$ have very small coefficients, it is possible to find a small vector in the lattice $\mathcal{L}_{\mathbf{h}}$ corresponding to the vector $(\mathbf{f}(x), \mathbf{g}(x))$.

Also for RLWE, the hard mathematical problems is to recover the secret $s \in R_q$. As mentioned at the end of Section 2.2.2, elements of R_q corresponds to a circular lattice $\mathcal{L}_a$ in $\mathbb{Z}^n$. From the public key $(a, b = a \cdot s + e \pmod{q})$, we use the polynomial a to construct the matrix $\mathbf{A}_a \in \mathbb{Z}_q^{n \times n}$ generating $\mathcal{L}_a$, and the polynomial b to obtain the relation $b^t = s^t\mathbf{A} + e^t \pmod{q}$ as a vector. Then, finding the secret $s \in R_q$ corresponds to solving SVP and CVP for $\mathcal{L}_a$.

2.3.1 Bounds and Heuristics

We will now look at how we can estimate the length of short elements in a lattice.

Definition 2.19. For a given lattice $\mathcal{L}$ of dimension n the *Hermite's constant* is defined as

$$\gamma_n = \sup_{\mathcal{L}} \frac{\lambda_1(\mathcal{L})^2}{\det(\mathcal{L})^{2/n}}.$$

In other words, the estimate for the smallest nonzero vector $v \in \mathcal{L}$ is

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

To improve Hermite's constant we apply Theorem 2.8, but first an estimation on the volume of a closed ball.

Definition 2.20. The *Gamma function* $\Gamma(s)$ for $s > 0$ is the integral

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

Theorem 2.21. *The volume of $B_r(a)$ is*

$$\mathrm{Vol}(B_r(a)) = \frac{\pi^{n/2}r^n}{\Gamma(1+n/2)}.$$

For large values of n we have the following approximation of the volume,

$$\mathrm{Vol}(B_r(a))^{1/n} \approx \sqrt{\frac{2\pi e}{n}} \cdot r.$$

Proof. Proving the first part of the theorem is done by using basic integration techniques with polar coordinates. The second part follows from applying the Stirling formula for the gamma function, which gives us

$$\mathrm{Vol}(B_r(a))^{1/n} = \frac{\pi^{n/2}r^n}{\Gamma(1+n/2)} \approx \frac{\pi^{n/2}r^n}{(n/2e)^{1/2}} = \sqrt{\frac{2\pi e}{n}} \cdot r \qquad \square$$

Now, let $B_r = B_r(0)$ be a ball at center 0 with radius r. By Theorem 2.8 we want a radius to satisfy,

$$\mathrm{Vol}(B_r) \geq 2^n \mathrm{det}(\mathcal{L}),$$

such that B_r contains a nonzero lattice point. Using the approximation of the volume above we get a ball with radius

$$r \gtrapprox \sqrt{\frac{2n}{\pi e}} \cdot \mathrm{det}(\mathcal{L})^{1/n}.$$

This means there exists a nonzero vector $v \in \mathcal{L}$ satisfying,

$$\|v\| \lessapprox \sqrt{\frac{2n}{\pi e}} \cdot \mathrm{det}(\mathcal{L})^{1/n}.$$

Lastly, we need the approximation, $|B_r \cap \mathcal{L}| \approx \mathrm{Vol}(B_r)/\mathrm{Vol}(\mathcal{L})$, of how many copies of $\mathcal{F}$ that fits into the ball B_r. This is not very useful when n is large and the radius r small, so assume the ratio $\mathrm{Vol}(B_r)/\mathrm{Vol}(\mathcal{F})$ is close to 1. Then for a large value of n we have

$$\left(\frac{2\pi e}{n}\right)^{n/2} \cdot r^n \approx \mathrm{Vol}(B_r) = \mathrm{Vol}(\mathcal{F}) = \mathrm{det}(\mathcal{L})$$

$$\Rightarrow r \approx \sqrt{\frac{2n}{\pi e}} \cdot \mathrm{det}(\mathcal{L})^{1/n}.$$

We have come to the following definition for the smallest vector in a lattice.

Definition 2.22. Let $\mathcal{L}$ be a lattice of dimension n. The *Gaussian expected shortest length* is

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

The *Gaussian heuristics* says that a length of a shortest nonzero vector in a random lattice $\mathcal{L}$ will satisfy

$$\lambda_1(\mathcal{L}) \approx \sigma(\mathcal{L}).$$

More precisely, for a fixed $\epsilon > 0$, a randomly chosen lattice $\mathcal{L}$ will satisfy,

$$(1-\epsilon)\sigma(\mathcal{L}) \leq \lambda_1(\mathcal{L}) \leq (1+\epsilon)\sigma(\mathcal{L}).$$

This estimate increases with the dimension of the lattice, and the bounds for the estimated size of the smallest element will become too large. Therefore, it provides an ambiguous result when attempting to find a small vector.

2.4 The LLL Reduction Algorithm

We describe the well known *Lenstra–Lenstra–Lovász (LLL) algorithm* for lattice reduction to present the main idea of how to solve the hard mathematical problems. The LLL algorithm solves both apprSVP$_\gamma$ and apprCVP$_\gamma$ within a factor depending on the dimension of the lattice. The algorithm does very well for small dimension lattices, but as shown in Section 2.3.1 the approximation factor have a tendency to become too large and imprecise for high dimensions.

To describe the LLL algorithm we first introduce the *Gaussian lattice reduction* algorithm, a reduction algorithm for 2-dimensional lattices. The LLL algorithm builds on the same concept, but for higher dimension lattices. The idea of the Gaussian algorithm is to find a good basis for a given lattice, i.e. to find basis vectors that are as short and as orthogonal as possible to each other. Since we are working in a lattice, the coordinates of the basis vectors are integers, and fully orthogonal basis vectors may not exist.

Assume $\mathcal{L} \subset \mathbb{R}^2$ is a 2-dimensional lattice with the basis vectors v_1 and v_2. If $\|v_1\| > \|v_2\|$, we swap the vectors. Otherwise, we make v_2 smaller by subtracting a multiple of v_1. From linear algebra

$$v_2^* = v_2 - \frac{v_1 \cdot v_2}{\|v_1\|^2} \cdot v_1$$

is the projection of v_2 onto the orthogonal complement of v_1. Since $\frac{v_1 \cdot v_2}{\|v_1\|^2}$ is most likely not an integer, v_2^* may not be in the lattice. Therefore, we subtract with the closest round off to an integer instead,

$$v_2^* = v_2 - m \cdot v_1 \quad \text{where} \quad m = \left\lfloor \tfrac{v_1 \cdot v_2}{\|v_1\|^2} \right\rceil$$

and set $v_2 = v_2^*$. If $\|v_1\| < \|v_2\|$, we are done and we return the reduced basis vectors. If $\|v_1\| > \|v_2\|$, we swap v_1 and v_2, and do the same processes until $\|v_1\| < \|v_2\|$. This can be summed up with the following algorithm from Silverman [8].

Algorithm 4 Gaussian Lattice Reduction Algorithm

```
Input: Lattice $\mathcal{L}$ with basis $v_1, v_2$.
Output: A reduced basis $v_1$ and $v_2$.
If $\|v_1\| > \|v_2\|$ swap $v_1$ and $v_2$
Compute $m = \lfloor \frac{v_a \cdot v_2}{\|v_1\|^2} \rceil$
if $m = 0$ then
    return $v_1$ and $v_2$.
else
    Replace $v_2$ with $v_2 - mv_1$.
end if
Repeat from step 1.
```

Now, assume $\mathcal{L} \subset \mathbb{R}^n$ is a lattice with a basis $\mathcal{B} = \{v_1, ..., v_n\}$. To find a good basis for $\mathcal{L}$ we need the following inequality,

$$\det(\mathcal{L}) = \mathrm{Vol}(\mathcal{F}) \leq \|v_1\| \cdots \|v_n\|,$$

where $\mathrm{Vol}(\mathcal{F})$ is the volume of the fundamental domain of $\mathcal{L}$. When the vectors are orthogonal this inequality becomes an equality. From the basis $\mathcal{B}$, we compute a set with Gram-Schmidt orthogonal basis vectors $\mathcal{B}^* = \{v_1^*, ..., v_n^*\}$, by the following algorithm

Algorithm 5 Gram-Schmidt Algorithm

```
Input: A basis $v_1, ..., v_n$ for a vector space $v \subset \mathbb{R}^n$.
Output: Orthogonal basis $v_1^*, ..., v_n^*$ for $V$.
Set: $v_1^* = v_1$
for $i = 2, 3, ..., n$ do
    Compute: $\mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2}$ for $1 \leq j \leq i-1$
    Set: $v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} \cdot v_j^*$
end for
```

The vectors in $\mathcal{B}^*$ are most likely not in $\mathcal{L}$, and therefore not a basis for the lattice, but

$$\det(\mathcal{L}) = \prod_{i=1}^{n} \|v_i^*\|.$$

This follows from Section 2.1. Let $M_{\mathcal{B}}$ be the matrix for the basis vectors in $\mathcal{B}$ and $M_{\mathcal{B}^*}$ be the matrix for $\mathcal{B}^*$. Let A be the change of basis matrix such that we have $AM_{\mathcal{B}^*} = M_{\mathcal{B}}$. Then,

$$\det(\mathcal{L}) = |\det(M_{\mathcal{B}})| = |\det(AM_{\mathcal{B}^*})| = |\det(A)\det(M_{\mathcal{B}^*})| = |\det(M_{\mathcal{B}^*})| = \prod_{i=1}^{n} \|v_i^*\|,$$

since $\det(A) = \pm 1$. Thus, the orthogonal basis $\mathcal{B}^*$ defines when a lattice is LLL-reduced.

Definition 2.23. Let $\mathcal{B} = \{v_1, ..., v_n\}$ be a basis for a lattice $\mathcal{L}$ and let $\mathcal{B}^* = \{v_1^*, ..., v_n^*\}$ be the associated Gram-Schmidt orthogonal basis. The basis $\mathcal{B}$ is said to be *LLL-reduced* if it satisfies the following conditions,

Size condition:	$\mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \leq \frac{1}{2}$	for all	$1 \leq j < i \leq n.$
Lovász condition:	$\|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{j,i-1}^2\right) \|v_{i-1}^*\|^2$	for all	$1 < i \leq n.$

Theorem 2.24 (Silverman [8]). *Let $\mathcal{L}$ be a lattice of dimension n. Any LLL-reduced basis $\mathcal{B}$ for $\mathcal{L}$ has the following properties,*

$$\textstyle\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} \det(\mathcal{L}) \quad \textit{and} \quad \|v_j\| \leq 2^{(i-1)/2} \|v_i^*\|$$

for all $1 \leq j < i \leq n$. Further, the initial vector in a LLL-reduced basis satisfies

$$\|v_1\| \leq 2^{(n-1)/4} |\det(\mathcal{L})|^{1/n} \quad \textit{and} \quad \|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in \mathcal{L}} \|v\|.$$

Thus an LLL-reduced basis solves apprSVP *within a factor of* $2^{(n-1)/2}$.

The LLL reduction algorithm can be summarized with the following algorithm from Silverman [8].

Algorithm 6 LLL Lattice Reduction Algorithm

```
Input: A basis $v_1, ..., v_n$ for lattice $\mathcal{L}$.
Output: LLL-reduced basis $v_1, ..., v_n$.
Set: $k = 2$
Set: $v_1^* = v_1$
while $k \leq n$ do
    for $j = k - 1, ..., 1$ do
        Set : $v_k = v_k - \lfloor \mu_{k,j} \rceil v_j$
    end for
    if $\|v_k^*\|^2 \geq (\frac{3}{4} - \mu_{k,k-1}^2)\|v_{k-1}^*\|^2$ then
        Set: $k = k + 1$
    else
        Swap: $v_{k-1}$ and $v_k$
        Set: $k = \max\{k - 1, 2\}$
    end if
end while
```

Remark 2.25. Every time we find a new representative for v_k we need to update the values of the Gram-Schmidt vectors $v_1^*, ..., v_n^*$ for the rest of the calculations.

The LLL algorithm solves apprSVP$_\gamma$, but if we combine it with *Babai's algorithm* we can also solve apprCVP$_\gamma$.

Theorem 2.26 (Silverman [8]). *There is a constant C such that for any lattice $\mathcal{L}$ of dimension n given by a basis $v_1, ..., v_n$ the following solves* apprCVP *within a factor of C^n.*

1. *Use LLL to reduce the basis $v_1, ..., v_n$.*
2. *Use Babai's algorithm on the LLL-reduced basis.*

Where Babai's algorithm is as follows,

Algorithm 7 Babais Round-off Algorithm

```
    Input: Lattice $\mathcal{L}$ with basis $v_1, ..., v_r$ and a vector $y \in \mathbb{R}^r$.
    Output: A vector $y' \in \mathcal{L}$ close to $y$.
1: Find a vector $t = (t_1, ..., t_r) \in \mathbb{R}^r$ by solving $t \cdot M_U = y$.
2: Round of each coordinate by setting $t = (\lfloor t_1 \rceil, ..., \lfloor t_r \rceil)$.
3: Write $y' = t_1 v_1 + ... + t_r v_r$.
```

We end this section with an example where we solve both SVP and CVP for a given lattice.

Example 2.27. Let $\mathcal{L} \subset \mathbb{R}^4$ be a lattice of dimension 4. Let $\mathcal{B} = \{v_1, v_2, v_3, v_4\}$ be a basis with the following coordinates

$$\begin{aligned} v_1 &= (10, 194, -118, 22) \\ v_2 &= (66, 163, -122, 15) \\ v_3 &= (-28, 63, 155, 65) \\ v_4 &= (-158, -39, -149, 146). \end{aligned}$$

We start by solving SVP. The smallest vector is v_3, with the size $\|v_3\| = 181.667$. For the Hadamard ratio, we have $\det(\mathcal{L}) = 431738302$ and $\prod_{i=1}^{4} \|v_i\| = 2354905381$, which gives

$$\mathcal{H}(\mathcal{B}) = \left(\frac{\det(\mathcal{L})}{\|v_1\| \dots \|v_4\|}\right)^{1/n} = \left(\frac{431738302}{2354905381}\right)^{1/4} = 0.65.$$

Algorithm 6 reduces the basis vectors to

$$\begin{aligned} v_1 &= (56, -31, -4, -7) \\ v_2 &= (28, 32, 151, 58) \\ v_3 &= (66, 163, -122, 15) \\ v_4 &= (-18, -69, -6, 190). \end{aligned}$$

For detailed step by step calculations see Appendix A.1. Now, the smallest vector is v_1, with the size $\|v_1\| = 64.513$. The product of the vectors are $\prod_{i=1}^{4} \|v_i\| = 470023074$ and the Hadamard ratio is

$$\mathcal{H}(\mathcal{B}) = \left(\frac{431738302}{470023074}\right)^{1/4} = 0.98.$$

Hence, the new basis vectors are almost orthogonal to each other and the smallest vector is much smaller.

To solve CVP, assume we have the vector

$$y = (52.43, -32.51, -2.39, 132.48) \in \mathbb{R}^4,$$

which is not in the lattice. We want to find the closest vector to y in $\mathcal{L}$. First we apply Algorithm 7 with the original basis, and then with the reduced basis. Let M_1 denote the matrix with the original basis and M_2 denote the matrix with the reduced basis. From Algorithm 7 the equation $t_1 \cdot M_1 = y$ gives the vector

$$t = (-2.97, 3.17, 0.84, 0.65) \approx (-3, 3, 1, 1)$$

corresponding to the closest vector

$$y_1 = -3 \cdot v_1 + 3 \cdot v_2 + v_3 + v_4 = (-18, -69, -6, 190) \in \mathcal{L}.$$

For the new and reduced basis, the equation $t_2 \cdot M_2 = y$ gives the vector

$$t = (0.82, 0.19, 0.20, 0.65) \approx (1, 0, 0, 1)$$

corresponding to the closest vector

$$y_2 = v_1 + v_4 = (38, -100, -10, 183) \in \mathcal{L}.$$

By comparing the size difference for both of these vectors

$$\|y - y_1\| = 98.05 > 85.87 = \|y - y_2\|,$$

we observe that the reduced basis gives us a vector that is closer to the original vector y.

2.4.1 Applying LLL to NTRU

We end the chapter by applying the LLL algorithm to find the secret key for the NTRU cryptosystem. This algorithm solves SVP for the NTRU lattice. Recall from Section 2.2.1 that we have the public parameters (n, p, q, d) and the cyclotomic rings on the form

$$R = \mathbb{Z}[x]/(x^n + 1), \quad R_p = \mathbb{Z}_p[x]/(x^n + 1) \quad \text{and} \quad R_q = \mathbb{Z}_q[x]/(x^n + 1).$$

The secret key is on the form $sk = (f(x), g(x))$, for two ternary polynomials $f(x), g(x) \in R$. The public key is the polynomial $pk = h(x) \in R_q$, with the relation $h(x) = f_q(x) \cdot g(x) \pmod{q}$.

For this example, let the public parameters be $(n, p, q, d) = (8, 3, 41, 2)$ and let the secret key consist of the polynomials,

$$f(x) = x^6 - x^4 + x^3 + x^2 - 1 \quad \text{and} \quad g(x) = x^6 + x^4 - x^2 - x.$$

We use Sage to compute the public key

$$h(x) = 34x^7 + 31x^6 + 5x^5 + 5x^4 - 21x^3 - 19x^2 - 38x - 12.$$

See Appendix B.2 for the calculations done by Sage. Form the public parameters and the public key we construct the NTRU lattice which is generated by the rows of the $2n \times 2n = 16 \times 16$-dimensional matrix,

$$M_h = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -38 & -19 & -21 & 5 & 5 & 31 & 34 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -34 & -12 & -38 & -19 & -21 & 5 & 5 & 31 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -31 & -34 & -12 & -38 & -19 & -21 & 5 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -5 & -31 & -34 & -12 & -38 & -19 & -21 & 5 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -5 & -5 & -31 & -34 & -12 & -38 & -19 & -21 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 21 & -5 & -5 & -31 & -34 & -12 & -38 & -19 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 19 & 21 & -5 & -5 & -31 & -34 & -12 & -38 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 38 & 19 & 21 & -5 & -5 & -31 & -34 & -12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41
\end{pmatrix}$$

We recover the secret key by finding the polynomials $f(x)$ and $g(x)$ in the reduced matrix of M_h. We reduce M_h by using the LLL algorithm. We use Sage to reduce M_h, see Appendix B.3, and we get the following reduced matrix,

$$M_h^{red} = \begin{pmatrix}
-1 & 0 & 1 & 0 & 1 & 0 & -1 & -1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & -1 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 1 & -1 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 1 & 0 \\
-1 & 0 & -1 & 0 & 1 & 1 & -1 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 1 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & 1 & 0 & -1 & -1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & -1 \\
-3 & -4 & -2 & 0 & -6 & 2 & -5 & 3 & -3 & -3 & 5 & -2 & 5 & 1 & -4 & 6 \\
2 & 0 & 6 & -2 & 5 & -3 & -3 & -4 & -5 & 2 & -5 & -1 & 4 & -6 & -3 & -3 \\
4 & 2 & 0 & 6 & -2 & 5 & -3 & -3 & 3 & -5 & 2 & -5 & -1 & 4 & -6 & -3 \\
-3 & -3 & -4 & -2 & 0 & -6 & 2 & -5 & -6 & -3 & -3 & 5 & -2 & 5 & 1 & -4 \\
0 & 6 & -2 & 5 & -3 & -3 & -4 & -2 & 2 & -5 & -1 & 4 & -6 & -3 & -3 & 5 \\
2 & -5 & 3 & 3 & 4 & 2 & 0 & 6 & 1 & -4 & 6 & 3 & 3 & -5 & 2 & -5 \\
-6 & 2 & 4 & 4 & 1 & 0 & 5 & -2 & -3 & 6 & 2 & 3 & -6 & 2 & -5 & -2 \\
5 & -2 & 6 & -2 & -4 & -4 & -1 & 0 & -5 & -2 & 3 & -6 & -2 & -3 & 6 & -2
\end{pmatrix}$$

To recover the polynomials $f(x)$ and $g(x)$ from M_h^{red}, we find the smallest vector and divide it in the middle into two vectors. If the two vectors corresponds to two nonzero polynomials with small coefficients, we are satisfied. In this case, the first eight rows all have length $\sqrt{9}$. We choose the first row corresponding to the vector

$$v_1 = (-1, 0, 1, 0, 1, 0, -1, -1, 1, 0, 1, 0, 0, 1, 1, 0)$$

and divide it into the two vectors

$$w_1 = (-1, 0, 1, 0, 1, 0, -1, -1) \quad \text{and} \quad w_2 = (1, 0, 1, 0, 0, 1, 1, 0).$$

Then each entry will be the coefficient for the two polynomials representing the secret key. The leftmost entry represents the constant term and the rightmost entry represents the coefficient for x^7. From the public parameter $d = 2$, we know that $f(x)$ should have 5 coefficients and $g(x)$ should have 4, so let w_1 be a representative for $f(x)$ and w_2 for $g(x)$. The secret key is then,

$$\tilde{f}(x) = -1 + x^2 + x^4 - x^6 - x^7 \quad \text{and} \quad \tilde{g}(x) = 1 + x^2 + x^5 + x^6.$$

If we multiply these polynomials with x^4, we recover the exact same secret key as we had in the beginning,

$$f(x) = x^4 \cdot \tilde{f}(x) = x^6 - x^4 + x^3 + x^2 - 1 \quad \text{and} \quad g(x) = x^4 \cdot \tilde{g}(x) = x^6 + x^4 - x^2 - x.$$

Hence, it is the same polynomials up to units, and $sk = (\tilde{f}(x), \tilde{g}(x))$ works just as fine for decryption as the original secret key $sk = (f(x), g(x))$.

In this example the LLL algorithm worked perfectly, in fact we recovered precisely the key set that we wanted.

Chapter 3

Cyclotomic Rings

The mathematical theory needed for S-unit attacks lies within cyclotomic fields and p-adic numbers. With this chapter we present the most important mathematical theory that can be used against lattice-based cryptography. We introduce the concept of S-units and demonstrate how to find them for a given cyclotomic ring.

A cyclotomic ring is the ring of integers in the cyclotomic field of the field extension $K/\mathbb{Q}$. For the most part we will be looking at cyclotomic rings on the form $R = \mathbb{Z}[x]/(x^n + 1)$. S-units are elements in the cyclotomic ring on the form

$$a = \prod_i a_i^{n_i},$$

where a_i are generators for the S-unit group, consisting of cyclotomic units and generators of prime ideals in R.

3.1 Cyclotomic Fields

3.1.1 Basics

Lattice-based cryptography are often based on polynomial rings, where the ring comes form various field extensions. In our case it comes form cyclotomic fields, so it is necessary to understand the mathematical theory of such fields when considering a attack against them.

Definition 3.1. The *mth roots of unity* are all the roots $\zeta_m = e^{2\pi i/m}$ of the equation $x^m - 1 = 0$. It is called *primitive* if m is the smallest integer such that $x^k - 1 = 0$ for $k < m$.

Definition 3.2. A *cyclotomic field* of mth roots of unity is the splitting field of the polynomial $X^m - 1$ over the rational field $\mathbb{Q}$, written as $\mathbb{Q}(\zeta_m)$ or $K = \mathbb{Q}(x)/\Phi_m(x)$, where $\Phi_m(x)$ is the minimal polynomial.

Let $K = \mathbb{Q}(\zeta_m)$ be a cyclotomic field where $m = q = p^a$ for a prime $p \in \mathbb{Z}$ and integer $a > 0$. The Euler function at q is $\phi(q) = p^{a-1}(p-1)$ and $\Phi_m(x)$ is the polynomial

$$\Phi_m(x) = \prod_{(m,j)=1} (x - \zeta_m^j) = \frac{x^{p^a} - 1}{x^{p^{a-1}} - 1} = t^{p-1} + t^{p-2} + ... + t + 1,$$

where $t = x^{p^{a-1}}$. We have the following properties for a cyclotomic field.

Properties 3.3.

1. $[\mathbb{Q}(\zeta_m) : \mathbb{Q}] = \phi(q) = p^{a-1}(p-1)$ is the degree of the field extension.
2. $\Phi_m(x)$ is irreducible over $\mathbb{Q}$ and is the minimal polynomial of ζ_m.
3. The cyclotomic ring is $R = \mathbb{Z}[\zeta_m]$.
4. The discriminant of $K/\mathbb{Q}$ is $\Delta_K = \pm p^{p^{a-1}(ap-a-1)}$.
5. The prime $p \in \mathbb{Z}$ is the only prime that ramifies in R with ramification index $e = \Phi_m(q) > 1$.
6. $\mathrm{Gal}(\mathbb{Q}(\zeta_m)/\mathbb{Q}) \simeq (\mathbb{Z}/m\mathbb{Z})^*$.
7. The element $\alpha = 1 - \zeta_m$ generates a prime ideal $\alpha R \subset R$ and $pR = (\alpha R)^{\phi(m)}$ for the prime element $p \in \mathbb{Z}$, where $p|m$.

Remark 3.4. We will be working with cyclotomic fields where $m = 2n$ and $n = 2^k$. Then the minimal polynomial is $\Phi_m(x) = x^n + 1$ and the cyclotomic field can be written on the form $K = \mathbb{Q}(x)/(x^n + 1)$ with the corresponding cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$.

From Property 6 in 3.3 we define the following embedding.

Definition 3.5. Let $G = \mathrm{Gal}(\mathbb{Q}(\zeta_m)/\mathbb{Q})$ be the Galois group such that $\sigma_j \in G$ is an embedding of K into $\mathbb{C}$ defined as,

$$\begin{aligned} \sigma_j : K &\longrightarrow \mathbb{C} \\ x &\longmapsto \zeta^j \end{aligned}$$

for all $j \in (\mathbb{Z}/m\mathbb{Z})^*$.

Since cyclotomic fields are separable field extensions, the Galois group and the embedding defined above, can be used to calculate the norm and trace for elements in K.

Theorem 3.6 (Janusz [6]). *For an element $a \in K$ we have*

- *The trace of K over $\mathbb{Q}$ is $T_{K/\mathbb{Q}}(a) = \sum_i^n \sigma_i(a)$,*
- *The norm of K over $\mathbb{Q}$ is $\mathcal{N}_{K/\mathbb{Q}}(a) = \prod_i^n \sigma_i(a)$.*

An important part of S-unit attacks is the prime ideals. Therefore, we need to know how the prime ideals behave and how to find them for a cyclotomic ring. We use the following two theorems to identify the prime ideals, which we later use in Section 3.2.1 to find a generator for the same prime ideals.

Theorem 3.7 (Washington [15]). *Let p be a prime such that $p \nmid m$ and let f be the smallest positive integer such that $p^f \equiv 1 \pmod{m}$. Then p splits into $g = \phi(m)/f$ distinct primes in $\mathbb{Q}(\zeta_m)$ each with residue class degree f. In particular, p splits completely $\Leftrightarrow p \equiv 1 \pmod{m}$.*

Theorem 3.8 (Kummers Theorem [15]). *Let A be a Dedekind domain with quotient field K, let L/K be a finite separable extension, and let B be the integral closure of A in L. Suppose $B = A[\alpha]$ for some $\alpha \in L$ and let $f(x)$ the irreducible polynomial for α over K. Let $\mathfrak{p}$ be a prime ideal of A and $\overline{f(x)}$ denote reduction modulo $\mathfrak{p}$. Suppose*

$$\overline{f(x)} = \overline{g_1^{e_1}} \dots \overline{g_t^{e_t}}$$

is the factorization of $f(x)$ mod $\mathfrak{p}$ into powers of distinct monic irreducible polynomials over $(A/\mathfrak{p})[x]$. Let $g_i(x) \in A[x]$ be a monic polynomial which reduces mod $\mathfrak{p}$ to $\overline{g_i(x)}$. Let

P_i be the ideal of B generated by $\mathfrak{p}$ and $g_i(\alpha)$. Then P_i is a prime ideal of B lying over $\mathfrak{p}$, e_t is the ramification index, the P_i's are distinct, and

$$\mathfrak{p}B = P_1^{e_1} \cdots P_t^{e_t}$$

is the factorization of $\mathfrak{p}$ in B.

Again, since a cyclotomic field is a splitting field over $\mathbb{Q}$, i.e. $K/\mathbb{Q}$ is a finite separable extension, we can apply this theorem by factoring the minimal polynomial $\Phi_m(x)$ mod p as

$$\overline{\Phi_m(x)} = \overline{g_1(x)^{e_1}} \dots \overline{g_t(x)^{e_t}}.$$

Then $P_i = (p, g_i(\zeta_m))$ is a prime ideal in R. Furthermore, by Remark 3.4 the minimal polynomial is on the form

$$\Phi_m(x) = x^n + 1 = \prod_{(m,j)=1} (x - \zeta_m^j),$$

and whenever $p \equiv 1 \pmod{m}$ we have that there exists an integer a such that $\Phi_m(a) = \prod_{(m,j)=1}(a - \zeta_m^j) \equiv 0 \pmod{p}$. Hence, $P_i = (p, a - \zeta_m)$ will be a prime ideal in R.

Lastly, we need to introduce the maximal real embedding of a cyclotomic field. For any embedding of $\mathbb{Q}(\zeta_m)$ into $\mathbb{C}$, the complex conjugate acts as an automorphism by sending $\zeta_m \longmapsto \zeta_m^{-1}$. This can be used to define the maximal real embedding of $\mathbb{Q}(\zeta_m)$.

Definition 3.9. The *maximal real embedding* of $\mathbb{Q}(\zeta_m)$ is

$$\mathbb{Q}(\zeta_m)^+ = \mathbb{Q}(\zeta_m + \zeta_m^{-1}) = \mathbb{Q}(\cos(2\pi/m)).$$

The extension $\mathbb{Q}(\zeta_m)/\mathbb{Q}(\zeta_m + \zeta_m^{-1})$ is of degree 2, i.e. ζ_m is a root of the polynomial $x^2 - (\zeta_m + \zeta_m^{-1})x + 1$, and the associated ring is $R^+ = \mathbb{Z}(\zeta_m + \zeta_m^{-1})$.

3.1.2 Class Group

The theory of class groups is important when considering attack against lattice-based cryptography. We give a brief overview of the theory, and for the rest of the thesis we only focus on the part pertaining to principal ideal domains (PIDs).

Definition 3.10. A *fractional ideal* of R is a finitely generated R-submodule $I \subseteq K$ such that there exists a nonzero element $r \in R$ where $rI \subseteq R$. Each fractional ideal I has an inverse $I^{-1} = \{a \in K \mid aI \subseteq R\}$.

The set of fractional ideals of R forms an abelian group, where $R = (1)$ is the identity, the product of two fractional ideals I, J is defined as $IJ = \{\sum_i a_i b_i \mid a_i \in I, b_i \in J\}$ and I is invertible if there exists a fractional ideal I^{-1} such that $II^{-1} = R$.

From this we define the class group, which can be used as a measure of how far an ring of algebraic integers R is from being a PID.

Definition 3.11. Let $\mathbf{I}(R)$ be the collection of all fractional ideals of R, the *ideal group*, and $\mathbf{P}(R)$ the collection of all the principal ideals. Then the *class group* of R is defined as

$$\mathbf{C}(R) = \mathbf{I}(R)/\mathbf{P}(R).$$

The order of $\mathbf{C}(R)$ is the *class number* of K, denoted h_K. Notice that if $\mathbf{C}(R) = 1$, then R is a PID.

The class group can be seen as the set of equivalence classes of nonzero fractional ideals of R. Two nonzero ideals α and β are equivalent if $\alpha = \beta c$ for some nonzero $c \in K$. Then the equivalence class of α is denoted by $[\alpha]$.

In lattice-based cryptography the cyclotomic rings are of high dimensions, which makes them more likely to not be PIDs. If that is the case we would first need to find the principal representations of the ideals in the ring, called an *S-generator*, and then perform an S-unit attack. Whenever m is a power of 2 it is conjectured [10] that all cyclotomic rings have class number 1 in the maximal real subfield. This means that for sufficiently large values of m, we can first map the elements to the maximal real subfield, and then find a short generator using S-unit attacks. For cyclotomic rings where m is not a power of 2, Jean-François Biasse and Fang Song [7], [3] have described quantum algorithms for finding S-generators. However, in our case we choose small values for m such that the cyclotomic rings are PIDs.

3.1.3 Cyclotomic Units

In a general algebraic number field it can be hard to determine all the units in the ring of integers. However, for cyclotomic rings we have something called cyclotomic units, which can be given explicitly with finite index in the full group of units.

An element $a \in R$ is a unit if there exists an element $b \in R$ such that $ab = 1$ in R. Moreover, a unit will always have norm equal to 1 and vice versa, i.e. $u \in R$ is a unit $\Leftrightarrow \mathcal{N}(u) = 1$. Let r_1 be the number of real embeddings of K and r_2 be the number of complex embeddings of K. We have the following theorem describing the full group of units.

Theorem 3.12 (Dirichlet Unit Theorem [6]). *The group of units U_K in the ring of algebraic integers R of an algebraic number field K can be written as the direct product of a finite cyclic group and a free abelian group of rank $r_1 + r_2 - 1$,*

$$U_K = \mu(K) \times \mathbb{Z}^{r_1+r_2-1}$$

where $\mu(K)$ is the group generatet by the roots of unity. Equivalently, there exists units $u_1, ..., u_{r_1+r_2-1} \in R$ such that every $u \in R$ can be written as

$$u = w u_1^{b_1} \cdots u_{r_1+r_2-1}^{b_{r_1+r_2-1}}$$

for some root of unity w and integer b_j.

This states that the unit group can be infinite, which is hard to work with, so we introduce the cyclotomic units to make it is possible to define the units to use for an S-unit attack.

Definition 3.13. Let U_K be the group of units of $\mathbb{Z}[\zeta_m]$ and V_K be the multiplicative group generated by $\{\pm\zeta_m, 1 - \zeta_m^a \mid 1 \leq a \leq m-1\}$, then

$$C_K := U_K \cap V_K$$

is the group of *cyclotomic units* of $\mathbb{Z}[\zeta_m]$.

First of all, the roots of unity ζ_m^a for odd $a < m$ are units, because for each ζ_m^a there exists a nonzero b such that $a \cdot b \equiv 0 \pmod{m}$ and $\zeta_m^{ab} = 1$. Next, we need some results to understand how we can determine the group of cyclotomic units C_K more explicit.

Lemma 3.14 (Washington [15]). *If α is an algebraic integer where all of its conjugates have absolute value 1, then α is a root of unity.*

Proof. Let α be an algebraic integer, then α is a root of an irreducible polynomial with coefficients in $\mathbb{Z}$. Since the coefficients are bounded by the degree of α over $\mathbb{Q}$, there can only be finitely many irreducible polynomials that has a power of α as a root. Hence, there are only finitely many distinct powers of α, and therefore a root of unity. □

Theorem 3.15 (Washington [15]). *Let u be a unit of $\mathbb{Z}[\zeta_m]$. Then there exists a unit $u_1 \in \mathbb{Z}[\zeta_m + \zeta_m^{-1}]$ and $r \in \mathbb{Z}$ such that $u = \zeta_m^r u_1$.*

Proof. Let $\zeta_m = \zeta$ and $\alpha = u/\overline{u}$. Then $\alpha \in R$ since $\overline{u}$ is a unit and α is a root of unity by Lemma 3.14. So we can write $\alpha = \pm\zeta^a$ for some a. If $\alpha = -\zeta^a = u/\overline{u}$, write $u = b_0 + b_1\zeta + ... + b_{m-2}\zeta^{m-2}$. Then

$$u \equiv b_0 + b_1 + ... + b_{m-2} \pmod{(1-\zeta)},$$

and

$$\begin{aligned}\overline{u} &= b_0 + b_1\zeta^{-1} + ... + b_{m-2}\zeta^{m-2} \\ &\equiv b_0 + b_1\zeta + ... + b_{m-2}\zeta^{m-2} \pmod{(1-\zeta)} \\ &= u = -\zeta^a\overline{u} = -\overline{u}.\end{aligned}$$

This implies that $2\overline{u} \equiv 0 \pmod{(1-\zeta)}$, but $2 \notin (1-\zeta)$ and $(1-\zeta)$ is a prime ideal, so we also have that $\overline{u} \notin (1-\zeta)$ since $\overline{u}$ is a unit. Hence, we can conclude that $\alpha = u/\overline{u} = \zeta^a$.

Now, let $2r \equiv a \pmod{m}$ and $u_1 = \zeta^{-r}u$, then $u = \zeta^r u_1$ and $\overline{u_1} = u_1$, which is what we wanted since, $u/\overline{u} = \zeta^a = \zeta^{2r} \Rightarrow u = \zeta^{2r}\overline{u} = \zeta^{2r}\zeta^{-r}\overline{u_1} = \zeta^r\overline{u_1} = \zeta^r u_1$. □

We can now explicitly describe the cyclotomic units in a cyclotomic ring and justify the choice of the cyclotomic units we will be using for S-unit attacks.

Theorem 3.16 (Washington [15]). *Let $m = p^n$ for prime p and $n \geq 1$.*

1. *The cyclotomic units u of $\mathbb{Z}[\zeta_m]^+$ are generated as a multiplicative group by $\{-1\}$ and the units $u_a = \zeta_m^{(1-a)/2} \cdot \frac{1-\zeta_m^a}{1-\zeta_m}$, where $1 < a < m/2$ and $\gcd(a,p) = 1$.*
2. *The cyclotomic units of $\mathbb{Z}[\zeta_m]$ are generated by ζ_m and the cyclotomic units $u \in \mathbb{Z}[\zeta_m]^+$.*

Theorem 3.17 (Washington [15]). *Let $m = p^a$ and g be a primitive root $\mod m$. Then*

$$\zeta_m^{(1-g)/2}\frac{1-\zeta_m^g}{1-\zeta_m}$$

generates $C_K^+/\{\pm 1\}$ as a module over $\mathbb{Z}[\mathrm{Gal}(\mathbb{Q}(\zeta_m)^+/\mathbb{Q})]$.

Proof. Let $\gcd(a,p) = 1$, then $a \equiv g^r \mod m$ for some $r > 0$ and

$$\begin{aligned}\zeta_m^{(1-a)/2}\frac{1-\zeta_m^a}{1-\zeta_m} = \zeta_m^{(1-g^r)/2}\frac{1-\zeta_m^{g^r}}{1-\zeta_m} &= \prod_{i=0}^{r-1}\zeta_m^{(g^i-g^{i+1})/2}\frac{1-\zeta_m^{g^{i+1}}}{1-\zeta_m^i} \\ &= \prod_{i=0}^{r-1}\sigma_i\left(\zeta_m^{(1-g)/2}\frac{1-\zeta_m^g}{1-\zeta_m}\right).\end{aligned}$$

The rest follows from Theorem 3.16. □

Notice that we are working with $p = 2$, so by letting $g = -3$ we have that $a \equiv (-3)^r$ or $-a \equiv (-3)^r \mod 2^a$. This gives us the cyclotomic units on the form $1 + \zeta_m^c + \zeta_m^{-c}$ for odd $c < m$ as follows,

$$\begin{aligned}
\zeta_m^{(1-a)/2}\frac{1-\zeta_m^a}{1-\zeta_m} &= \zeta_m^{(1-(-3)^r)/2}\frac{1-\zeta_m^{(-3)^r}}{1-\zeta_m} = \prod_{i=0}^{r-1}\zeta_m^{((-3)^i-(-3)^{i+1})/2}\frac{1-\zeta_m^{(-3)^{i+1}}}{1-\zeta_m^i} \\
&= \prod_{i=0}^{r-1}\sigma_i\left(\zeta_m^{(1+3)/2}\frac{1-\zeta_m^{-3}}{1-\zeta_m}\right) = \prod_{i=0}^{r-1}\sigma_i\left(\zeta_m^2\frac{-\zeta_m^2-\zeta_m-1}{\zeta_m^3}\right) \\
&= \prod_{i=0}^{r-1}\sigma_i\left(\zeta_m^{-1}(-\zeta_m^2-\zeta_m-1)\right) = \prod_{i=0}^{r-1}\sigma_i\left((-1)(1+\zeta_m+\zeta_m^{-1})\right).
\end{aligned}$$

By the following lemma, these cyclotomic units generate a subgroup of finite index in the full group of units.

Lemma 3.18 (Washington [15]). *Let $K/\mathbb{Q}$ be a finite Galois extension. If K is real let $\sigma_1, ..., \sigma_{r+1}$ be the elements of Gal($K/\mathbb{Q}$). If K is complex let $\sigma_1, ..., \sigma_{r+1}, \overline{\sigma_1}, ..., \overline{\sigma_{r+1}}$ be elements of Gal($K/\mathbb{Q}$). There exists a unit $u \in R$ such that the set of units $\{\sigma_i(u) \mid 1 \leq i \leq r\}$ is multiplicative independent, hence generates a subgroup of finite index in the full group of units, called a Minkowski unit.*

In conclusion, let $x = \zeta_m$ such that the cyclotomic units can be written on the form

$$u_0 = x \quad \text{and} \quad u_c = 1 + x^c + x^{-c},$$

for odd $c < m$. In the cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$ we have that $x^n = -1$, which implies that $x^c = x^n x^{c-n} = -x^{c-n}$ for $c > n$ and $x^{-c} = -x^{n-c}$ for $c < n$. Therefore, we have $m/2$ cyclotomic units on the form

$$1 + x + x^{-1}, 1 + x^3 + x^{-3}, ..., 1 + x^{n-1} + x^{-(n-1)}.$$

Because of the relation $u_1 u_3 \cdots u_{n-1} = \pm 1$, and by Dirichlets unit theorem, we have $m/2 - 1$ multiplicative independent units

$$1 + x + x^{-1}, 1 + x^3 + x^{-3}, ..., 1 + x^{n-3} + x^{-(n-3)},$$

with finite index in the full group of units U_K. These are the generators we will use for the group of cyclotomic units C_K and the units we will use for S-unit attacks.

3.1.4 Embeddings

Now that we have some basic knowledge about cyclotomic fields, we review the theory that connects them to lattices and see how they relate to lattice-based cryptography. There are two important embeddings, the Minkowski and the logarithmic embedding.

Let $m = 2n$ and $K = \mathbb{Q}(x)/(x^n + 1)$ be the cyclotomic field with the cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$. Let $G = \text{Gal}(K/\mathbb{Q})$ be the Galois group such that $\sigma_j \in G$ are all the embeddings of K into $\mathbb{C}$. Let r_1 be the number of embeddings into $\mathbb{R}$ and r_2 be the pairs of conjugate embeddings into $\mathbb{C}$, where $\sigma_j \neq \overline{\sigma_j}$ and $\overline{\sigma_j}(a) = \overline{\sigma_j(a)}$. First, define the map $v : K \to \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ as

$$v(a) = (\sigma_1(a), ..., \sigma_{r_1}(a), ..., \sigma_{r_1+r_2}(a)).$$

Since $\mathbb{C}$ has a structure as a two-dimensional $\mathbb{R}$-vector space, we can view $V = \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ as an $t = r_1 + 2r_2$-dimensional $\mathbb{R}$-vector space, and we can define the embedding v as

$$\begin{aligned} v(a) = (\sigma_1(a), ..., &\sigma_{r_1}(a), \\ &\mathrm{Re}(\sigma_{r_1+1}(a)), \mathrm{Im}(\sigma_{r_1+1}(a)), ..., \mathrm{Re}(\sigma_{r_1+r_2}(a)), \mathrm{Im}(\sigma_{r_1+r_2}(a))). \end{aligned}$$

Then $v(a)$ is a vector in $\mathbb{R}^t$ for $t = r_1 + 2r_2$, and we have the following theorem, which is useful when estimating the size of the smallest element for a given ideal.

Theorem 3.19 (Modified theorem from Janusz [6]). *Let $\alpha \subseteq R$ be a nonzero ideal, then*

1. *$v(\alpha)$ is a full lattice in $\mathbb{R}^t$ with volume* $\mathrm{Vol}(v(\alpha)) = 2^{-r_2}\mathcal{N}(\alpha)|\Delta_R|^{1/2}$,
2. *There exists a nonzero element $a \in \alpha$ such that*
$$|\mathcal{N}_{K/\mathbb{Q}}(a)| \leq \frac{t!}{t^t}\left(\frac{4}{\pi}\right)^{r_2}\mathcal{N}(\alpha)|\Delta_R|^{1/2},$$
3. *For any nonzero fractional ideal β of R there is an $\alpha \in [\beta]$ such that $\alpha \subseteq R$ and*
$$\mathcal{N}(\alpha) \leq \frac{t!}{t^t}\left(\frac{4}{\pi}\right)^{r_2}|\Delta(R/\mathbb{Z})|^{1/2}.$$

The logarithmic embedding is mostly used for units, and is the embedding we utilize for S-unit attacks. Let U_K be the group of units in R and define the function $\mathrm{Log} : K^* \to \mathbb{R}^{r_1+r_2}$ as

$$\mathrm{Log}(a) = (\log|\sigma_1(a)|, ..., \log|\sigma_{r_1}(a)|, 2\log|\sigma_{r_1+1}(a)|, ..., 2\log|\sigma_{r_1+r_2}(a)|).$$

Notice that for a unit $u \in R$ we have the norm $\mathcal{N}(u) = \prod_i \sigma_i(u) = 1$, which implies that $\log|\mathcal{N}(u)| = \log|\prod_i \sigma_i(u)| = \sum_i \log|\sigma_i(u)| = 0$. The following theorem embedding the unit group as a lattice.

Theorem 3.20 (Janusz [6]). *The homomorphism* Log *maps the unit group U_K of R onto a lattice in the $r_1 + r_2 - 1$ dimensional subspace of $V_0 = \mathbb{R}^{r_1+r_2}$ consisting of all vectors*

$$\{v = (v_1, ..., v_{r_1+r_2}) \in \mathbb{R}^{r_1+r_2} \mid \sum v_i = 0\}.$$

Proof. For $u \in U_K$ we have that $\mathcal{N}(u) = \pm 1$, which gives the following relation of the coordinates of $\log(u)$,

$$\mathrm{Log}(u) = (\log|\sigma_1(u)|, ..., \log|\sigma_{r_1}(u)|, 2\log|\sigma_{r_1+1}(u)|, ..., 2\log|\sigma_{r_1+r_2}(u)|)$$

$$\begin{aligned} \Rightarrow \sum_{i=1}^{r_1}\log|\sigma_i(u)| + \sum_{i=r_1+1}^{r_1+r_2} 2\log|\sigma_i(u)| &= \log|\sigma_1(u)\ldots\sigma_{r_1}(u)\sigma_{r_1+1}(u)^2\ldots\sigma_{r_1+r_2}(u)^2| \\ &= \log|\mathcal{N}(u)| = \log|\pm 1| = 0. \end{aligned}$$

Hence, $\mathrm{Log}(U_K)$ lies in the hyperplane of V_0. To prove that it is a lattice we need to show that every cube in V_0 contains a finite number of points in $\mathrm{Log}(U_K)$. Let $\delta_i = 1$ for $1 \leq i \leq r_1$ and $\delta_i = 2$ for $r_1 + 1 \leq i \leq r_1 + r_2$, and define $\log_i(u) = \delta_i \log|\sigma_i(u)|$. Let $U_{K,a}$ be the set of all $u \in U_K$ such that $|\log_i(u)| \leq a$ for a positive constant a. Then $\mathrm{Log}(u)$ is in the cube with center at the origin and sides of length $2a$. Further,

$$|\log_i(u)| = |\delta_i \log|\sigma_i(u)|| \leq a \Rightarrow |\sigma_i(u)| \leq e^{a/\delta_i},$$

and by the Minkowski embedding $v : K \to \mathbb{R}^{r_1+2r_2}$ we can conclude that $v(U_{K,a})$ lies in a bounded subset of $v(R)$. Further, we know that $v(R)$ is a lattice hence $v(U_{K,a})$ is a finite set. Then $U_{K,a}$ is also a finite set because v is injective, so $\mathrm{Log}(U_{K,a})$ is finite and we have that $\mathrm{Log}(U_K)$ is a lattice. □

3.1.5 Characters

For an S-unit attack we want to work with principal ideals such that we can find a generator. As discussed in Section 3.1.2 we are working in a PID, so this should be possible, but it gets harder and harder as the dimension for the cyclotomic ring grows. Therefore, we will present a somewhat easier way to find the prime ideal factorization for a prime element p and a generator for those prime ideals. To do so we need to introduce two types of characters, namely multiplicative and additive.

Definition 3.21. Multiplicative character

1. A *multiplicative character* χ is the group homomorphism
$$\chi : \mathbb{F}_p^* \longrightarrow \mathbb{C}^*.$$
2. The trivial multiplicative character is denoted by ϵ, and $\epsilon(a) = 1$ for all $a \in \mathbb{F}_p^*$.
3. We can extend χ to $\mathbb{F}_p$ by letting $\chi(0) = 0$ and $\epsilon(0) = 1$ for the trivial character.
4. If χ is defined with an order m, it is the smallest m such that χ is periodic, i.e. $\chi(a+m) = \chi(a)$ for $a \in \mathbb{Z}$.

Definition 3.22. Additive character

1. An *additive character* ψ is a group homomorphism
$$\psi : \mathbb{F}_p \longrightarrow \mathbb{C}^*.$$
2. The trivial additive character is denoted by ψ_0, and $\psi_0(a) = 0$ for all $a \in \mathbb{F}_p$.

Remark 3.23. In addition to the definitions above there are some important notations to remark.

- We will denote the additive character by $\psi(a) = \zeta_p^a$, for $a \in \mathbb{F}_p$.
- The order m of the multiplicative character χ is co-prime to p and divides $|\mathbb{F}_p^*| = p-1$.
- The additive character ψ have order p.
- The inverses are $\overline{\chi}(a) = \chi^{-1}(a) = \chi(a^{-1})$ and $\overline{\psi}(a) = \psi^{-1}(a) = \psi(-a) = \zeta_p^{-a}$.

3.1.6 Gauss Sum and Jacobi Sum

Let $p \in \mathbb{Z}$ be a prime number, we want to find the prime ideals in the cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$ containing p, by using characters.

Definition 3.24. Let χ be a multiplicative character and ψ a nontrivial additive character of $\mathbb{F}_p$. The *Gauss sum* for these characters is defined as
$$\tau(\chi) = \tau(\chi, \psi) = \sum_{a \in \mathbb{F}_p^*} \chi(a)\psi(a) = \sum_{a \in \mathbb{F}_p^*} \chi(a)\zeta_p^a.$$

Notice that if χ has order m, then $\tau(\chi) \in \mathbb{Q}(\zeta_{mp})$ since $\chi \in \mathbb{Q}(\zeta_m)$ and $\psi \in \mathbb{Q}(\zeta_p)$. The following lemma shows that the Gauss sum $\tau(\chi)$ will contain exactly the prime ideals in R that lies above p.

Lemma 3.25 (Washington [15]). *If $\chi \neq 1$, then $|\tau(\chi)|^2 = \tau(\chi)\overline{\tau(\chi)} = p$.*

Proof. For $a, b \in \mathbb{F}_p$ we get,

$$\begin{aligned}
\tau(\chi)\overline{\tau(\chi)} &= \sum_{a,b\neq 0} \chi(a)\zeta_p^a \overline{\chi(b)\zeta_p^b} \\
&= \sum_{a,b\neq 0} \chi(a)\zeta_p^a \chi(b^{-1})\zeta_p^{-b} \\
&= \sum_{a,b\neq 0} \chi(ab^{-1})\zeta_p^{a-b} && \text{Let } c = ab^{-1} \\
&= \sum_{b,c\neq 0} \chi(c)\zeta_p^{b(c-1)} \\
&= \sum_{b\neq 0} \chi(1)\zeta_p^0 + \sum_{c\neq 0,1} \chi(c) \sum_{b\neq 0} \zeta_p^{b(c-1)} \\
&= (p-1) + \sum_{c\neq 0,1} \chi(c)(-1) = p.
\end{aligned}$$

The last part comes from the fact that $\chi(1) = 1$ and $b \in \mathbb{F}_p^*$, which means

$$\sum_{b\neq 0} \chi(1)\zeta_p^0 = \sum_{b\neq 0} 1 = p - 1$$

and

$$\begin{aligned}
&1 + \zeta_p^{(c-1)} + \zeta_p^{2(c-1)} + \dots + \zeta_p^{(p-1)(c-1)} = 0 \\
&\Rightarrow 1 + \sum_{b\neq 0} \zeta_p^{b(c-1)} = 0 \\
&\Rightarrow \sum_{b\neq 0} \zeta_p^{b(c-1)} = -1.
\end{aligned}$$

Likewise $\sum_{c\neq 0,1} -\chi(c) = 1$, since $\sum_{c\neq 0} \chi(c) = 0$ in the same manner as above. □

Definition 3.26. Let χ_1 and χ_2 be two multiplicative characters on $\mathbb{F}_p$. The *Jacobi sum* is defined as

$$J(\chi_1, \chi_2) = \sum_{\substack{a_i \in \mathbb{F}_p \\ a_1 + a_2 = 1}} \chi_1(a_1)\chi_2(a_2) = \sum_{a \in \mathbb{F}_p} \chi_1(a)\chi_2(1-a).$$

Now, if we combine the Gauss and Jacobi sum, we get a very useful result which tells us that the Jacobi sum can be written by Gauss sums, it is integral and it eliminates ζ_p.

Corollary 3.27 (Washington [15]). *Let χ_1, χ_2 have orders dividing m, where m is an integer not dividing p, and $\chi_1\chi_2 \neq 1$. Then*

$$J(\chi_1, \chi_2) = \tau(\chi_1)\tau(\chi_2)/\tau(\chi_1\chi_2) \in \mathbb{Z}[\zeta_m]$$

and

$$J(\chi^b, \chi) = \tau(\chi)^b\tau(\chi)/\tau(\chi^{b+1})$$

for $b \in \mathbb{Z}$.

For S-unit attacks we use Jacobi sums on the following form. Let n be the degree of the minimal polynomial of the cyclotomic ring R and let $p \in 1 + 2n\mathbb{Z}$ be a prime element. For $J(\chi_1, \chi_2)$ define $\chi_1 = \chi^i$ for some $i \in \mathbb{Z}$ and $\chi_2 = \chi$, such that we have the Jacobi sum

$$J(\chi_1, \chi_2) = J(\chi^i, \chi) = \sum_{a \in \mathbb{F}_{p^*-1}} \chi^i(a)\chi(1-a).$$

Then by Corollary 3.27 we have that

$$J(\chi^i, \chi) = \tau(\chi)^i \tau(\chi)/\tau(\chi^{i+1}) \in R,$$

and by Lemma 3.25 we have that

$$|J(\chi_1, \chi_2)|^2 = |\tau(\chi_1)\tau(\chi_2)/\tau(\chi_1\chi_2)|^2 = |\tau(\chi_1)|^2 |\tau(\chi_2)|^2 / |\tau(\chi_1\chi_2)|^2 = p^2/p = p.$$

Hence, the Jacobi sum is an element in R acting as a generator for an ideal. Also, the prime ideal factorization of the ideal contains only prime ideals that lies above p. To end this section we give an example that verifies this theory.

Example 3.28. In this example we show that the Gauss and Jacobi sum produce the same element for a character of order m relatively prime to p. We also show that this element is in the cyclotomic ring R with the prime ideal factorization of prime ideals lying above p. Most of the computations are done in Sage, see appendix B.1 for the Sage code.

Let $n = 2$ and $m = 2n = 4$ such that $R = \mathbb{Z}[x]/(x^2 + 1)$. Define the multiplicative character to be $\chi(2) = \zeta_4$. Since m is relatively prime to p, χ is defining a group for $p = 1 + 2n \cdot 3 = 13$. We get the following group elements,

$$\begin{array}{ll}
\chi(2) = \chi(2) = \zeta_4 & \chi(2^7) = \chi(11) = \zeta_4^7 \\
\chi(2^2) = \chi(4) = \zeta_4^2 & \chi(2^8) = \chi(9) = \zeta_4^8 \\
\chi(2^3) = \chi(8) = \zeta_4^3 & \chi(2^9) = \chi(5) = \zeta_4^9 \\
\chi(2^4) = \chi(3) = \zeta_4^4 & \chi(2^{10}) = \chi(10) = \zeta_4^{10} \\
\chi(2^5) = \chi(6) = \zeta_4^5 & \chi(2^{11}) = \chi(7) = \zeta_4^{11} \\
\chi(2^6) = \chi(12) = \zeta_4^6 & \chi(2^{12}) = \chi(1) = \zeta_4^{12}.
\end{array}$$

From this we calculate the Gauss sum for $p = 13$, as follows

$$\begin{aligned}
\tau(\chi) = \sum_{a \in \mathbb{F}_{13^*}} \chi(a)\zeta_{13}^a &= \chi(1)\zeta_{13}^1 + \chi(2)\zeta_{13}^2 + \chi(3)\zeta_{13}^3 + ... + \chi(12)\zeta_{13}^{12} \\
&= 1 \cdot \zeta_{13}^1 + i \cdot \zeta_{13}^2 + 1 \cdot \zeta_{13}^3 + ... + (-1) \cdot \zeta_{13}^{12}.
\end{aligned}$$

Notice that $\zeta_4 = e^{2\pi i/4} = i$ such that $\chi(2^b) = \zeta_4^b = i^b$. Now, let $x = \zeta_{13}$. Then the Gauss sum is

$$\tau(\chi) = x + ix^2 + x^3 - x^4 + ix^5 + ix^6 - ix^7 - ix^8 + x^9 - x^{10} - ix^{11} - x^{12}.$$

From this we get the Gauss sums

$$\tau(\chi^2) = x - x^2 + x^3 + x^4 - x^5 - x^6 - x^7 - x^8 + x^9 + x^{10} - x^{11} + x^{12}$$

and

$$\tau(\chi)^2 = (3 - 2i) \cdot (x - x^2 + x^3 + x^4 - x^5 - x^6 - x^7 - x^8 + x^9 + x^{10} - x^{11} + x^{12}).$$

By corollary 3.27 we get $\tau(\chi)^2/\tau(\chi^2) = 3 - 2i \in R$, which is in fact a prime ideal in R, $(3-2i)(3+2i) = 13$ and

$$|\tau(\chi)|^2 = \tau(\chi)\overline{\tau(\chi)} = 13.$$

The Gauss sum computations increases in complexity for higher values of m, but computing the Jacobi sum directly give the same element with fewer computations. Let $\chi_1 = \chi = \chi_2$. By Definition 3.26 we get

$$\begin{aligned}
J(\chi_1,\chi_2) = J(\chi,\chi) &= \sum_{a\in\mathbb{F}_{13^*-1}} \chi(a)\chi(1-a)\\
&= \chi(2)\chi(1-2) + \chi(3)\chi(1-3) + ... + \chi(12)\chi(1-12)\\
&= i\cdot(-1) + 1\cdot(-i) + (-1)\cdot(-1) + i\cdot 1 + i\cdot(-i) +\\
&(-i)\cdot(-i) + (-i)\cdot 1 + 1\cdot i + (-1)\cdot(-1) + (-i)\cdot 1 + (-1)\cdot i\\
&= 3 - 2i = \tau(\chi)^2/\tau(\chi^2).
\end{aligned}$$

Hence, the prime ideals in R lying above $p = 13$ are $P_1 = (3-2i)$ and $P_2 = (3+2i)$.

3.2 S-units

Now that we have established which units and prime ideals to use in an S-unit attack, we can define the S-unit group and the logarithmic embedding for these elements. At the end we provide a method for finding the generators for the S-unit group.

Let $n = 2^k$ and $m = 2n$ such that we have the cyclotomic ring $R = \mathbb{Z}[x]/(x^n+1)$. Recall that the Galois group $\mathrm{Gal}(K/\mathbb{Q})$ gives us the ring homomorphism

$$\begin{aligned}
\sigma_c : K &\longrightarrow \mathbb{C}\\
x &\longmapsto \zeta_m^c
\end{aligned}$$

for each odd $c < m$. To define the S-units we need some notion about $\mathfrak{p}$-adic valuations and two norms, namely the infinite and the finite norm for cyclotomic fields.

Definition 3.29. Let K be a number field. The map $v_{\mathfrak{p}} : K^* \longrightarrow \mathbb{Z}$ is defined as the $\mathfrak{p}$-*adic valuation* of nonzero elements in K where $v_{\mathfrak{p}}$ satisfy the following,

1. $v_{\mathfrak{p}}(a) \in \mathbb{Z}$ for each nonzero $a \in K$ and $v_{\mathfrak{p}}(0) = \infty$,
2. $v_{\mathfrak{p}}(ab) = v_{\mathfrak{p}}(a) + v_{\mathfrak{p}}(b)$,
3. $v_{\mathfrak{p}}(a+b) \geq \min\{v_{\mathfrak{p}}(a), v_{\mathfrak{p}}(b)\}$.

Let $I \subset K$ be a nonzero fractional ideal. The $\mathfrak{p}$-adic valuation of I is the exponent of the highest power of the prime ideal $\mathfrak{p}$ that appears in the prime ideal factorization of I. Let $I = \prod_i \mathfrak{p}_i^{a_i}$, then the $\mathfrak{p}_i$-adic valuation is

$$v_{\mathfrak{p}_i}(I) = a_i.$$

If $I \subset R$, then $a_i \leq 0$.

Definition 3.30. A function $|\cdot| : K \to \mathbb{R}$ is an *absolute value* or *norm* on K if it satisfies the following

1. $|a| \geq 0$ for every $a \in K$ and $|a| = 0$ if and only if $a = 0$.

2. $|a||b| = |ab|$ for all $a, b \in K$.
3. $|a+b| \leq C \max\{|a|, |b|\}$ for all $a, b \in K$ and some a positive constant C.

It is *nonarchimedean* if $C = 1$, and *archimedean* otherwise.

Definition 3.31. Let a be a nonzero element in R. The $\mathfrak{p}-$*adic norm*, or *finite norm*, of K is a nonarchimedean absolute value defined as

$$|a|_{\mathfrak{p}} = \mathcal{N}_{K/\mathbb{Q}}(\mathfrak{p})^{-v_{\mathfrak{p}}(a)}$$

for each nonzero prime ideal $\mathfrak{p} \subset R$.

Next, the usual absolute value, or the infinite norm, for an element $a \in \mathbb{Q}$ is defined as

$$|a|_\infty = \begin{cases} a & \text{if } x > 0 \\ -a & \text{if } x \leq 0. \end{cases}$$

We are now in a field extension of $\mathbb{Q}$, so we need to define the infinite norm differently. First, let K be the field extension of $\mathbb{Q}$ and let $(p)R = \prod_{i=1}^{c} P_i^{e_i}$ be the prime ideal factorization of the prime element $p \in \mathbb{Z}$.

Lemma 3.32 (Specialized version from Janusz [6]). *For an absolute value $|\cdot|_p$ in $\mathbb{Q}$, the extensions to the field K may be replaced by suitable powers to obtain a set of absolute values $|\cdot|_i$ on K, for $1 \leq i \leq c$, such that for every $a \in K$ we have*

$$\prod_i |a|_i = |\mathcal{N}_{K/\mathbb{Q}}(a)|_p.$$

Let $|a|_p = |a|_{\mathbb{C}}$, where $|\cdot|_{\mathbb{C}} = |\cdot|^2$ is the usual absolute value on $\mathbb{C}$. Let K be the field extension of $\mathbb{Q}$ such that we have all the embeddings $\sigma_1, ..., \sigma_c$ of K into $\mathbb{C}$. Let $|a|_i = |\sigma_i(a)|_{\mathbb{C}}$, then by Lemma 3.32 we have

$$\prod_i |a|_i = \prod_i |\sigma_i(a)|_{\mathbb{C}} = |\sigma(\mathcal{N}_{L/K}(a))|_{\mathbb{C}} = |\mathcal{N}_{L/K}(a)|_p.$$

We can now define the infinite norm of K.

Definition 3.33. The *infinite norm* of K is an achimedean absolute value defined as

$$|a|_c = |\sigma_c(a)|^2 = \sigma_c(a)\sigma_{-c}(a),$$

for $a \in K$ and for odd $c < m$.

By combining the finite and infinite norm we have the following theorem.

Theorem 3.34 (The product formula [6]). *Let K be an algebraic number field and p a prime number of K. Then there is an absolute value such that for each nonzero element $a \in K$ we have the formula*

$$\prod_{p \text{ prime}} |a|_p = 1,$$

where the product is taken over all prime numbers in K.

Proof. Let p be any prime in $\mathbb{Q}$ and let $|\cdot|_p$ be the normalized norm, consisting of the infinite and finite norm. Let $P_1, ..., P_c$ be the distinct prime ideals lying above p in K, then by lemma 3.32 we have that we can find an absolute value $|\cdot|_{P_i}$ such that

$$\prod_{i=1}^{c} |a|_{P_i} = |\mathcal{N}_{K/\mathbb{Q}}(a)|_p.$$

Further we have that

$$\prod_P |a|_P = \prod_p \prod_{P_i \text{ over } p} |a|_{P_i} = \prod_p |\mathcal{N}_{K/\mathbb{Q}}(a)|_p = 1.$$

The last equation follows from the product formula for $\mathbb{Q}$, where we have that

$$\prod_p |p|_p = |p|_\infty |p|_p = p \cdot \frac{1}{p}. \qquad \square$$

Now, let S be a finite set in R where $\{\infty\} \subset S$ and for each prime ideal $P \subseteq R$ we have

$$S = \{\infty\} \cup \{P \in R \mid \mathcal{N}(P) \leq y\}$$

for a chosen parameter y and $P \cap \mathbb{Z} \neq (2)$. The elements in S defines the finite and infinite norms that will be used for an S-unit attack. We exclude the prime 2 as a consequence from Proposition 3.3. Since $[K : \mathbb{Q}] = n = 2^k$, the prime $p = 2$ ramifies completely in R as $(2)R = (1 + x)^n$. It is the only prime that ramifies, and as a result the 2-adic valuation differs from all the other valuations. In Section 4.2 we discuss in more detail how to handle the prime ideal $P_2 = (1 + x)$.

Definition 3.35. The *S-units* are elements of the *S-unit group* U_S, generated by the units of R and the prime ideals $P \in S$ such that

$$U_S = \{u \in R^* \mid |u|_P = 1 \; \forall \; P \notin S\}.$$

Since the S-unit group is a finitely generated group with rank $r + s - 1$, where $r - 1$ is the rank of the unit group U_K and s is the number of prime ideals in S, Dirichlet unit theorem also holds for S-units.

Let $r + s$ be the number of elements in S, where r is the number of cyclotomic units corresponding to the infinite norm and s be the number of prime ideals corresponding to the finite norm. For an element $a \in R$ the logarithmic embedding for the set S is defined as

$$\log(a) = (\log|a|_1, ..., \log|a|_r, \log|a|_{P_1}, ..., \log|a|_{P_s}).$$

From Theorem 3.34 the sum of these coordinates are 0, so by Theorem 3.20 this will generate a lattice in $\mathbb{R}^{r+s-1}$ and we have the following definition.

Definition 3.36. The *S-unit lattice* is the logarithmic embedding of the S-unit group, $\mathcal{L}_S = \text{Log}(U_S)$, where the generators for the lattice are the logarithmic embedding of each generator for U_S.

We end the section with an example that demonstrates how to construct the S-unit lattice.

Example 3.37. Let $n = 4$ and $m = 2n = 8$ such that the cyclotomic ring is $R = \mathbb{Z}[x]/(x^4 + 1)$. The cyclotomic units are $u_0 = x$, $u_1 = 1 + x^1 + x^{-1}$ and $u_3 = 1 + x^3 + x^{-3}$, where the unit group C_K is generated by $\{u_0, u_1\}$. If we start with $S = \{\infty\}$, the logarithmic embedding is defined as

$$\text{Log}(a) = (\log|a|_1, \log|a|_3),$$

and the unit lattice $\mathcal{L}_U = \text{Log}(C_K)$ is generated by

$$\text{Log}(u_1) = (\log|u_1|_1, \log|u_1|_3) = (1.763, -1.763).$$

Now, choose the parameter for S to be $y = 10$. We want to find the prime ideals in R lying above all the prime numbers p less than 10, except 2. As described in Example 5.2.1 we get the prime ideals

$$P_{3,1} = (x^2 + x - 1) \quad \text{and} \quad P_{3,2} = (-x^3 - x^2 - 1).$$

This gives the following generators for the S-unit lattice $\mathcal{L}_S$

$$\begin{aligned}
\operatorname{Log}(u_1) &= (\log|u_1|_1, \log|u_1|_3, \log|u_1|_5, \log|u_1|_{P_{3,1}}, \log|u_1|_{P_{3,2}}) \\
&= (1.763, -1.763, 0, 0), \\
\operatorname{Log}(P_{3,1}) &= (\log|P_{3,1}|_1, \log|P_{3,1}|_3, \log|P_{3,1}|_5, \log|P_{3,1}|_{P_{3,1}}, \log|P_{3,1}|_{P_{3,2}}) \\
&= (1.099, 1.099, -2.197, 0), \\
\operatorname{Log}(P_{3,2}) &= (\log|P_{3,2}|_1, \log|P_{3,2}|_3, \log|P_{3,2}|_5, \log|P_{3,2}|_{P_{3,1}}, \log|P_{3,2}|_{P_{3,2}}) \\
&= (1.099, 1.099, 0, -2.197),
\end{aligned}$$

with the corresponding matrix

$$M_U = \begin{pmatrix} 1.763 & -1.763 & 0 & 0 \\ 1.099 & 1.099 & -2.197 & 0 \\ 1.099 & 1.099 & 0 & -2.197 \end{pmatrix}.$$

3.2.1 Finding S-units

The last part needed to perform an S-unit attack are the generators for the prime ideals in S. This is one of the main concepts that differentiate S-unit attacks from other reduction algorithms. By applying the mathematical theory from Section 3.1, we present a method for finding elements to use in the attack. This grants S-unit attacks the potential of being more effective and precise when reducing elements. Recall that

$$S = \{\infty\} \cup \{P \in R \mid \mathcal{N}(P) \leq y\}$$

for a chosen parameter y. By Theorem 3.8 we find the non-principal prime ideals by factoring the minimal polynomial of K. By Section 3.1.5 and 3.1.6 we find a generator for ideals in R, where the prime ideal factorization contains the same prime ideals as in S. In this section we provide a concrete method for finding a generator for the prime ideals in S. These are also the generators for the S-unit group. We end the section by giving two detailed examples.

We use the following concepts to find the generators for the S-unit group:

1. Cyclotomic units (Units),
2. Jacobi sums (S-units: prime ideal factorizations),
3. Generators of $P_c P_{-c}$ (S-units: principal ideals),
4. Square roots (S-units: prime ideals).

First we define the cyclotomic units as described in Section 3.1.3, which are generated by $u_0 = \zeta_m$ and $u_c = \{1 + \zeta_m^c + \zeta_m^{-c}\}$ for odd $c < n$. Then we define the set $S = \{\infty\} \cup \{P \subset R \mid \mathcal{N}(P) \leq y\}$ for a chosen parameter y. Let $p \in 1 + 2n\mathbb{Z}$ be a prime number less than or equal y. We want to find the prime ideals $P \in R$ such that $P \cap \mathbb{Z} = p$. We start by factorizing the minimal polynomial modulo p,

$$x^n + 1 = \prod_i \overline{g_i}(x)^{e_i} \pmod{p},$$

where $g_i(x) \in R$ and reduces to $\overline{g_i}(x)$ mod p. By Theorem 3.8, $P_i = (p, g_i(\zeta_m))$ is a prime ideal in R lying above p. Since $p \in 1 + 2n\mathbb{Z} = 1 + m\mathbb{Z}$, we have that $p \equiv 1 \pmod{m}$ and by Theorem 3.7 P_i splits completely. Thus, we can write the prime ideals on the form $P_i = (p, a - \zeta_m^i)$, where $a \pmod{p}$ is of order m. There could be many choices for the integer a, but we choose the the smallest a such that $a^n + 1 \equiv 0 \pmod{p}$.

These prime ideals are not principal, but with the help of Jacobi sums we can find a generator for them. See Appendix B.1 for the calculations done in Sage. Recall that for a given prime number p we have,

$$|J(\chi)|^2 = p \quad \text{and} \quad J(\chi_1, \chi_2) = J(\chi^i, \chi) = \textstyle\sum_{a \in \mathbb{F}_{p^*-1}} \chi^i(a)\chi(1-a).$$

Let $P_c = (p, a - \zeta_m^c) = (p, a - x^c)$ and $P_{-c} = (p, a - \zeta_m^{-c}) = (p, a - x^{-c}) = (p, a + x^{n-c})$ for odd $c < n$. Then,

$$P_c P_{-c} = (p^2, p(a - x^c), p(a - x^{-c}), a^2 - a(x^c + x^{-c}) + 1),$$

is an ideal in the maximal real subfield R^+, and by Section 3.1.2 it is a principal ideal. For the remaining part we use the following steps:

1. For $i = 1, ..., n$, find the Jacobi sums $J(\chi^i, \chi)$ for a character χ of order p, with the corresponding prime ideal factorization, as described in Section 3.1.6.

2. Look at the prime ideal factorization of the Jacobi sums and choose two Jacobi sums where the factorization differ by P_c and P_{-c} for only one c.

3. Divide the chosen Jacobi sums such that we end up with the fractional ideal P_c/P_{-c} and a corresponding polynomial.

4. Multiply this polynomial with the polynomial g_c, corresponding to the generator for the principal ideal $P_c P_{-c}$. This gives a new polynomial corresponding to the square of the prime ideal P_c.

5. Find a unit $u \in C_K$ to multiply the polynomial with, such that the square root of the polynomial is still an element in R. This element will be a generator for the prime ideal P_c.

Remark 3.38. When we have found a generator for P_c we automatically find a generator for P_{-c} by taking the complex conjugate, $\sigma_{-1}(P_c) = P_{-c}$.

This creates the S-unit group, namely the generators for the group of cyclotomic units and the generators for the prime ideals in S. If we want to expand S, by adding more prime ideals, we repeat the same procedure as above. We round of this section with two examples, showcasing how this can be constructed in the cyclotomic rings of degree $n = 4$ and $n = 8$.

3.2.2 First Example

In this example we demonstrate how to calculate the generators for the prime ideals in S. First directly by using Sage, then by following the approach in Section 3.2.1. At the end we compare the two results. See Appendix B.1 for the calculations done by Sage.

Let $n = 4$ and $m = 2n = 8$ such that the cyclotomic ring is $R = \mathbb{Z}[x]/(x^4 + 1)$. For $S = \{\infty\}$, we have the cyclotomic units

$$\{1 + x + x^{-1}, 1 + x^3 + x^{-3}\},$$

and from Theorem 3.12 the group of cyclotomic units C_K is generated by $u_0 = x$ and $u_1 = 1 + x + x^{-1}$. To expand S we add prime ideals from R, containing a prime number on the form $p \in 1 + 2n\mathbb{Z}$. For this case, the first prime number is $p = 1 + 8 \cdot 2 = 17$. Hence, we want to find the prime ideals in R containing $p = 17$.

We compute the prime ideal factorization directly using Sage,

$$17 = (-x^3 - 2x^2)(-2x + 1)(-x^3 + 2)(x^3 - 2x^2).$$

Each factor is a prime ideal and they are all principal. Now, assume that we do not have these prime ideals and we want to find them by using the approach described in Section 3.2.1. We start by factoring the minimal polynomial of R modulo 17. The minimal polynomial can be written as

$$\begin{aligned} x^4 + 1 &= \prod_{\text{odd } c < m} (x - \zeta_8^c) = (x - \zeta_8)(x - \zeta_8^3)(x - \zeta_8^5)(x - \zeta_8^7) \\ &= (x - \zeta_8)(x - \zeta_8^3)(x - \zeta_8^{-3})(x - \zeta_8^{-1}) \\ &= (x - \zeta_8)(x - \zeta_8^3)(x + \zeta_8)(x + \zeta_8^3). \end{aligned}$$

Let $x = 2$ be the smallest integer such that,

$$2^4 + 1 = 17 = (2 - \zeta_8)(2 - \zeta_8^3)(2 + \zeta_8)(2 + \zeta_8^3) \equiv 0 \pmod{17}.$$

Pulled back to $R = \mathbb{Z}[x]/(x^4 + 1)$, i.e. set $x = \zeta_8$, we have the factorization

$$17 = (2 - x)(2 - x^3)(2 + x)(2 + x^3). \tag{3.1}$$

Then by Theorem 3.8 the prime ideals in R containing $p = 17$ are

$$\begin{aligned} P_1 &= (17, 2 - x) \quad & P_{-1} &= (17, 2 + x^3) \\ P_3 &= (17, 2 - x^3) \quad & P_{-3} &= (17, 2 + x). \end{aligned}$$

We have found four non-principal prime ideals in R containing $p = 17$. To continue, we use Sage to find generators for the ideals P_1P_{-1} and P_3P_{-3}. We get that

$$P_1P_{-1} = (17^2, 17(2 - x), 17(2 + x^3), (2 - x)(2 + x^3))$$

and

$$P_3P_{-3} = (17^2, 17(2 - x^3), 17(2 + x), (2 - x^3)(2 + x))$$

corresponds to the polynomials

$$g_1 = 3x^3 - 3x - 1 \quad \text{and} \quad g_3 = -3x^2 + x - 3$$

as generators. To calculate the Jacobi sums we define a character forming a group of order 17. Let $\chi(3) = \zeta_8$, then for $p = 17$ we get the following Jacobi sum

$$\begin{aligned} J_1 = J(\chi, \chi) = &\sum_{a \in \mathbb{F}_{17}^* - 1} \chi(a)\chi(1-a) = \tau(\chi)^2/\tau(\chi^2) \\ &= \chi(2)\chi(1-2) + \chi(3)\chi(1-3) + ... + \chi(16)\chi(1-16) \\ &= x^{14} \cdot x^8 + x \cdot x^6 + x^{12} \cdot x^9 + x^5 \cdot x^4 + x^{15} \cdot x^{13} + x^{11} \cdot x^7 + x^{10} \cdot x^3 + x^2 \cdot x^2 \\ &+ x^3 \cdot x^{10} + x^7 \cdot x^{11} + x^{13} \cdot x^{15} + x^4 \cdot x^5 + x^9 \cdot x^{12} + x^6 \cdot x + x^8 \cdot x^{14} \\ &= -2x^3 - 2x - 3 \in R. \end{aligned}$$

By using Sage this factors into the prime ideals $P_{-3}P_{-1}$. The next Jacobi sum is

$$\begin{aligned} J_2 = J(\chi^2, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^2\chi(1-a) = \tau(\chi^2)\tau(\chi)/\tau(\chi^3) \\ &= \chi(2)^2\chi(1-2) + \chi(3)^2\chi(1-3) + ... + \chi(16)^2\chi(1-16) \\ &= x^{28} \cdot x^8 + x^2 \cdot x^6 + x^{24} \cdot x^9 + x^{10} \cdot x^4 + x^{30} \cdot x^{13} + x^{22} \cdot x^7 + x^{20} \cdot x^3 + x^4 \cdot x^2 \\ &+ x^6 \cdot x^{10} + x^{14} \cdot x^{11} + x^{26} \cdot x^{15} + x^8 \cdot x^5 + x^{18} \cdot x^{12} + x^{12} \cdot x + x^{16} \cdot x^{14} \\ &= 1 - 4x^2 \in R. \end{aligned}$$

Again, by using Sage this factors into P_3P_{-1}. The remaining Jacobi sums, with the corresponding prime ideal factorization, are as follows

$$\begin{aligned} J_3 = J(\chi^3, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^3\chi(1-a) = \tau(\chi)^3 \cdot \tau(\chi)/\tau(\chi^4) \\ &= 2x^3 + 2x + 3, \end{aligned}$$

corresponding to the factorization $P_{-3}P_{-1}$.

$$\begin{aligned} J_4 = J(\chi^4, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^4\chi(1-a) = \tau(\chi)^4\tau(\chi)/\tau(\chi^5) \\ &= 2x^3 + 2x + 3, \end{aligned}$$

corresponding to the factorization $P_{-3}P_{-1}$.

$$\begin{aligned} J_5 = J(\chi^5, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^5\chi(1-a) = \tau(\chi)^5\tau(\chi)/\tau(\chi^6) \\ &= -4x^2 + 1, \end{aligned}$$

corresponding to the factorization P_3P_{-1}.

$$\begin{aligned} J_6 = J(\chi^6, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^6\chi(1-a) = \tau(\chi)^6\tau(\chi)/\tau(\chi^7) \\ &= -2x^3 - 2x - 3, \end{aligned}$$

corresponding to the factorization $P_{-3}P_{-1}$.

$$\begin{aligned} J_7 = J(\chi^7, \chi) = &\sum_{a \in \mathbb{F}_{17^*} - 1} \chi(a)^7\chi(1-a) = \tau(\chi)^7\tau(\chi)/\tau(\chi^8) \\ &= -1 \Rightarrow (1). \end{aligned}$$

After the Jacobi sums J_1 and J_2, we do not get any new prime ideals in the factorization, so for further calculations we only need these two. The prime ideal factorization of the ideals, that J_1 and J_2 generates, differ with the prime ideals P_3 and P_{-3}. So, by dividing these Jacobi sums we get the following polynomial

$$J_2/J_1 = -6/17x^3 + 12/17x^2 + 10/17x - 3/17$$

which corresponds to the prime ideals P_3/P_{-3}. Then, by multiplying with the generator g_3 of P_3P_{-3}, we get the polynomial

$$\begin{aligned} g_3 \cdot J_2/J_1 &= (-3x^2 + x - 3)(-6/17x^3 + 12/17x^2 + 10/17x - 3/17) \\ &= -x^2 - 3x + 3, \end{aligned}$$

corresponding to P_3^2. Now, find a unit $u \in C_K$ to multiply the polynomial with, such that if we take the square root, we get an element in R. We find that the unit $u = u_0u_1$ gives us the polynomial

$$\begin{aligned} u_0u_1 \cdot g_3 \cdot J_2/J_1 &= x(1 + x + x^{-1})(-x^2 - 3x + 3) \\ &= -4x^3 - x^2 + 4 = x^6 - 4x^3 + 4 \\ &= (x^3 - 2)(x^3 - 2) = (x^3 - 2)^2, \end{aligned}$$

and by taking the square root we have a generator

$$\sqrt{u_0u_1 \cdot g_3 \cdot J_2/J_1} = x^3 - 2$$

for P_3. We see that $P_3 = (x^3 - 2)$ is the same, up to units, as one of the factors in the factorization done in Equation 3.1. To find the generator of P_{-3} we take the complex conjugate

$$\sigma_{-1}(x^3 - 2) = x^{-3} - 2 = -x - 2,$$

and $P_{-3} = (x + 2)$. This is also a factor in Equation 3.1.

Since the Jacobi sums only generated ideals with the prime ideal P_{-1}, and not P_1, the approach for finding a generator for P_1 is slightly different. By looking at the prime ideal factorization of the ideals we already have, we combine their generators as follows,

$$\frac{g_1^2 \cdot g_3}{J_1 \cdot J_2} = 3x^2 + 3x - 1$$

which corresponds to the prime ideal factorization

$$\frac{P_1^2 P_{-1}^2 P_3 P_{-3}}{P_3 P_{-3} P_{-1}^2} = P_1^2.$$

Then we find a unit $u \in C_K$ to multiply the polynomial with,

$$\begin{aligned} u_0u_3 \cdot \frac{g_1^2 \cdot g_3}{J_1 \cdot J_2} &= x^2 - 4x + 4 = (x - 2)(x - 2) = (x - 2)^2 \\ \Rightarrow \sqrt{u_0u_3(3x^2 + 3x - 1)} &= x - 2, \end{aligned}$$

and we have found the prime ideal $P_1 = (x - 2)$. Lastly, we have the following generator for P_{-1},

$$\sigma_{-1}(x - 2) = x^{-1} - 2 = -x^3 - 2.$$

Again, these are also the same up to units as the factors in Equation 3.1. We can now compare the generators we have found with the generators from Sage,

$$\begin{aligned} P_1 &= (x-2) = -x^2(x^3-2x^2) \\ P_{-1} &= (x^3+2) = x^3(-2x+1) \\ P_3 &= (x^3-2) = -(-x^3+2) \\ P_{-3} &= (x+1) = x^2(-x^3-2x^2). \end{aligned}$$

In conclusion, the approach described in Section 3.2.1 gives the same generators as Sage. Further, the set S consists of the elements $\{u_1, u_3, P_1, P_{-1}, P_3, P_{-3}\}$ corresponding to the infinite and finite norms, and the S-unit group is generated by the elements $\{u_0, u_1, P_1, P_{-1}, P_3, P_{-3}\}$.

3.2.3 Second Example

This example demonstrates how much more complicated it gets to find generators in a cyclotomic ring by solely increasing the dimension of the ring by 2. See Appendix B.1 for the calculations done in Sage.

Let $n = 8$ and $m = 2n = 16$ such that we have the cyclotomic ring $R = \mathbb{Z}[x]/(x^8+1)$. The cyclotomic units in S are

$$\{1+x+x^{-1}, 1+x^3+x^{-3}, 1+x^5+x^{-5}, 1+x^7+x^{-7}\},$$

corresponding to the infinite norms. By Theorem 3.12, the unit group C_K is generated by the cyclotomic units

$$\{x, 1+x+x^{-1}, 1+x^3+x^{-3}, 1+x^5+x^{-5}\}.$$

Next, we have the prime element $p = 1 + 2 \cdot 8 = 17$ and we want to find the prime ideals in R containing $p = 17$. From Sage, we have the factorization

$$17 = (17, x-3)(17, x+3)(17, x-5)(17, x+5)(17, x-6)(17, x+6)(17, x-7)(17, x+7),$$

where none of the factors are principal. By following the approach in Section 3.2.1 we attempt to find the generators. The minimal polynomial of R factors as

$$\begin{aligned} x^8+1 &= \prod_{\text{c odd}} (x-\zeta_{16}^c) \\ &= (x-\zeta_{16})(x-\zeta_{16}^3)(x-\zeta_{16}^5)(x-\zeta_{16}^7)(x-\zeta_{16}^{-7})(x-\zeta_{16}^{-5})(x-\zeta_{16}^{-3})(x-\zeta_{16}^{-1}) \\ &= (x-\zeta_{16})(x-\zeta_{16}^3)(x-\zeta_{16}^5)(x-\zeta_{16}^7)(x+\zeta_{16})(x+\zeta_{16}^3)(x+\zeta_{16}^5)(x+\zeta_{16}^7). \end{aligned}$$

Let $x = 3$ be the smallest integer such that

$$\begin{aligned} 3^8+1 &= 2 \cdot 17 \cdot 193 \\ &= (3-\zeta_{16})(3-\zeta_{16}^3)(3-\zeta_{16}^5)(3-\zeta_{16}^7)(3-\zeta_{16}^{-7})(3-\zeta_{16}^{-5})(3-\zeta_{16}^{-3})(3-\zeta_{16}^{-1}) \\ &\equiv 0 \pmod{17}. \end{aligned}$$

Set $\zeta_{16} = x$ and we have the factorization

$$2 \cdot 17 \cdot 193 = (3-x)(3-x^3)(3-x^5)(3-x^7)(3+x)(3+x^3)(3+x^5)(3+x^7).$$

The factorization from Sage yields,

$$\begin{aligned}
(3-x) &= (2,x+1)(17,x-3)(193,x-3)\\
(3-x^3) &= (2,x+1)(17,x-7)(193,x+27)\\
(3-x^5) &= (2,x+1)(17,x+5)(193,x+50)\\
(3-x^7) &= (2,x+1)(17,x+6)(193,x-64)\\
(3+x) &= (2,x+1)(17,x+3)(193,x+3)\\
(3+x^3) &= (2,x+1)(17,x+7)(193,x-27)\\
(3+x^5) &= (2,x+1)(17,x-5)(193,x-50)\\
(3+x^7) &= (2,x+1)(17,x-6)(193,x+64).
\end{aligned}$$

Now, define the prime ideals in R that contains $p = 17$ as

$$\begin{aligned}
P_1 &= (17, 3-x) = (17, x-3) & P_{-1} &= (17, 3+x^7) = (17, x-6)\\
P_3 &= (17, 3-x^3) = (17, x-7) & P_{-3} &= (17, 3+x^5) = (17, x-5)\\
P_5 &= (17, 3-x^5) = (17, x+5) & P_{-5} &= (17, 3+x^3) = (17, x+7)\\
P_7 &= (17, 3-x^7) = (17, x+6) & P_{-7} &= (17, 3+x) = (17, x+3).
\end{aligned}$$

We have eight non-principal prime ideals we need to find generators for. By using Sage, the principal ideals

$$\begin{aligned}
P_7P_{-7} &= (17^2, 17(3-x^7), 17(3+x), (3-x^7)(3+x))\\
P_5P_{-5} &= (17^2, 17(3-x^5), 17(3+x^3), (3-x^5)(3+x^3))\\
P_3P_{-3} &= (17^2, 17(3-x^3), 17(3+x^5), (3-x^3)(3+x^5))\\
P_1P_{-1} &= (17^2, 17(3-x), 17(3+x^7), (3-x)(3+x^7))
\end{aligned}$$

have the following generators

$$\begin{aligned}
g_7 &= (x^7 - x^5 + x^4 + x^2 + 1)\\
g_5 &= (-x^7 - x^6 + x^5 - x + 1)\\
g_3 &= (x^7 + x^6 - x^2 - x + 1)\\
g_1 &= (-x^6 - x^4 - x^3 + x + 1).
\end{aligned}$$

Next, with help from Sage, we find the following Jacobi sums with the corresponding prime ideal factorization

$$\begin{aligned}
J_1 &= 2x^7 + 2x^6 - x^4 + 2x^2 - 2x & P_7P_{-5}P_{-3}P_1\\
J_2 &= x^7 - 2x^6 - 3x^5 + x^4 - x^3 - x & P_{-7}P_{-5}P_{-3}P_1\\
J_3 &= x^7 + 2x^6 - x^5 + 3x^3 + x - 1 & P_7P_{-5}P_3P_1\\
J_4 &= x^7 + x^5 + x^3 - 2x^2 - 3x + 1 & P_{-7}P_{-5}P_{-3}P_1\\
J_5 &= -x^7 - 2x^6 + x^5 - x^4 - x^3 - 3x & P_7P_5P_{-3}P_1\\
J_6 &= -2x^6 - 3x^4 + 2x^2 & P_{-7}P_{-5}P_3P_1\\
J_7 &= 2x^6 - 2x^5 - 2x^3 - 2x^2 + 1 & P_7P_{-5}P_{-3}P_1.
\end{aligned}$$

The Jacobi sums after J_7 do not give any new factorization, so we only need the ones listed above for finding the square roots. Let the polynomial for the Jacobi sum be on the left and the corresponding prime ideal factorization on the right. First, we find the generators for P_7 and P_{-7}

$$\begin{aligned}
J_1/J_2 &= \tfrac{1}{17}(-3x^7 - 8x^6 + 7x^5 - 4x^4 - 5x^3 - 2x^2 - 11x - 1) & P_7/P_{-7}\\
g_7 \cdot J_1/J_2 &= -x^4 - x^3 + x^2 - x + 1 & P_7^2\\
(u_5 \cdot g_7 \cdot J_1/J_2)^{1/2} &= x^6 - x^2 + x & P_7
\end{aligned}$$

and for P_{-7} the complex conjugate gives the generator

$$\sigma_{-1}(x^6 - x^2 + x) = x^{-6} - x^{-2} + x^{-1} = -x^2 + x^4 - x^7 = -x^2(x^5 - x^4 + 1).$$

Next, we find the generators for P_5 and P_{-5}.

$$\begin{array}{lr}
J_5/J_1 = \frac{1}{17}(7x^7 + 2x^6 + 3x^5 - 4x^4 + 11x^3 + 8x^2 - 5x + 1) & P_5/P_{-5} \\
g_5 \cdot J_5/J_1 = x^7 + x^5 - x^4 + x^2 + x & P_5^2 \\
(u_0 \cdot u_1 \cdot g_5 \cdot J_1/J_5)^{1/2} = x^7 - x^4 - x^3 & P_5
\end{array}$$

and the complex conjugate gives the generator for P_{-5},

$$\sigma_{-1}(x^7 - x^4 - x^3) = x^{-7} - x^{-4} - x^{-3} = -x + x^4 + x^5.$$

The last generators we can find using this method is for P_3 and P_{-3}. For P_3 we have

$$\begin{array}{lr}
J_3/J_1 = \frac{1}{17}(-3x^7 + 2x^6 - 7x^5 - x^4 - 5x^3 - 8x^2 + 11x + 4) & P_3/P_{-3} \\
g_3 \cdot J_3/J_1 = x^7 + x^6 + x^4 - x^2 + x & P_3^2 \\
(u_7 \cdot g_3 \cdot J_3/J_1)^{1/2} = x^7 + x^3 - x^2 & P_3
\end{array}$$

and for P_{-3} we have

$$\sigma_{-1}(x^7 + x^3 - x^2) = x^{-7} + x^{-3} - x^{-2} = -x - x^5 + x^6.$$

The remaining prime ideals are P_1 and P_{-1}. The ideals generated by the Jacobi sums only contains P_1, hence we find the generators for P_1 and P_{-1} by looking at the prime ideal factorization of the ideals we already have. For P_{-1} have the following,

$$\begin{array}{lr}
(g_1^2 \cdot g_3 \cdot g_5 \cdot g_7)/(J_5 \cdot J_6) & \frac{P_1^2 P_{-1}^2 P_3 P_{-3} P_5 P_{-5} P_7 P_{-7}}{P_1^2 P_3 P_{-3} P_5 P_{-5} P_7 P_{-7}} = P_{-1}^2 \\
(u_0 \cdot u_3 \cdot \frac{g_1^2 \cdot g_3 \cdot g_5 \cdot g_7}{J_5 \cdot J_6})^{1/2} = x^7 + x^4 - 1 & P_{-1}
\end{array}$$

and for P_1 we take the complex conjugate,

$$P_1 = \sigma_{-1}(P_{-1}) = \sigma_{-1}((x^7 + x^4 - 1)) = (x^{-7} + x^{-4} - 1) = (-x - x^4 - 1).$$

After reducing the generators by units, we have that the set S contains the elements

$$\begin{array}{lll}
u_1 = 1 + x + x^{-1} & P_1 = (x^4 + x + 1) & P_{-1} = (x^7 + x^4 - 1) \\
u_3 = 1 + x^3 + x^{-3} & P_3 = (x^5 + x - 1) & P_{-3} = (x^5 - x^4 - 1) \\
u_5 = 1 + x^5 + x^{-5} & P_5 = (x^5 - x - 1) & P_{-5} = (x^4 + x^3 - 1) \\
u_7 = 1 + x^7 + x^{-7} & P_7 = (x^5 - x + 1) & P_{-7} = (x^4 - x + 1),
\end{array}$$

representing the infinite and finite norm. The S-unit group is generated by the elements

$$\begin{array}{lll}
u_0 = x & P_1 = (x^4 + x + 1) & P_{-1} = (x^7 + x^4 - 1) \\
u_1 = 1 + x + x^{-1} & P_3 = (x^5 + x - 1) & P_{-3} = (x^5 - x^4 - 1) \\
u_3 = 1 + x^3 + x^{-3} & P_5 = (x^5 - x - 1) & P_{-5} = (x^4 + x^3 - 1) \\
u_5 = 1 + x^5 + x^{-5} & P_7 = (x^5 - x + 1) & P_{-7} = (x^4 - x + 1).
\end{array}$$

Chapter 4

S-unit Attacks in Cryptography

From the mathematical theory of cyclotomic fields and S-units, we describe the algorithms that attempts to solve the hard mathematical problems in lattice-based cryptography. Recall from Section 2.3 that we are interested in solving the following problems for a lattice $\mathcal{L} \subset \mathbb{R}^n$:

- **The Approximate Shortest Vector Problem** (apprSVP$_\gamma$): Find a nonzero vector $v \in \mathcal{L}$ satisfying $\|v\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.
- **The Approximate Closest Vector Problem** (apprCVP$_\gamma$): For a given vector $w \in \mathbb{R}^n$, find a nonzero vector $v \in \mathcal{L}$ such that $\|v - w\| \leq \gamma \cdot \operatorname{dist}(w, \mathcal{L})$.

We start by presenting a step by step procedure for a unit attack. Then we do the same for an S-unit attack, along with some commentary to the most essential aspects to the algorithm.

4.1 Unit Attacks

For unit attacks, we have $S = \{\infty\}$. This means we are only working with the group of cyclotomic units C_K in the cyclotomic ring R. The goal of this attack is to find a short nonzero generator of an known ideal $I \subset R$.

We start with a high level description of the algorithm. Consider an ideal $I \subset R$ with a generator $\alpha \in R$ such that $(\alpha) = I$. We want to find a unit $u \in C_K$ which outputs α/u where $\|\alpha/u\| < \|\alpha\|$. We can do this because multiplying a generator by units still gives an element that generates the same ideal. The algorithm can be summarized with the following steps.

Unit attack:

1. Compute the $r - 1$ generators for the group of cyclotomic units C_K, as described in Section 3.1.3.
2. Define the set S with $r = r_1 + r_2$ elements corresponding to the infinite norms, such that Theorem 3.34 holds, and define the Log-embedding, $\operatorname{Log}(a) = (\log|a|_1, ..., \log|a|_r)$.
3. Compute the unit lattice $\mathcal{L}_U = \operatorname{Log}(U_K) = (\operatorname{Log}(u_1), ..., \operatorname{Log}(u_{r-1}))$ and its corresponding matrix, $M_U = (\operatorname{Log}(u_{i,j}))_{i,j}$, with $\operatorname{Log}(u_i) = (\log|u_i|_1, ..., \log|u_i|_r)$ as rows.

4. Embed the known generator α of the ideal I as a vector $y = \text{Log}(\alpha)$.

5. Use Algorithm 7 to find a vector $t \in \mathbb{R}^r$, such that $y' = \sum_{i=1}^{r} t_i \text{Log}(u_i)$, and $\|y - y'\|$ is minimized.

6. Pull y' back to R corresponding to a unit $u = \prod_i u_i^{t_i} \in C_K$. Divide by this unit such that $\alpha' = \alpha/u$ is the new candidate for the generator for the ideal I.

7. Check that $\|\alpha'\| < \|\alpha\|$. If so, replace α with α' and repeat the process until $\|\alpha'\| \geq \|\alpha\|$.

Remark 4.1. In Step 3, we have a over-determined system of r equations, but only $r - 1$ unknown elements. This is because of the requirement form Theorem 3.34. The coordinates must sum to 0 and the unit lattice is a hyperplane in $\mathbb{R}^r$. Consequently, we use the unit normal vector to the hyperplane $y_0 = (1, 1, ..., 1)$ and parameterize y as $y' = y + \lambda y_0$. Then λ is the constant needed for the sum of the coordinates of y to be zero. This gives us the following system of equations,

$$\begin{aligned} y + \lambda y_0 &= t \cdot M \\ y &= t \cdot M - \lambda y_0 \\ y &= (t_1, ..., t_{r-1}, -\lambda) \cdot \begin{pmatrix} \text{Log}(u_1) \\ \vdots \\ \text{Log}(u_{r-1}) \\ y_0 \end{pmatrix}. \end{aligned}$$

This is now a system with r equations and r unknown elements. For later, when we write $t \cdot M_U = y$, it is implied that we have already done this parameterization.

4.2 S-unit Attack

We now describe the main steps for an S-unit attack. The approach is similar to unit attacks, but we expand S by including prime ideals $P \subset R$. The goal of the attack is to find a smaller element in a known ideal $I \subset R$.

We have the following high-level description of the algorithm. For a chosen parameter y, we define the set $S = \{\infty\} \cup \{P \mid \mathcal{N}(P) \leq y\}$ and the S-unit group U_S. The choice of y depends on the size of the element we want to reduce, see Remark 4.2. For a known ideal $I \subseteq R$ as input, choose an element $\alpha \in I$, and find an S-unit $u \in U_S$ such that $\|\alpha/u\| < \|\alpha\|$.

S-unit attack:

1. Compute the group of cyclotomic units C_K, with $r - 1$ elements.

2. For an element $\alpha \in I$, compute the size $\|\alpha\|$ and choose the parameter y. Find all the prime ideals $P \subset R$, with norm $\mathcal{N}(P) \leq y$, and add them to S, such that $S = \{\infty\} \cup \{P \subset R \mid \mathcal{N}(P) \leq y\}$.

3. Define the Log-embedding, $\text{Log}(a) = (\log|a|_1, ..., \log|a|_r, \log|a|_{P_1}, ..., \log|a|_{P_s})$, according to the elements in S, with r elements corresponding to the infinite norm and s elements corresponding to the finite norm.

4. Compute the S-unit lattice $\mathcal{L}_S = \text{Log}(U_S) = (\text{Log}(u_1), ..., \text{Log}(u_{r+s-1}))$, and the corresponding matrix M_S, with $\text{Log}(u_i)$ as rows.

5. Embed $\alpha \in I$ as a vector $y = \text{Log}(\alpha)$ and solve $t \cdot M_S = y$ by using Algorithm 7. Remark 4.1 applies here.

6. Pull $y \in \mathcal{L}_S$ back to R, corresponding to a S-unit $u \in U_S$. Divide α by this S-unit such that $\alpha' = \alpha/u$ is a new element in I.

7. Check if $\|\alpha'\| < \|\alpha\|$. If so, replace α with α'.

8. If the size of α' is still very large, increase the parameter y and try to reduce the new element with a larger set S.

9. For the prime ideal $P_2 = (x + 1) \notin S$, divide α' by $(x + 1)^k$ starting at $k = 1$ and gradually increase k. For each k, check if the size get smaller and if the element is in the ideal I.

10. Sometimes it is necessary to preform an additional unit attack to get an even smaller element.

Remark 4.2. The choice of the parameter y depends on the size of α, the element we want to reduce. We start by choosing a small y compared to the size of α, check if we can reduce the element, and slowly increase the value. However, y should always be smaller than the size of the element we want to reduce.

Remark 4.3. An important difference for S-unit attacks is that we might divide by prime ideals from the generator of the ideal, such that the new element is no longer in the ideal. Therefore, we always have to check if $\alpha/u \in I$. If $v_P(\alpha/u) < v_P(I)$ for some P, then $\alpha/u \notin I$ and we need to multiply α/u with a generator of the prime ideal P, until $v_P(\alpha/u) \geq v_P(I)$ and $\alpha/u \in I$. Otherwise, $v_P(\alpha/u) \geq v_P(I)$ for all the prime ideals $P \in S$ and $\alpha/u \in I$.

If we know the algebraic norm of the ideal, we can choose the parameter y accordingly. By letting y be strictly smaller than the norm of I, the S-unit group do not contain any of the same prime ideals as the generator of I, and we cannot divide out by the generator of I.

In conclusion, Remark 4.2 and 4.3 are the most interesting aspects when analyzing S-unit attacks. These are the elements of the algorithm that yields the potential of it being more effective and precise than other reduction algorithms, such as the LLL algorithm from Section 2.4. Because it consider the cyclotomic ring directly, rather than looking at the problem as a lattice, it has the potential of providing more information of the cryptosystem. As a result we can choose which elements to reduce more adequately. Although, the mathematical theory indicates this to be the case, it is still uncertain how well the actual performance will be. In Chapter 6, we provide different aspects that could be interesting to study further. In the next chapter, we consider examples of both unit and S-unit attacks, where these elements are taken into consideration.

Chapter 5

Examples of S-unit Attacks

We now compile the theoretic foundations from Chapter 3 and 4 to construct examples of different attacks against lattice-based cryptography. We start by applying unit and S-unit attacks for general ideals. Then we provide an example with one possible method of applying S-units for break the lattice-based cryptosystem, NTRU.

For the remainder of this chapter, let $m = 2n$ and $n = 2^k$ for a positive integer k. Let $R = \mathbb{Z}[x]/(x^n + 1)$ be the cyclotomic ring with the m'th root of unity $x = \zeta_m = e^{2\pi i/m}$. Let $S = \{\infty\} \cup \{P \subset R \,|\, \mathcal{N}(P) \leq y\}$ for a chosen parameter y, corresponding to the infinite and finite norms, and let $U_S = \{u \subset R^* \mid |u|_P = 1 \, \forall P \notin S\}$ be the S-unit group.

5.1 Applying Unit Attacks to a General Ideal

For preparation to the more complex examples of S-unit attacks, we present an example of a unit attack for a general ideal in a cyclotomic ring.

5.1.1 Unit Attack for $n = 8$

Let $R = \mathbb{Z}[x]/(x^8 + 1)$ be the cyclotomic ring. We start by finding the cyclotomic units as described in Section 3.16. The set S consists of the cyclotomic units

$$\begin{aligned} u_1 &= 1 + x + x^{-1} \\ u_3 &= 1 + x^3 + x^{-3} \\ u_5 &= 1 + x^5 + x^{-5} \\ u_7 &= 1 + x^7 + x^{-7}, \end{aligned}$$

where $u_7 = 1/(u_1 u_3 u_5) = 1 + x^7 + x^{-7}$ and $u_1 u_3 u_5 u_7 = -1$, corresponding to the infinite norms. The unit group C_K is generated by the cyclotomic units

$$\begin{aligned} u_0 &= x \\ u_1 &= 1 + x + x^{-1} \\ u_3 &= 1 + x^3 + x^{-3} \\ u_5 &= 1 + x^5 + x^{-5}. \end{aligned}$$

From S we define the following Log-embedding

$$\text{Log}(a) = (\log|a|_1, \log|a|_3, \log|a|_5), \log|a|_7),$$

where the numbers $1, 3, 5$ and 7 represent the infinite norms in the ring R. Compute the unit lattice $\mathcal{L}_U = \mathrm{Log}(C_K)$ which is generated by the vectors

$$\begin{aligned} v_1 &= \mathrm{Log}(u_1) = (2.093, 1.137, -2.899, -0.330) \\ v_2 &= \mathrm{Log}(u_3) = (1.137, -0.330, 2.093, -2.899) \\ v_3 &= \mathrm{Log}(u_5) = (-2.899, 2.093, -0.330, 1.137). \end{aligned}$$

By adding the unit vector $(1, 1, 1, 1)$ we have the corresponding matrix,

$$M_U = \begin{pmatrix} 2.093 & 1.137 & -2.899 & -0.330 \\ 1.137 & -0.330 & 2.093 & -2.899 \\ -2.899 & 2.093 & -0.330 & 1.137 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Now, take a random element from R

$$\alpha = -11x^7 - 3x^6 + 3x^5 + 3x^4 - 2x^3 - 6x^2 + x + 7,$$

with the size $\|\alpha\| = 15.43$ and define the ideal $I = (\alpha)$. We want to find a shorter generator for I by dividing with units from C_K. To find such a unit, we compute the vector $y = \mathrm{Log}(\alpha) = (5.467, 6.546, -2.597, 2.926)$, using the same Log-embedding. Then, to find an element $u \in C_K$ close to α, we solve the linear system $t \cdot M_U = y$. We get

$$(t_1, t_2, t_3, \lambda) \cdot \begin{pmatrix} 2.093 & 1.137 & -2.899 & -0.330 \\ 1.137 & -0.330 & 2.093 & -2.899 \\ -2.899 & 2.093 & -0.330 & 1.137 \\ 1 & 1 & 1 & 1 \end{pmatrix} = (5.467, 6.546, -2.597, 2.926),$$

which gives the vector $t = (1.943, 0.073, 0.610, 3.085) \approx (2, 0, 1, 3)$. We omit the last coordinate since it is the parameter for y. The closest element to y in $\mathcal{L}_U$, is the vector $w = 2 \cdot v_1 + 0 \cdot v_2 + 1 \cdot v_3 = 2v_1 + v_3$, corresponding to the unit $u = u_1^2 u_5 \in C_K$. We divide α by this unit,

$$\alpha' = \alpha/u = -2x^7 - 2x^6 - 2x^5 + x^4 - 2x^2 + 3x + 2.$$

By comparing the sizes

$$\|\alpha'\| = 5,48 < 15,43 = \|\alpha\|,$$

we see that the new element is much smaller and we let α' be the new generator for I. If we now try to repeat the same process with the new generator, we get that $t = (-0.057, 0.073, -0.390) \approx (0, 0, 0)$, and there is no further reduction. This is expected since the algorithm is supposed to find the closest unit in C_K, so there should not exist any other after we have divided by the unit. In conclusion, we have found a smaller generator for the ideal, $I = (\alpha') = (-2x^7 - 2x^6 - 2x^5 + x^4 - 2x^2 + 3x + 2)$, and we are done.

As a result, we see that only reducing with units have the potential of being very effective. In spite of the dimension being low for the cyclotomic ring, compared to an actual cryptosystem, finding the cyclotomic units does not require much work because of the properties discussed in Chapter 3. Therefore, it suggests that the complexity of computing the units should not be an obstacle for this type of attack. However, unit attacks might not be as effective when the element to reduce is far from being the element to recover, and consisting of large prime ideals. This is where S-unit attacks could be the solution.

5.2 Applying S-unit Attacks to a General Ideal

We now turn to the examples of S-unit attacks on general ideals. We start with the cyclotomic ring of degree 4, where we showcase a more detailed approach on how to apply an S-unit attack for reducing an element from an ideal. Then, for the cyclotomic ring of degree 8, we analyze different outcomes when attempting to reduce different elements from the same ideal using S-units.

5.2.1 S-unit Attack for $n = 4$

Let $R = \mathbb{Z}[x]/(x^4 + 1)$ be the cyclotomic ring. Let $I \subset R$ be the known ideal, with the unknown generator $g = -5x^2 - 3x + 5$. Our goal is to find g, or an element close to g.

We start by choosing a random element $\alpha = -5x^3 + 115x^2 + 5x \in I$, with the size $\|\alpha\| = 115.22$. There are many prime elements below 115, so by choosing the parameter for S to be small compared to the size of α, we hope to reduce α. The cyclotomic units in S are

$$u_1 = 1 + x + x^{-1} \text{ and } u_3 = 1 + x^3 + x^{-3}.$$

The unit group C_K is generated by $u_0 = x$ and $u_1 = 1 + x + x^{-1}$, since $1/u_1 = -u_3$. Next, let $y = 17$ and extend S with prime ideals in R. To find such prime ideals, we take every prime number $2 < p \leq 17$ and find the prime ideal factorization in R. If the norm of these prime ideals are less than or equal 17, we add them to S. We use Sage and the same approach as in Example 3.2.3 to find the factorization and norm. See Appendix B.5 for the computations done in Sage.

$\boldsymbol{p = 3:}$

$$\begin{array}{ll}
(3)R = (x^2 + x - 1)(-x^3 - x^2 - 1) \bmod 3 & \\
P_{3,1} = (x^2 + x - 1) & \mathcal{N}(P_{3,1}) = 9 \leq 17 \\
P_{3,2} = (-x^3 - x^2 - 1) & \mathcal{N}(P_{3,2}) = 9 \leq 17,
\end{array}$$

$\boldsymbol{p = 5:}$

$$\begin{array}{ll}
(5)R = (x^2 + 2)(x^2 - 2) \bmod 5 & \\
P_{5,1} = (x^2 + 2) & \mathcal{N}(P_{5,1}) = 25 > 17 \\
P_{5,2} = (x^2 - 2) & \mathcal{N}(P_{5,2}) = 25 > 17,
\end{array}$$

$\boldsymbol{p = 7:}$

$$\begin{array}{ll}
(7)R = (2x^2 + x + 2)(-2x^2 + x - 2) \bmod 7 & \\
P_{7,1} = (2x^2 + x + 2) & \mathcal{N}(P_{7,1}) = 49 > 17 \\
P_{7,2} = (-2x^2 + x - 2) & \mathcal{N}(P_{7,2}) = 49 > 17,
\end{array}$$

$\boldsymbol{p = 11:}$

$$\begin{array}{ll}
(11)R = (x^3 + x + 3)(x^3 - 3x^2 - x) \bmod 11 & \\
P_{11,1} = (x^3 + x + 3) & \mathcal{N}(P_{11,1}) = 121 > 17 \\
P_{11,2} = (x^3 - 3x^2 - x) & \mathcal{N}(P_{11,2}) = 121 > 17,
\end{array}$$

$\boldsymbol{p = 13:}$

$$\begin{array}{ll}
(13)R = (-3x^2 - 2)(2x^2 + 3) \bmod 13 & \\
P_{13,1} = (-3x^2 - 2) & \mathcal{N}(P_{13,1}) = 169 > 17 \\
P_{13,2} = (2x^2 + 3) & \mathcal{N}(P_{13,2}) = 169 > 17,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p = 17:} & \\
(17)R = (2-x)(2-x^3)(2+x)(2+x^3) \bmod 17 & \\
P_{17,1} = (2-x) & \mathcal{N}(P_{17,1}) = 17 \leq 17 \\
P_{17,3} = (2-x^3) & \mathcal{N}(P_{17,3}) = 17 \leq 17 \\
P_{17,-1} = (2+x^3) & \mathcal{N}(P_{17,-1}) = 17 \leq 17 \\
P_{17,-3} = (2+x) & \mathcal{N}(P_{17,-3}) = 17 \leq 17.
\end{array}$$

Hence, we extend S with the primes ideals $\{P_{3,1}, P_{3,2}, P_{17,1}, P_{17,3}, P_{17,-1}, P_{17,-3}\}$ and S contains the elements

$$\begin{array}{ll}
u_1 = 1 + x + x^{-1} & u_3 = 1 + x^3 + x^{-3} \\
P_{3,1} = x^2 + x - 1 & P_{3,2} = x^3 + x^2 + 1 \\
P_{17,3} = 2 - x^3 & P_{17,-3} = 2 + x \\
P_{17,1} = 2 - x & P_{17,-1} = 2 + x^3,
\end{array}$$

corresponding to the infinite and finite norms. This gives us the Log-embedding,

$$\begin{aligned}
\mathrm{Log}(a) = (&\log|a|_1, \log|a|_3, \log|a|_{P_{3,1}}, \log|a|_{P_{3,2}}, \\
&\log|a|_{P_{17,1}}, \log|a|_{P_{17,3}}, \log|a|_{P_{17,-1}}, \log|a|_{P_{17,-3}}).
\end{aligned}$$

Further, the S-unit group is generated by

$$\begin{array}{ll}
u_0 = x & u_1 = 1 + x + x^{-1} \\
P_{3,1} = x^2 + x - 1 & P_{3,2} = x^3 + x^2 + 1 \\
P_{17,3} = 2 - x^3 & P_{17,-3} = 2 + x \\
P_{17,1} = 2 - x & P_{17,-1} = 2 + x^3,
\end{array}$$

and by computing the Log-embedding for each generator we get the following generators for the S-unit lattice $\mathcal{L}_S = \mathrm{Log}(U_S)$,

$$\begin{aligned}
\mathrm{Log}(u_1) = (&\log|u_1|_1, \log|u_1|_3, \log|u_1|_{P_{3,1}}, \log|u_1|_{P_{3,2}}, \\
&\log|u_1|_{P_{17,1}}, \log|u_1|_{P_{17,3}}, \log|u_1|_{P_{17,-1}}, \log|u_1|_{P_{17,-3}}) \\
= (&1.763, 1.763, 0, 0, 0, 0, 0, 0),
\end{aligned}$$

$$\begin{aligned}
\mathrm{Log}(P_{3,1}) = (&\log|P_{3,1}|_1, \log|P_{3,1}|_3, \log|P_{3,1}|_{P_{3,1}}, \log|P_{3,1}|_{P_{3,2}}, \\
&\log|P_{3,1}|_{P_{17,1}}, \log|P_{3,1}|_{P_{17,3}}, \log|P_{3,1}|_{P_{17,-1}}, \log|P_{3,1}|_{P_{17,-3}}) \\
= (&1.099, 1.099, -2.197, 0, 0, 0, 0, 0),
\end{aligned}$$

$$\begin{aligned}
\mathrm{Log}(P_{3,2}) = (&\log|P_{3,2}|_1, \log|P_{3,2}|_3, \log|P_{3,2}|_{P_{3,1}}, \log|P_{3,2}|_{P_{3,2}}, \\
&\log|P_{3,2}|_{P_{17,1}}, \log|P_{3,2}|_{P_{17,3}}, \log|P_{3,2}|_{P_{17,-1}}, \log|P_{3,2}|_{P_{17,-3}}) \\
= (&1.099, 1.099, 0, -2.197, 0, 0, 0, 0),
\end{aligned}$$

$$\begin{aligned}
\mathrm{Log}(P_{17,1}) = (&\log|P_{17,1}|_1, \log|P_{17,1}|_3, \log|P_{17,1}|_{P_{3,1}}, \log|P_{17,1}|_{P_{3,2}}, \\
&\log|P_{17,1}|_{P_{17,1}}, \log|P_{17,1}|_{P_{17,3}}, \log|P_{17,1}|_{P_{17,-1}}, \log|P_{17,1}|_{P_{17,-3}}) \\
= (&0.775, 2.058, 0, 0, -2.833, 0, 0, 0),
\end{aligned}$$

$$\mathrm{Log}(P_{17,3}) = (\log|P_{17,3}|_1, \log|P_{17,3}|_3, \log|P_{17,3}|_{P_{3,1}}, \log|P_{17,3}|_{P_{3,2}},$$

$$\begin{aligned}&\log|P_{17,3}|_{P_{17,1}}, \log|P_{17,3}|_{P_{17,3}}, \log|P_{17,3}|_{P_{17,-1}}, \log|P_{17,3}|_{P_{17,-3}})\\ &= (2.058, 0.775, 0, 0, 0, -2.833, 0, 0),\end{aligned}$$

$$\begin{aligned}\mathrm{Log}(P_{17,-1}) &= (\log|P_{17,-1}|_1, \log|P_{17,-1}|_3, \log|P_{17,-1}|_{P_{3,1}}, \log|P_{17,-1}|_{P_{3,2}},\\ &\quad \log|P_{17,-1}|_{P_{17,1}}, \log|P_{17,-1}|_{P_{17,3}}, \log|P_{17,-1}|_{P_{17,-1}}, \log|P_{17,-1}|_{P_{17,-3}})\\ &= (0.775, 2.058, 0, 0, 0, 0, -2.833, 0),\end{aligned}$$

$$\begin{aligned}\mathrm{Log}(P_{17,-3}) &= (\log|P_{17,-3}|_1, \log|P_{17,-3}|_3, \log|P_{17,-3}|_{P_{3,1}}, \log|P_{17,-3}|_{P_{3,2}},\\ &\quad \log|P_{17,-3}|_{P_{17,1}}, \log|P_{17,-3}|_{P_{17,3}}, \log|P_{17,-3}|_{P_{17,-1}}, \log|P_{17,-3}|_{P_{17,-3}})\\ &= (2.058, 0.775, 0, 0, 0, 0, 0, -2.833).\end{aligned}$$

Define the generators for the S-unit lattice $\mathcal{L}_S$ as the vectors

$$\begin{array}{ll} v_1 = \mathrm{Log}(u_1) & v_5 = \mathrm{Log}(P_{17,3}) \\ v_2 = \mathrm{Log}(P_{3,1}) & v_6 = \mathrm{Log}(P_{17,-1}) \\ v_3 = \mathrm{Log}(P_{3,2}) & v_7 = \mathrm{Log}(P_{17,-3}). \\ v_4 = \mathrm{Log}(P_{17,1}) & \end{array}$$

Let these be the rows for the 8×8-matrix M_S, with the unit vector $y_0 = (1, 1, 1, 1, 1, 1, 1, 1)$ as the last row,

$$M_S = \begin{pmatrix} 1.763 & 1.763 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.099 & 1.099 & -2.197 & 0 & 0 & 0 & 0 & 0 \\ 1.099 & 1.099 & 0 & -2.197 & 0 & 0 & 0 & 0 \\ 0.775 & 2.058 & 0 & 0 & -2.833 & 0 & 0 & 0 \\ 2.058 & 0.775 & 0 & 0 & 0 & -2.833 & 0 & 0 \\ 0.775 & 2.058 & 0 & 0 & 0 & 0 & -2.833 & 0 \\ 2.058 & 0.775 & 0 & 0 & 0 & 0 & 0 & -2.833 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We can now try to reduce $\alpha = -5x^3 + 115x^2 + 5x \in I$ with the S-unit group. Embed α as a vector using the Log-embedding,

$$y = \mathrm{Log}(\alpha) = (9.49, 9.49, -4.39, 0.0, 0, 0, 0, 0).$$

Then use the S-unit matrix to solve $t \cdot M_S = y$, for the unknown vector $t = (t_1, ..., t_{n-1}, -\lambda)$. We get $t = (0, 2, 0, 0, 0, 0, 0, 1)$. By omitting the last coordinate, we get the S-unit lattice vector $v = 2v_1$, corresponding to the S-unit $u = P_{3,1}^2$. If we divide α by this S-unit, we get

$$\beta = \alpha/u = -25x^3 - 25x - 15 \in I.$$

By comparing the sizes

$$\|\beta\| = 38.41 < 115.22 = \|\alpha\|,$$

we see that α has been reduced significantly. Since $\|\beta\| = 38.41$ and $y = 17$, it looks like we can attempt to reduce the element even further, by choosing a larger value for y.

Let $y = 25$, such that we have $S = \{\infty\} \cup \{P \mid \mathcal{N}(P) \leq 25\}$. By looking at the prime elements above, this gives us at least two new prime ideals, namely $P_{5,1} = (x^2 - 2)$ and $P_{5,2} = (x^2 + 2)$. The remaining prime elements $p < 25$, are $p = 19$ and $p = 23$, and by using Sage the corresponding prime ideals in R have the norm 361 and 529, respectively. The new S contains the elements

$$\{u_1, u_3, P_{3,1}, P_{3,2}, P_{5,1}, P_{5,2}, P_{17,1}, P_{17,3}, P_{17,-1}, P_{17,-3}\},$$

which gives the Log-embedding

$$\begin{aligned}\mathrm{Log}(a) = (&\log|a|_1, \log|a|_3, \log|a|_{P_{3,1}}, \log|a|_{P_{3,2}}, \log|a|_{P_{5,1}}, \log|a|_{P_{5,2}},\\ &\log|a|_{P_{17,1}}, \log|a|_{P_{17,3}}, \log|a|_{P_{17,-1}}, \log|a|_{P_{17,-3}}).\end{aligned}$$

The new S-unit group is now generated by

$$U_S = \{u_0, u_1, P_{3,1}, P_{3,2}, P_{5,1}, P_{5,2}, P_{17,1}, P_{17,3}, P_{17,-1}, P_{17,-3}\},$$

which gives us two new vectors for the S-unit lattice,

$$\begin{aligned}v_5 &= \mathrm{Log}(P_{5,1}) = (1.609, 1.609, 0, 0, -3.219, 0, 0, 0, 0, 0)\\ v_6 &= \mathrm{Log}(P_{5,2}) = (1.609, 1.609, 0, 0, 0, -3.219, 0, 0, 0, 0).\end{aligned}$$

In conclusion, we have the following generators for $\mathcal{L}_S$,

$$\begin{array}{lll} v_1 = \mathrm{Log}(u_1) & v_5 = \mathrm{Log}(P_{5,1}) & v_8 = \mathrm{Log}(P_{17,3})\\ v_3 = \mathrm{Log}(P_{3,1}) & v_6 = \mathrm{Log}(P_{5,2}) & v_9 = \mathrm{Log}(P_{17,-1})\\ v_4 = \mathrm{Log}(P_{3,2}) & v_7 = \mathrm{Log}(P_{17,1}) & v_{10} = \mathrm{Log}(P_{17,-3}),\end{array}$$

corresponding to the matrix M_S by including the unit vector y_0. Now, with a larger group of S-units, we try to further reduce the element $\beta = -25x^3 - 25x - 15$. The Log-embedding of β is,

$$y = \mathrm{Log}(\beta) = (7.296, 0, 0, 0, -3.219, -3.219, 0, 0, 0, 0).$$

Then by solving the equation $t \cdot M_S = y$, we get the vector $t = (0, 0, 0, 0, 1, 1, 0, 0, 0, 0)$, corresponding to the lattice element $v = v_5 + v_6 \in \mathcal{L}_S$ and the S-unit $u = P_{5,1}P_{5,2} \in U_S$. If we divide by this S-unit we get

$$\gamma = \beta/u = 5x^3 + 5x + 3 \in I,$$

and by comparing the sizes

$$\|\gamma\| = 7.68 < 38.41 = \|\beta\|,$$

we have managed to reduce the element even more. Also, $\|\gamma\|$ is much smaller than the parameter $y = 25$, so the element will most likely not be reduced any further by increasing S. In fact, we have found the exact same polynomial as the unknown generator we started with, and we are done.

5.2.2 S-unit Attack for $n = 8$

In the previous example we looked at a case where an S-unit attack worked perfectly. Now, for a slightly larger cyclotomic ring, we showcase how much more complicated the computation of the S-unit group gets and we acknowledge different outcomes when performing an S-unit attack on the same ideal.

Let $R = \mathbb{Z}[x]/(x^8 + 1)$ be the cyclotomic ring. Let $I = (x^6 + x^4 - x^2 + x - 1)$ be the given ideal, where the generator $g = x^6 + x^4 - x^2 + x - 1$ is unknown. The goal is to find the generator or an element close to it, by using S-units. First of all, we have the cyclotomic units

$$\begin{aligned} &u_0 = x & & \\ &u_1 = 1 + x + x^{-1} & &u_3 = 1 + x^3 + x^{-3} \\ &u_5 = 1 + x^5 + x^{-5} & &u_7 = 1 + x^7 + x^{-7}, \end{aligned}$$

where $1/(u_1 u_3 u_5) = -u_7$ and the unit group C_K is generated by u_0, u_1, u_3 and u_5. Assume that we choose the first random element to be $\alpha = -x^6+3x^5+x^4-8x^3+2x^2-x+199 \in I$, where $\|\alpha\| = 199.20$. We want to choose the parameter y to be less than 199. Let $y = 1+2\cdot 8\cdot 6 = 97$, which is the second prime element after 17, on the form $p \in 1+2n\mathbb{Z}$. We use Sage to find the prime ideals in R bounded by 97, which gives the prime ideals lying above the prime elements $3, 7, 17$ and 97. See Appendix B.5 for Sage code and Appendix A.2 for a more comprehensive computation for finding these prime elements.

For the prime elements 3 and 7, we use Sage to find their prime ideal factorization in R and the corresponding generators. We get the following prime ideals,

$$\begin{aligned} P_{3,1} &= (x^4 + x^2 - 1) & P_{3,-1} &= (x^4 - x^2 - 1) \\ P_{7,1} &= (x^2 + x - 1) & P_{7,-1} &= (x^2 - x - 1) \\ P_{7,3} &= (x^5 + x^3 + 1) & P_{7,-3} &= (x^5 + x^3 - 1). \end{aligned}$$

From Example 3.2.3 we have that the generators for the prime ideals lying above 17 are

$$\begin{aligned} P_{17,1} &= (-x^4 - x - 1) & P_{17,-1} &= (x^7 + x^4 - 1) \\ P_{17,3} &= (x^7 + x^3 - x^2) & P_{17,-3} &= (x^6 - x^5 - x) \\ P_{17,5} &= (x^7 - x^4 - x^3) & P_{17,-5} &= (x^5 + x^4 - x) \\ P_{17,7} &= (x^6 - x^2 + x) & P_{17,-7} &= (x^4 - x + 1). \end{aligned}$$

Lastly, we need the prime ideals lying above 97. By choosing the smallest integer for x such that $x^8 + 1 \equiv 0 \pmod{97}$, we get the following factorization

$$\begin{aligned} x^8 + 1 &= \prod_{\text{odd } i<m} (x - \zeta_8^i), \\ 8^8 + 1 &= 97 \cdot 257 \cdot 673 = \prod_{\text{odd } i<m} (8 - \zeta_8^i) \equiv 0 \pmod{97}. \end{aligned}$$

Then by Theorem 3.8 and by letting $x = \zeta_8$, we have the non-principle prime ideals,

$$\begin{aligned} P_{97,1} &= (97, 8 - x) & P_{97,-1} &= (97, 8 - x^{15}) = (97, 8 - x^{-1}) = (97, 8 + x^7) \\ P_{97,3} &= (97, 8 - x^3) & P_{97,-3} &= (97, 8 - x^{13}) = (97, 8 - x^{-3}) = (97, 8 + x^5) \\ P_{97,5} &= (97, 8 - x^5) & P_{97,-5} &= (97, 8 - x^{11}) = (97, 8 - x^{-5}) = (97, 8 + x^3) \\ P_{97,7} &= (97, 8 - x^7) & P_{97,-7} &= (97, 8 - x^9) = (97, 8 - x^{-7}) = (97, 8 + x). \end{aligned}$$

By following the same procedure as we used in Example 3.2.3, with Theorem 3.8 and the the Jacobi sums, we find the generators

$$\begin{aligned} P_{97,1} &= (x^4 + x^3 + 2x^2 + 2x + 1) & P_{97,-1} &= (x^4 + 2x^3 + 2x^2 + x + 1) \\ P_{97,3} &= (x^3 - x^2 + 2x - 1) & P_{97,-3} &= (x^3 - 2x^2 + x - 1) \\ P_{97,5} &= (x^6 + x^5 - 2x^3 - x^2 + 2) & P_{97,-5} &= (x^6 + 2x^5 - x^3 - x^2 + 2) \\ P_{97,7} &= (x^5 - x^3 - x^2 + 2) & P_{97,-7} &= (-2x^5 + x^3 + x^2 - 1). \end{aligned}$$

See Appendix A.2 for a more detailed computation. We have the following generators for the S-unit group U_S:

$$
\begin{array}{ll}
u_0 = x & u_3 = 1 + x^3 + x^{-3} \\
u_1 = 1 + x + x^{-1} & u_5 = 1 + x^5 + x^{-5} \\
P_{3,1} = (x^4 + x^2 - 1) & P_{3,-1} = (x^4 - x^2 - 1) \\
P_{7,1} = (x^2 + x - 1) & P_{7,-1} = (x^2 - x - 1) \\
P_{7,3} = (x^5 + x^3 + 1) & P_{7,-3} = (x^5 + x^3 - 1) \\
P_{17,1} = (-x^4 - x - 1) & P_{17,-1} = (x^7 + x^4 - 1) \\
P_{17,3} = (x^7 + x^3 - x^2) & P_{17,-3} = (x^6 - x^5 - x) \\
P_{17,5} = (x^7 - x^4 - x^3) & P_{17,-5} = (x^5 + x^4 - x) \\
P_{17,7} = (x^6 - x^2 + x) & P_{17,-7} = (x^4 - x + 1) \\
P_{97,1} = (x^4 + x^3 + 2x^2 + 2x + 1) & P_{97,-1} = (x^4 + 2x^3 + 2x^2 + x + 1) \\
P_{97,3} = (x^3 - x^2 + 2x - 1) & P_{97,-3} = (x^3 - 2x^2 + x - 1) \\
P_{97,5} = (x^6 + x^5 - 2x^3 - x^2 + 2) & P_{97,-5} = (x^6 + 2x^5 - x^3 - x^2 + 2) \\
P_{97,7} = (x^5 - x^3 - x^2 + 2) & P_{97,-7} = (-2x^5 + x^3 + x^2 - 1).
\end{array}
$$

Now, define the Log-embedding as,

$$
\begin{aligned}
\operatorname{Log}(a) = (&\log|a|_1, \log|a|_3, \log|a|_5, \log|a|_7, \log|a|_{P_{3,1}}, \log|a|_{P_{3,-1}}, \\
&\log|a|_{P_{7,1}}, \log|a|_{P_{7,-1}}, \log|a|_{P_{7,3}}, \log|a|_{P_{7,-3}}, \log|a|_{P_{17,1}}, \log|a|_{P_{17,3}}, \\
&\log|a|_{P_{17,5}}, \log|a|_{P_{17,7}}, \log|a|_{P_{17,-1}}, \log|a|_{P_{17,-3}}, \log|a|_{P_{17,-5}}, \log|a|_{P_{17,-7}}, \\
&\log|a|_{P_{97,1}}, \log|a|_{P_{97,3}}, \log|a|_{P_{97,5}}, \log|a|_{P_{97,7}}, \log|a|_{P_{97,-1}}, \log|a|_{P_{97,-3}}, \\
&\log|a|_{P_{97,-5}}, \log|a|_{P_{97,-7}}).
\end{aligned}
$$

From the Log-embedding we compute the generators for the S-unit lattice $\mathcal{L}_S$, by embedding each generator of the S-unit group U_S. Then, by letting these vectors be the rows, with the last row being the unit vector y_0, we have the corresponding matrix M_S.

We now go back to the random element $\alpha = -x^6 + 3x^5 + x^4 - 8x^3 + 2x^2 - x + 199 \in I$ and see if there is any S-units in U_S that will reduce the element. Again, we start by embedding α as

$$
\begin{aligned}
y = \operatorname{Log}(\alpha) = (&10.56, 10.66, 10.46, 10.66, 0, 0, 0, 0, 0, \\
&0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0).
\end{aligned}
$$

Then, by solving $t \cdot M_S = y$ we get

$$
\begin{aligned}
t = (&-1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\
&0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
\end{aligned}
$$

which corresponds to the S-unit $u = u_1^{-1} u_5^{-1}$. If we divide α by this S-unit we get

$$
\alpha' = \alpha/u = -203x^7 + 207x^6 + 208x^5 - 4x^4 - 205x^3 - 210x^2 + 206x + 206.
$$

Immediately, this element is much larger than what we started with. Hence, we cannot reduce the element with the current S-unit group. From $\|\alpha\| = 199$ and the next prime element being $1 + 2 \cdot 8 \cdot 7 = 113$, which is still smaller than $\|\alpha\|$, we expand S with $y = 113$. Let $P_{113,c}$ denote the prime ideals in R lying above 113, for $c \in \{\pm 1, \pm 3, \pm 5, \pm 7\}$. By adding these to the S-unit group and by preforming another S-unit attack, we find the S-unit $u = P_{113,-5} u_1 u_3 u_5$. This gives a new element

$$
\alpha' = \alpha/u = 88x^7 - 65x^6 - 28x^5 + 40x^4 - 46x^3 - 4x^2 - 29x + 11.
$$

Now, the size is $\|\alpha'\| = 132$ and we managed to slightly reduce α. However, $\alpha' \notin I$ and the prime ideal $P_{113,-5}$ must be in the generator of the ideal. This brings us back to the original element α. Lastly, we try to divide by the prime ideal $P_2 = (x+1)$, as described in Section 4.2. We get

$$\begin{aligned}\alpha/P_2 &= (-x^6+3x^5+x^4-8x^3+2x^2-x+199)/(x+1)\\ &= \frac{1}{2}(-207x^7+207x^6-209x^5+215x^4-213x^3+197x^2-193x+191).\end{aligned}$$

This means α does not contain P_2 and we cannot reduce by it. Also, from the prime ideal factorization of α' we have two large prime ideals remaining,

$$\alpha' = (13981313, x+6367574)(1548446177, x+75120017),$$

which confirms that we cannot reduce it any further. In conclusion, we cannot find any smaller element in I from α.

We try to reduce another random element from I. Let $\beta = 3x^7+x^5-x^3+x+94 \in I$, with $\|\beta\| = 94.06$. This indicates that we can try to reduce β by choosing the parameter $y = 17$ for S. Then we have the elements

$$\begin{array}{ll}
u_1 = 1+x+x^{-1} & u_5 = 1+x^5+x^{-5}\\
u_3 = 1+x^3+x^{-3} & u_7 = 1+x^7+x^{-7}\\
P_{3,1} = (x^4+x^2-1) & P_{3,-1} = (x^4-x^2-1)\\
P_{7,1} = (x^2+x-1) & P_{7,-1} = (x^2-x-1)\\
P_{7,3} = (x^5+x^3+1) & P_{7,-3} = (x^5+x^3-1)\\
P_{17,1} = (-x^4-x-1) & P_{17,-1} = (x^7+x^4-1)\\
P_{17,3} = (x^7+x^3-x^2) & P_{17,-3} = (x^6-x^5-x)\\
P_{17,5} = (x^7-x^4-x^3) & P_{17,-5} = (x^5+x^4-x)\\
P_{17,7} = (x^6-x^2+x) & P_{17,-7} = (x^4-x+1),
\end{array}$$

in S defining the infinite and finite norms, with the Log-embedding

$$\begin{aligned}\mathrm{Log}(a) = (&\log|a|_1, \log|a|_3, \log|a|_5, \log|a|_7, \log|a|_{P_{3,1}}, \log|a|_{P_{3,-1}},\\ &\log|a|_{P_{7,1}}, \log|a|_{P_{7,-1}}, \log|a|_{P_{7,3}}, \log|a|_{P_{7,-3}}, \log|a|_{P_{17,1}}, \log|a|_{P_{17,3}},\\ &\log|a|_{P_{17,5}}, \log|a|_{P_{17,7}}, \log|a|_{P_{17,-1}}, \log|a|_{P_{17,-3}}, \log|a|_{P_{17,-5}}, \log|a|_{P_{17,-7}}).\end{aligned}$$

The S-unit group U_S is generated by,

$$\begin{array}{ll}
u_0 = x & u_3 = 1+x^3+x^{-3}\\
u_1 = 1+x+x^{-1} & u_5 = 1+x^5+x^{-5}\\
P_{3,1} = (x^4+x^2-1) & P_{3,-1} = (x^4-x^2-1)\\
P_{7,1} = (x^2+x-1) & P_{7,-1} = (x^2-x-1)\\
P_{7,3} = (x^5+x^3+1) & P_{7,-3} = (x^5+x^3-1)\\
P_{17,1} = (-x^4-x-1) & P_{17,-1} = (x^7+x^4-1)\\
P_{17,3} = (x^7+x^3-x^2) & P_{17,-3} = (x^6-x^5-x)\\
P_{17,5} = (x^7-x^4-x^3) & P_{17,-5} = (x^5+x^4-x)\\
P_{17,7} = (x^6-x^2+x) & P_{17,-7} = (x^4-x+1),
\end{array}$$

and we compute the generator for the S-unit lattice $\mathcal{L}_S$. Embed β as a vector,

$$\begin{aligned}y = \mathrm{Log}(\beta) = (&9.03, 9.11, 9.07, 9.14, 0, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, -2.83, 0, -2.83, 0),\end{aligned}$$

and solve $t \cdot M_S = y$, such that we get

$$t = (-1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0).$$

This corresponds to the S-unit $u = u_1^{-1} u_5^{-1} P_{17,5} P_{17,7}$. Divide by this S-unit to get

$$\beta' = \beta/u = -22x^7 - 33x^6 - 4x^5 + 23x^4 + 16x^3 - x^2 - 28x + 23.$$

which has the size

$$\|\beta'\| = 60.73 < 94.06 = \|\beta\|.$$

The new element is smaller, and since the next prime ideals to include in S are the prime ideals above $p = 97$, we cannot reduce β' any further by expanding S. Instead we try to reduce β' with $P_2 = (x + 1)$,

$$\begin{aligned}
\beta'/P_2 &= -25x^7 + 3x^6 - 36x^5 + 32x^4 - 9x^3 + 25x^2 - 26x - 2 \\
&\vdots \\
\beta'/P_2^6 &= -2529x^7 + 3041x^6 - 3086x^5 + 2658x^4 - 1828x^3 + 719x^2 + 497x - 1635 \\
\beta'/P_2^7 &= \frac{1}{2}(-11729x^7 + 6671x^6 - 589x^5 - 5583x^4 + 10899x^3 - 14555x^2 + 15993x - 14999).
\end{aligned}$$

Hence, we divide β' by P_2^6. However, the new element $\beta'' = \beta'/P_2^6$ is significantly larger than β', but if we preform a unit attack on β'', we get

$$\gamma = \beta''/(u_1^{-2} u_3^{-4} u_5^{-1}) = -14x^7 + 9x^6 - 6x^5 - 22x^3 + 9x^2 + 8x - 11,$$

with $\|\gamma\| = 32.60$. We managed to further reduce β, and if we look at the prime ideal factorization of γ

$$\gamma = (113, x - 48)(2921814977, x - 636425943),$$

we cannot reduce it any further. Because, from the first element α we know that the prime ideal above $p = 113$ is in the generator for I, and the other prime ideal is too large. Hence, we have reduced $\beta = 3x^7 + x^5 - x^3 + x + 94$ to $\gamma = -14x^7 + 9x^6 - 6x^5 - 22x^3 + 9x^2 + 8x - 11$, where

$$\|\beta\| = 94.06 > 32.60 = \|\gamma\|.$$

We end this example by choosing an element from I where we manage to reduce it perfectly. Let $\delta = -136x^7 - 397x^6 - 206x^5 + 181x^4 + 435x^3 + 174x^2 - 104x - 158$ be the element from I with $\|\delta\| = 711.75$. Because of the large size, we start by choosing the parameter $y = 97$ for S. We use the same S-unit group and Log-embedding as the first element α. The following Log-embedding and the solution to $t \cdot M_S = y$ give us

$$\begin{aligned}
y = \mathrm{Log}(\delta) = (&12.58, 14.36, 7.08, 9.22, -8.79, 0, -3.89, 0, 0, -11.68, \\
&0, 0, 0, 0, -2.83, -8.50, 0, -2.83, 0, 0, 0, 0, 0, 0, 0, 0)
\end{aligned}$$

and

$$t = (-1, 1, 0, 2, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0),$$

which corresponds to the S-unit $u = u_1^{-1} u_3 P_{3,1}^2 P_{7,1}^2 P_{7,4}^3 P_{17,5} P_{17,-5}^3 P_{17,-7}$. If we reduce by this, we get

$$\delta' = \delta/u = x^7 + x^3 + x^2 - x - 1,$$

with $\|\delta'\| = 2.23$. This is significantly smaller than our initial δ. Further, we can see that δ' is the same up to units as the unknown generator $g = x^6 + x^4 - x^2 + x - 1$ for the ideal I, i.e. $\delta' = -u_7 \cdot g$.

In conclusion, if we choose the right element from the known ideal we find a very small element, in fact even the generator for the ideal. But, in most cases we only manage to choose elements that slightly reduces, and in some cases not at all. This might not be the case when we are working with an actual cryptosystem and not just random elements from a general ideal. We discuss this further in Chapter 6. We should also mention the complexity of computing the generators for the prime ideals. As shown in this example it became a lot more complicated to find the generators, merely by doubling the dimension of the cyclotomic ring. This indicates that it only gets harder and harder for higher dimensional cyclotomic rings, and maybe not even possible.

5.3 Applying S-unit Attacks to NTRU

To end this chapter we study one method of using S-units for a key recovery attack against NTRU. We provide one case where we manage to fully recover the keys and two cases where it does not work. Since the secret key consists of polynomials with coefficients equal to $-1, 0$ or 1, they are be small and possibly S-units. This is because we choose S-units depending on the size of the public key, which is much larger than the secret key. They also have the relation described in Section 2.2.1.

5.3.1 An Example Where it Works

For this example, the S-unit attack will work perfectly. Let $(n, p, q, d) = (8, 3, 41, 2)$ and let $R = \mathbb{Z}[x]/(x^8 + 1)$ be the cyclotomic ring for the NTRU cryptosystem. Let the secret key be

$$f(x) = x^6 - x^4 + x^3 + x^2 - 1 \quad \text{and} \quad g(x) = x^6 + x^4 - x^2 - x$$

which gives the public key

$$h(x) = -7x^7 - 10x^6 + 5x^5 + 5x^4 + 20x^3 - 19x^2 + 3x - 12.$$

Observe that $\|h(x)\| = 33.36$ and we choose the parameter $y = 30$ for S. We have that S consists of the cyclotomic units

$$\begin{array}{ll} u_1 = 1 + x + x^{-1} & u_5 = 1 + x^5 + x^{-5} \\ u_3 = 1 + x^3 + x^{-3} & u_7 = 1 + x^7 + x^{-7}. \end{array}$$

From Example 5.2.2 we have the prime ideals with norm less than 33,

$$\begin{array}{ll} P_{3,1} = (x^4 + x^2 - 1) & P_{3,-1} = (x^4 - x^2 - 1) \\ P_{7,1} = (x^2 + x - 1) & P_{7,-1} = (x^2 - x - 1) \\ P_{7,3} = (x^5 + x^3 + 1) & P_{7,-3} = (x^5 + x^3 - 1) \\ P_{17,1} = (-x^4 - x - 1) & P_{17,-1} = (x^7 + x^4 - 1) \\ P_{17,3} = (x^7 + x^3 - x^2) & P_{17,-3} = (x^6 - x^5 - x) \\ P_{17,5} = (x^7 - x^4 - x^3) & P_{17,-5} = (x^5 + x^4 - x) \\ P_{17,7} = (x^6 - x^2 + x) & P_{17,-7} = (x^4 - x + 1). \end{array}$$

The S-unit group U_S is generated by

$$\begin{array}{ll}
u_0 = x & u_3 = 1 + x^3 + x^{-3} \\
u_1 = 1 + x + x^{-1} & u_5 = 1 + x^5 + x^{-5} \\
P_{3,1} = (x^4 + x^2 - 1) & P_{3,-1} = (x^4 - x^2 - 1) \\
P_{7,1} = (x^2 + x - 1) & P_{7,-1} = (x^2 - x - 1) \\
P_{7,3} = (x^5 + x^3 + 1) & P_{7,-3} = (x^5 + x^3 - 1) \\
P_{17,1} = (-x^4 - x - 1) & P_{17,-1} = (x^7 + x^4 - 1) \\
P_{17,3} = (x^7 + x^3 - x^2) & P_{17,-3} = (x^6 - x^5 - x) \\
P_{17,5} = (x^7 - x^4 - x^3) & P_{17,-5} = (x^5 + x^4 - x) \\
P_{17,7} = (x^6 - x^2 + x) & P_{17,-7} = (x^4 - x + 1).
\end{array}$$

As before we have the lattice $\mathcal{L}_S = \mathrm{Log}(U_S)$ with the corresponding matrix M_S. By embedding the public key $h(x)$ as $y = \mathrm{Log}(h(x))$ and solving the linear system $t \cdot M_S = y$, we get the S-unit $u = u_3^{-1} P_{17,2} P_{17,8}$. This S-unit is a representative for $f(x)$,

$$\tilde{f}(x) = u = -2x^7 + x^6 - x^5 + 2x^4 - x^3 - x^2 - 1.$$

Some of the coefficients are bigger than ±1 and 0, so by performing a unit attack, we get the unit $u = u_3^{-1}$ and the following representative for $f(x)$,

$$\tilde{f}(x) \cdot u_3 \equiv x^7 - x^5 - x^3 + x - 1 \equiv -x^3 \cdot f(x) \pmod{41}.$$

If we now multiply this representative with $h(x)$ modulo 41, we get

$$\tilde{f}(x) \cdot h(x) = \tilde{g}(x) \equiv -x^7 - x^6 - x^3 - x \equiv -x^3 \cdot g(x) \pmod{41}$$

as a representative for $g(x)$. Hence, we have recovered both of the polynomials for the secret key.

Remark 5.1. It is not obvious that this method of an S-unit attack will work for arbitrary instances of NTRU. This example works because the prime ideals in $f(x)$ was also in $h(x)$, i.e. $f(x)$ was an S-unit. Also, $g(x)$ did not contain the same prime ideals as $f(x)$ and it was not an S-unit.

Remark 5.2. We should also mention that after finding the S-unit representing the secret key $f(x)$, we had to preform an additional unit attack. Does this mean it would be easier to look at one S-unit factor at a time, for the representative of $f(x)$, instead of first calculating the representative with all the factors from the S-unit group and then reduce it? For this example, we only had to reduce $\tilde{f}(x)$ by a cyclotomic unit consisitng of only one factor. Thus, the approach done in the example gives less computations and is more effective.

5.3.2 Two Examples Where it Does Not Work

By using a more realistic approach, we give two examples where this method of S-unit attack against NTRU does not work. We do this by not checking the factorization of the secret and public keys beforehand. Instead, we generate an arbitrary key set.

Example 5.3. Let $(n, p, q, d) = (8, 3, 41, 3)$ and let $R = \mathbb{Z}[x]/(x^8 + 1)$ be the cyclotomic ring for the NTRU cryptosystem. Let the secret key be

$$f(x) = x^7 - x^6 - x^4 + x^3 + x^2 - x + 1 \quad \text{and} \quad g(x) = x^6 - x^5 - x^4 - x^3 + x^2 + 1$$

with the public key

$$h(x) = -8x^7 + 2x^6 + 9x^5 + 17x^4 - 12x^3 - 10x^2 + 2x + 15.$$

The public key has the size $\|h(x)\| = 30.18$, and we use the same S-unit group as the previous example. By computing the Log-embedding for $h(x)$ and solving $t \cdot M_S = y$, we get the S-unit $u = u_1 u_3 P_{17,2} P_{17,2}$. This corresponds to the polynomial

$$\tilde{f}(x) = -x^7 + x^6 - x^4 - x^3 - x^2 + x + 1.$$

The coefficients for this polynomials looks similar to $f(x)$, but it is not the same polynomial up to units. By looking at the prime ideal factorization of $f(x)$, it contains the prime ideals $P_{17,1}$ and $P_{17,5}$, whereas $\tilde{f}(x)$ contains the prime ideals $P_{17,-1}$ and $P_{17,3}$. Moreover, if we use $\tilde{f}(x)$ to find a representative for $g(x)$, we get

$$\tilde{g}(x) = \tilde{f}(x) \cdot h(x) \equiv 8x^7 - 18x^6 + 15x^5 + 5x^4 + 2x^3 - 17x^2 - 19x - 19 \pmod{41},$$

which is far from the polynomial $g(x)$. Also, by performing a unit attack we do not find any units that reduces it.

Example 5.4. Let $(n, p, q, d) = (8, 3, 41, 2)$ and let $R = \mathbb{Z}[x]/(x^8 + 1)$ be the cyclotomic ring for the NTRU cryptosystem. Let the secret key be

$$f(x) = x^6 + x^5 + x^4 - x - 1 \quad \text{and} \quad g(x) = x^6 - x^3 + x^2 - x,$$

with the public key

$$h(x) = 17x^7 - 6x^6 + 3x^5 + 17x^4 - 15x^3 + 2x^2 - 17x + 13,$$

with $\|h(x)\| = 36.19$. This means we use the same S-unit group once again. By embedding $h(x)$ and solving $t \cdot M_S = y$, we get the S-unit

$$\tilde{f}(x) = u_3^{-1} = x^7 - x^6 + x^2 - x + 1$$

representing $f(x)$. Again, the coefficients are as wanted, but by looking at the prime ideal factorization for $f(x)$ and $\tilde{f}(x)$ we get $P_{97} = (97, x + 8)$ and (1), respectively. Hence, they are not the same polynomial up to units. We also get the polynomial

$$\tilde{g}(x) = \tilde{f}(x) \cdot h(x) \equiv 15x^7 + 19x^6 - 6x^5 - 16x^4 - 7x^3 - 18x^2 + 18x + 14 \pmod{41},$$

representing $g(x)$, with no further reduction from a unit attack.

From these three examples, we can draw the conclusion that this method of S-unit attack against NTRU, is not very effective. It only worked when we carefully chose the right polynomials for the secret key, which gave us the right polynomial for the public key. The main issue with this approach is that we are working with modulo q in NTRU. Even thought we have the relations $f_q(x) = f^{-1}(x) \pmod{q}$ and $h(x) = f_q(x) \cdot g(x) \pmod{q}$, which indicates that $h(x)$, $f_q(x)$ and $g(x)$ would contain the same prime ideal factors, the modulo will most likely change the factorization and we cannot find the same S-units in the secret key and the public key.

Chapter 6

Conclusion

In post-quantum cryptography the goal is to develop cryptographic schemes that are both secure on classical and quantum computers. The most promising candidate to obtain this, is lattice-based cryptography, and for this reason very interesting to analyze. By NIST suggestion to use it as the new standard for public key encryption, it is important to consider all the potential attacks, whereas S-unit attacks is one of them.

As studied in this thesis, S-unit attacks acquire a very different approach than other well known reduction algorithms. By considering the properties of the cyclotomic ring, rather than only considering the lattice, it proposes some compelling aspects that are worth studying. For this thesis, the most important aspects to consider, lies within the assumptions we have made.

First of all, it is worth mentioning that we have not taken into consideration the time estimate nor the storage capacity needed for performing an S-unit attack, two important elements when analyzing attacks against a cryptosystem. However, Example 5.2.2 indicates how the complexity of computing the S-unit group increases with the dimension of the cyclotomic ring.

By the assumption of working in a principal ideal domain, it is easier to find the generators for the S-unit group and the generators are more precise than the non-principal case. This is because each prime ideal have a generator which again generate the S-unit group, instead of the group being generated by S-generators for arbitrary ideals. Also, we assumed that the degree of the cyclotomic ring is a power of 2, which makes them even more likely to be a PID. For a cryptosystem, we have cyclotomic rings of high dimensions and the dimension is not necessarily a power of 2. Therefore, as discussed in Section 3.1.2, they are most likely not a PID. To solve this problem we can use the power of quantum computers. As mentioned in Section 3.1.2, finding a principal representative in the non-principal case, have an additional quantum algorithm to consider. Because of this algorithm, finding a such representative and performing an S-unit attack can be considered as to separate problems. Although, it is still uncertain of how much the non-principal case will influence the performance of an S-unit attack.

Another assumption we made is how to determine the parameter y for the prime ideals in S. The best way to do so is still unclear. In Section 4.2, we made the decision to choose y depending on size of the element we want to reduce. Start with a small value for y, compared to the size of the element. Then gradually increase y until no further reduction is possible, or until we no longer have an element in the ideal. As discussed in Remark 4.3, this choice of y makes it more likely for the element to stay in the ideal after the reduction. It also provides better control for which prime ideals to include and not include in S.

The decision of excluding the prime ideal $P_2 = (x+1)$ from the S-unit group is also an important aspect to consider. We decided to handle it as an isolated case at the end of the attack, because 2 is the only prime number that ramifies in the ring. Likewise, with the remark about units after Example 5.3.1. From our study, it suggest that these choices are the most effective.

We should also discuss the choice of the elements to reduce for a given ideal. The best way on how to do so is still unclear. From this thesis, we can conclude that an S-unit attack worked best for a general ideal in a cyclotomic ring. When we tried to reduce elements chosen from a general ideal, we managed for the most part to reduce the elements, and in some cases even fully.

In conclusion, we see that S-unit attacks can work very well for finding small elements in a general ideal. With more help from quantum algorithms, S-unit attacks seems plausible. It is much more prone to find small elements than the LLL-algorithm, or other similar lattice reduction algorithms for that matter. Which is discussed in more detail by Bernstein and Lange [4]. However, as seen in Section 5.3, our method for using S-units for a key recovery attack against NTRU did not execute very well. Either way, this thesis as compiled the most essential mathematical theory to consider when analyzing S-unit attacks. Which is the most important part for determining if it actually works and if it is a potential threat to lattice-based cryptography. We have also constructed new examples showcasing the computational details to S-unit attacks. At last, we have highlighted the most important aspects to consider and that could be interesting to analyze further.

As a continuation of this thesis, it would be interesting to investigate what happens in the non-principal case. How will the quantum algorithm for this affect the S-unit attack? Also, is it possible to find a quantum algorithm for computing the generators for the S-unit group? Bernstein also discusses [1] a potential of an algorithm for finding the norm for an arbitrary ideal. This will likely make the S-unit attacks even more precise, as noted in Section 4.3. Lastly, it would be interesting to analyze other methods for utilizing S-units for a key recovery, such as described by Stehlé [9].

References

[1] Bernstein, Daniel J. *Fast Norm Computation in Smooth-Degree Abelian Number Fields.* 2022. URL: https://s-unit.attacks.cr.yp.to/abeliannorms-20220731.pdf.

[2] Bernstein, Daniel J. *S-unit attacks.* 2021. URL: https://cr.yp.to/talks/2021.08.20/slides-djb-20210820-sunitattacks-4x3.pdf.

[3] Biasse, Jean-François. *Subexponential algorithms for finding a short generator of a principal ideal and solving γ-SVP in $\mathbb{Q}(\zeta_{p^s})$.* 2017. URL: https://arxiv.org/pdf/1503.03107.pdf.

[4] Daniel J. Bernstein, Tanja Lange. *Non-randomness of S-unit lattices.* 2021. URL: https://eprint.iacr.org/2021/1428.pdf.

[5] Ellis, James H. *Possibility of Non-Secret Encryption.* 1970. URL: https://cryptocellar.org/cesg/possnse.pdf.

[6] Janusz, Gerald J. *Algebraic Number Fields.* 2nd ed. Graduate Studies in Mathematics, Vol. 7. American Mathematical Society, 1996. ISBN: 0-8218-0429-4.

[7] Jean-François Biasse, Fang Song. *Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields.* 2016. URL: https://fangsong.info/files/pubs/BS_SODA16.pdf.

[8] Jeffrey Hoffstein Jill Pipher, Joseph H. Silverman. *An Introduction to Mathematical Cryptography.* 2nd ed. Springer New York Heidelberg Dordrecht London, 2014. ISBN: 978-1-4939-1710-5.

[9] Jöel Felderhoff, Alice Pellet-Mary and Stehlé, Damien. *On Module Unique-SVP and NTRU.* 2022. URL: https://eprint.iacr.org/2022/1203.pdf.

[10] Miller, John C. *Class Numbers of Totally Real Fields and Applications to the Weber Class Number Problem.* 2014. URL: https://arxiv.org/pdf/1405.1094.pdf.

[11] NIST, Computer Security Resource Center. *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms.* 2022. URL: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms (visited on 07/07/2022).

[12] Peikert, Chris. *A Decade of Lattice Cryptography.* 2016. URL: https://eprint.iacr.org/2015/939.pdf.

[13] Shor, Peter W. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* 1996. URL: https://arxiv.org/pdf/quant-ph/9508027.pdf.

[14] Stein, William. *SageMath, Open-Source Mathematical Software System.* 2005. URL: https://www.sagemath.org/.

[15] Washington, Lawrence C. *Introduction to Cyclotomic Fields*. 2nd ed. Graduate Texts in Mathematics, No. 83. Springer-Verlag New York, Inc., 1997. ISBN: 0-387-94762-0.

Appendix A

Computations

A.1 Computations for Example 2.27

A step by step computation of the LLL algorithm where most of the calculations are done by using Sage B.3.

1. Set $k = 2$ and set $v_1^* = v_1 = (10, 194, -118, 22)$.

2. Then for $k = 2 \leq 4$, we have $j = 1$ and we calculate

$$v_2 = v_2 - \lfloor \mu_{2,1} \rceil \cdot v_1 = (56, -31, -4, -7) \text{ and}$$
$$v_2^* = v_2 - \mu_{2,1} \cdot v_1^* = (56.985, -11.8912, -15.623, -4.833),$$

   where
$$\mu_{2,1} = \frac{v_2 \cdot v_1^*}{\|v_1^*\|^2} = 0.902 \Rightarrow \lfloor \mu_{2,1} \rceil = 1.$$

3. Now we have $v_1^* = (10, 194, -118, 22)$, $v_2^* = (56.985, -11.8912, -15.623, -4.833)$ and $v_2 = (56, -31, -4, -7)$, and we need to check if $\|v_2^*\|^2 \geq (\frac{3}{4} - \mu_{2,1}^2)\|v_1^*\|^2$ (Lovász condition 2.23). We get

$$\|v_2^*\|^2 = 3656.122 < 38602.122 = (\frac{3}{4} - \mu_{2,1}^2)\|v_1^*\|^2,$$

   and we need to swap v_1 and v_2.

4. Now $v_1 = v_1^* = (56, -31, -4, -7)$ and $v_2 = (10, 194, -118, 22)$. For $k = 2 \leq 4$, we have $j = 1$ and we calculate

$$v_2 = v_2 - \lfloor \mu_{2,1} \rceil \cdot v_1 = (66, 163, -122, 15) \text{ and}$$
$$v_2^* = v_2 - \mu_{2,1} \cdot v_1^* = (79.105, 155.745, -122.936, 13.362),$$

   where
$$\mu_{2,1} = \frac{v_2 \cdot v_1^*}{\|v_1^*\|^2} = -1.23 \Rightarrow \lfloor \mu_{2,1} \rceil = -1.$$

5. We check if $\|v_2^*\|^2 \geq (\frac{3}{4} - \mu_{2,1}^2)\|v_1^*\|^2$ (Lovász condition 2.23), and we get

$$\|v_2^*\|^2 = 45806.062 > 2893.562 = (\frac{3}{4} - \mu_{2,1}^2)\|v_1^*\|^2.$$

   We do not need to swap.

6. Now set $k = 3$ and try to reduce v_3.
   For $j = 2$ we set

$$v_3 = v_3 - \lfloor \mu_{3,2} \rceil v_2 = (-28, 63, 155, 65),$$

where

$$\mu_{3,2} = \frac{v_3 \cdot v_2^*}{\|v_2^*\|^2} = -0.231 \Rightarrow \lfloor \mu_{2,1} \rceil = 0.$$

For $v_3 = (-28, 63, 155, 65)$ and $j = 1$ we compute

$$v_3 = v_3 = v_3 - \lfloor \mu_{3,1} \rceil v_1 = (28, 32, 151, 58) \text{ and}$$
$$v_3^* = v_3 - \mu_{3,1} \cdot v_1^* - \mu_{3,2} \cdot v_2^* = (47.480, 62.455, 125.369, 60.370)$$

where

$$\mu_{3,1} = \frac{v_3 \cdot v_1^*}{\|v_1^*\|^2} = -1.104 \Rightarrow \lfloor \mu_{2,1} \rceil = -1.$$

7. We need to check the Lovász condition 2.23,

$$\|v_3^*\|^2 = 25479.620 < 31906.423 = (\frac{3}{4} - \mu_{3,2}^2)\|v_2^*\|^2.$$

We need to swap v_2 and v_3.

8. Set $k = 3$ and try to reduce v_3 again. Now $v_1 = (56, -31, -4, -7)$, $v_2 = (28, 32, 151, 58)$ and $v_3 = (66, 163, -122, 15)$
   For $j = 2$ we set

$$v_3 = v_3 - \lfloor \mu_{3,2} \rceil v_2 = (66, 163, -122, 15),$$

where

$$\mu_{3,2} = \frac{v_3 \cdot v_2^*}{\|v_2^*\|^2} = -0.379 \Rightarrow \lfloor \mu_{3,2} \rceil = 0.$$

For $j = 1$ we set

$$v_3 = v_3 - \lfloor \mu_{3,1} \rceil v_1 = (66, 163, -122, 15) \text{ and}$$
$$v_3^* = v_3 - \mu_{3,1} \cdot v_1^* - \mu_{3,2} \cdot v_2^* = (89.722, 167.879, -65.680, 35.354)$$

where

$$\mu_{3,1} = \frac{v_3 \cdot v_1^*}{\|v_1^*\|^2} = -0.234 \Rightarrow \lfloor \mu_{3,1} \rceil = 0.$$

9. We need to check if $\|v_3^*\|^2 \geq (\frac{3}{4} - \mu_{3,2}^2)\|v_2^*\|^2$ (Lovász condition 2.23). We have

$$\|v_3^*\|^2 = 46034 > 16930.485 = (\frac{3}{4} - \mu_{3,2}^2)\|v_2^*\|^2,$$

so we do not need to swap.

10. We set $k = 4$ and try to reduce v_4 for $j = 3, 2, 1$.
    For $j = 3$ we compute

$$v_4 = v_4 - \lfloor \mu_{4,3} \rceil v_3 = (-158, -39, -149, 146),$$

where

$$\mu_{4,3} = \frac{v_4 \cdot v_3^*}{\|v_3^*\|^2} = -0.138 \Rightarrow \lfloor \mu_{4,3} \rceil = 0.$$

For $j = 2$ we compute

$$v_4 = v_4 - \lfloor \mu_{4,2} \rceil v_2 = (-130, -7, 2, 204),$$

where

$$\mu_{4,2} = \frac{v_4 \cdot v_2^*}{\|v_2^*\|^2} = -0.735 \Rightarrow \lfloor \mu_{4,2} \rceil = -1.$$

For $j = 1$ we compute

$$v_4 = v_4 - \lfloor \mu_{4,1} \rceil v_1 = (-18, -69, -6, 190) \text{ and}$$
$$v_4^* = v_4 - \mu_{4,1} \cdot v_1^* - \mu_{4,2} \cdot v_2^* - \mu_{4,3} \cdot v_3^* = (-13.956, -56.308, -62.898, 176.438),$$

where

$$\mu_{4,1} = \frac{v_4 \cdot v_1^*}{\|v_1^*\|^2} = -2.04 \Rightarrow \lfloor \mu_{4,1} \rceil = -2.$$

11. Lastly, we check Lovász condition 2.23 and we have

$$\|v_4^*\|^2 = 41221 > 33549.094 = (\frac{3}{4} - \mu_{4,3}^2)\|v_3^*\|^2.$$

12. We are done and we have the reduce basis

$$\begin{aligned} v_1 &= (56, -31, -4, -7) \\ v_2 &= (28, 32, 151, 58) \\ v_3 &= (66, 163, -122, 15) \\ v_4 &= (-18, -69, -6, 190). \end{aligned}$$

A.2 Computations for Example 5.2.2

Here we showcase the computation for finding the prime ideals in R with norm less than or equal to 97. We have written out the computation for the primes 3 to 29, and in a similar manner we find the prime ideals for the primes above 31 to 97.

$$\begin{array}{ll}
\boldsymbol{p=3:} & \\
P_{3,1}=(x^4+x^2-1) & \mathcal{N}(P_{3,1})=81\leq 97\\
P_{3,-1}=(x^4-x^2-1) & \mathcal{N}(P_{3,-1})=81\leq 97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=5:} & \\
P_{5,1}=(x^4+2) & \mathcal{N}(P_{5,1})=625>97\\
P_{5,-1}=(x^4-2) & \mathcal{N}(P_{5,-1})=625>97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=7:} & \\
P_{7,1}=(x^2+x-1) & \mathcal{N}(P_{7,1})=49\leq 97\\
P_{7,3}=(x^5+x^3+1) & \mathcal{N}(P_{7,3})=49\leq 97,\\
P_{7,-1}=(x^2-x-1) & \mathcal{N}(P_{7,-1})=49\leq 97\\
P_{7,-3}=(x^5+x^3-1) & \mathcal{N}(P_{7,3})=49\leq 97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=11:} & \\
P_{11,1}=(x^6+x^2+3) & \mathcal{N}(P_{11,1})=14641>97\\
P_{11,2}=(x^6-3x^4-x^2) & \mathcal{N}(P_{11,2})=14641>97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=13:} & \\
P_{13,1}=(2x^4+3) & \mathcal{N}(P_{13,1})=28561>97\\
P_{13,-1}=(-2x^4+3) & \mathcal{N}(P_{13,-1})=28561>97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=17:} & \\
P_{17,1}=(-x^4-x-1) & \mathcal{N}(P_{17,1})=17\leq 97\\
P_{17,3}=(x^7+x^3-x^2) & \mathcal{N}(P_{17,3})=17\leq 97\\
P_{17,5}=(x^7-x^4-x^3) & \mathcal{N}(P_{17,5})=17\leq 97\\
P_{17,7}=(x^6-x^2+x) & \mathcal{N}(P_{17,7})=17\leq 97\\
P_{17,-1}=(x^7+x^4-1) & \mathcal{N}(P_{17,-1})=17\leq 97\\
P_{17,-3}=(x^6-x^5-x) & \mathcal{N}(P_{17,-3})=17\leq 97\\
P_{17,-5}=(x^5+x^4-x) & \mathcal{N}(P_{17,-5})=17\leq 97\\
P_{17,-7}=(x^4-x+1) & \mathcal{N}(P_{17,-7})=17\leq 97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=19:} & \\
P_{19,1}=(-3x^4+x^2+3) & \mathcal{N}(P_{19,1})=130321>97\\
P_{19,-1}=(3x^4+x^2-3) & \mathcal{N}(P_{19,-1})=130321>97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=23:} & \\
P_{23,1}=(x^4-x^3+x^2+x+1) & \mathcal{N}(P_{23,1})=529>97\\
P_{23,2}=(x^6-x^4-x^3-x+1) & \mathcal{N}(P_{23,-1})=529>97,\\
P_{23,2}=(x^6-x^4+x^3+x+1) & \mathcal{N}(P_{23,-1})=529>97,\\
P_{23,2}=(x^4+x^3+x^2-x+1) & \mathcal{N}(P_{23,-1})=529>97,
\end{array}$$

$$\begin{array}{ll}
\boldsymbol{p=29:} & \\
P_{29,1}=(2x^4+5) & \mathcal{N}(P_{29,1})=707281>97\\
P_{29,-1}=(-2x^4+5) & \mathcal{N}(P_{29,-1})=707281>97,
\end{array}$$

For finding the generators for the prime ideals lying above 97, we start by choosing the smallest integer for x such that $x^8+1 \equiv 0 \pmod{97}$. We get the following factorization of the minimal polynomial,

$$\begin{aligned} x^8+1 &= \prod_{\text{odd } i<m} (x-\zeta_8^i), \\ 8^8+1 = 97 \cdot 257 \cdot 673 &= \prod_{\text{odd } i<m} (8-\zeta_8^i) \equiv 0 \pmod{97}. \end{aligned}$$

Then by Kummers theorem 3.8 and by letting $x=\zeta_8$, we have the following non principle prime ideals,

$$\begin{array}{ll} P_{97,1} = (97, 8-x) & P_{97,-1} = (97, 8-x^{15}) = (97, 8-x^{-1}) = (97, 8+x^7) \\ P_{97,3} = (97, 8-x^3) & P_{97,-3} = (97, 8-x^{13}) = (97, 8-x^{-3}) = (97, 8+x^5) \\ P_{97,5} = (97, 8-x^5) & P_{97,-5} = (97, 8-x^{11}) = (97, 8-x^{-5}) = (97, 8+x^3) \\ P_{97,7} = (97, 8-x^7) & P_{97,-7} = (97, 8-x^9) = (97, 8-x^{-7}) = (97, 8+x). \end{array}$$

From Sage B.5 we get the following generators for the principle ideals,

$$\begin{array}{ll} P_{97,1}P_{97,-1} & \Rightarrow g_1 = x^7+2x^6+x^5+2x^4+x^3+2x^2-2 \\ P_{97,3}P_{97,-3} & \Rightarrow g_3 = 2x^7+x^6+2x^5+2x^3-2x \\ P_{97,5}P_{97,-5} & \Rightarrow g_5 = x^7+2x^6-2x^4-x^3+2x^2+x+2 \\ P_{97,7}P_{97,-7} & \Rightarrow g_7 = 3x^7+x^6-2x^4+2x^2-1. \end{array}$$

Again, by using Sage B.1, we get the following Jacobi sums with the corresponding prime ideal factorization,

$$\begin{array}{lr} J_1 = 4x^7+2x^6+2x^5-7x^4-2x^3+2x^2-4x & P_{97,7}P_{97,-5}P_{97,-3}P_{97,1} \\ J_2 = -5x^7+6x^6+x^5+x^4-x^3+4x^2+x+4 & P_{97,-7}P_{97,-5}P_{97,-3}P_{97,1} \\ J_3 = x^7+6x^6+x^5+4x^4+x^3-4x^2+5x+1 & P_{97,7}P_{97,-5}P_{97,3}P_{97,1} \\ J_4 = x^7-4x^6-x^5-4x^4-5x^3+6x^2+x+1 & P_{97,-7}P_{97,-5}P_{97,-3}P_{97,1} \\ J_5 = x^7-6x^6+x^5+x^4-5x^3-4x^2-x+4 & P_{97,7}P_{97,5}P_{97,-3}P_{97,1} \\ J_6 = -6x^6+5x^4+6x^2 & P_{97,-7}P_{97,-5}P_{97,3}P_{97,1} \\ J_7 = -2x^7+2x^6-4x^5-4x^3-2x^2-2x+7 & P_{97,7}P_{97,-5}P_{97,-3}P_{97,1}. \end{array}$$

Now we can find the generators for the prime ideals by looking at the prime ideal factorizations of the Jacobi sums, divide them and take the square root. We start by finding the generator for $P_{97,7}$

$$\begin{array}{lr} J_1/J_2 = \frac{1}{97}(41x^7+60x^6+5x^5-40x^4+29x^3-38x^2+13x-7) & P_{97,7}/P_{97,-7} \\ g_7 \cdot J_1/J_2 = -x^7-2x^6-2x^5-x^4+2x^3+x^2-2 & P_{97,7}^2 \\ (u_0u_5 \cdot g_7 \cdot J_1/J_2)^{1/2} = x^7-x^5-x^4+2x^2 & P_{97,7}. \end{array}$$

And for $P_{97,-7}$ we get the generator

$$\sigma_{-1}(x^7-x^5-x^4+2x^2) = x^{-7}-x^{-5}-x^{-4}+2x^{-2} = -x+x^3+x^4-2x^6.$$

Next, we find the generator for $P_{97,5}$,

$$\begin{array}{lr} J_5/J_1 = \frac{1}{97}(-5x^7-38x^6+41x^5+40x^4+13x^3+60x^2-29x-7) & P_{97,5}/P_{97,-5} \\ g_5 \cdot J_5/J_1 = -x^7-x^6+2x^5+3x^4-x^3-x^2-x+1 & P_{97,5}^2 \\ (u_0u_1u_5 \cdot g_5 \cdot J_5/J_1)^{1/2} = x^6+x^5-2x^3-x^2+2 & P_{97,5}. \end{array}$$

Which also gives the following generator for $P_{97,-5}$,

$$\sigma_{-1}(x^6 + x^5 - 2x^3 - x^2 + 2) = -x^2 - x^3 + 2x^5 + x^6 + 2.$$

For $P_{97,3}$ we get,

$$\begin{array}{lr}
J_3/J_1 = \frac{1}{97}(-41x^7 - 38x^6 + 5x^5 + 7x^4 - 29x^3 - 60x^2 + 13x - 40) & P_{97,3}/P_{97,-3} \\
g_3 \cdot J_3/J_1 = -x^7 - x^5 + 2x^4 + x^3 + x^2 + 3x & P_{97,3}^2 \\
(u_5 u_7 \cdot g_3 \cdot J_3/J_1)^{1/2} = x^5 - x^4 + 2x^3 - x^2 & P_{97,3}.
\end{array}$$

And for $P_{97,-3}$ we get,

$$\sigma_{-1}(x^5 - x^4 + 2x^3 - x^2) = -x^3 + x^4 - 2x^5 + x^6$$

Lastly, we need to find the generators for $P_{97,1}$ and $P_{97,-1}$, by using the following elements,

$$\begin{array}{ll}
(g_1^2 \cdot g_3 \cdot g_5 \cdot g_7)/(J_5 \cdot J_6) & P_{97,-1}^2 \\
(u_3 \cdot \frac{g_1^2 \cdot g_3 \cdot g_5 \cdot g_7}{J_5 \cdot J_6})^{1/2} = x^7 + 2x^6 + 2x^5 + x^4 + x^3 & P_{97,-1}
\end{array}$$

and for $P_{97,1}$ we get,

$$\sigma_{-1}(x^7 + 2x^6 + 2x^5 + x^4 + x^3) = -x - 2x^2 - 2x^3 - x^4 - x^5.$$

Appendix B

Sage documentation

B.1 Prime Ideal Generators

```
K.<x> = CyclotomicField(8)
UK = UnitGroup(K)
S = K.ideal(17).prime_factors()
US = UnitGroup(K,S=tuple(S))

#Finding all the prime ideals bounded by 17
K.primes_of_bounded_norm(17)

#Finding a generator for the prime ideals
US.gens_values()
```

Listing B.1: Prime ideal generators

```
m=4
p = 13

K.<x> = CyclotomicField(m)

G = DirichletGroup(p,CyclotomicField(m))
e = G([x]) #defining the conductor to be the root of unity

e.gauss_sum(1) #calculating the Gauss sum with expinent 1 for the additative character
abs(e.gauss_sum(1))^2 #checking the absolute avlue
```

Listing B.2: Gauss sum

```
m = 8   #Degree of cyclotomic field
p = 17  #Prime ideal for Gauss/Jacobi sum

K.<x> = CyclotomicField(m)  #Defining the cyclotomic field

#Defining the character for given prime p
G = DirichletGroup(p,CyclotomicField(m)).0

n=1              #Setting the exponent for the Jacobi sum
J_n = G.jacobi_sum(G^n) #Calculating the Jacobi sum over the character


J_n   #Printing the Jacobi sum
```

Listing B.3: Jacobi sum

```
K.<x> = CyclotomicField(16)

#listing the Jacobi sums
zeta16 = x
J1 = 2*zeta16^7 + 2*zeta16^6 - zeta16^4 + 2*zeta16^2 - 2*zeta16
J2 = zeta16^7 - 2*zeta16^6 - 3*zeta16^5 + zeta16^4 - zeta16^3 - zeta16
J3 = zeta16^7 + 2*zeta16^6 - zeta16^5 + 3*zeta16^3 + zeta16 - 1
J4 = zeta16^7 + zeta16^5 + zeta16^3 - 2*zeta16^2 - 3*zeta16 + 1
J5 = -zeta16^7 - 2*zeta16^6 + zeta16^5 - zeta16^4 - zeta16^3 - 3*zeta16
J6 = -2*zeta16^6 - 3*zeta16^4 + 2*zeta16^2
J7 = 2*zeta16^6 - 2*zeta16^5 - 2*zeta16^3 - 2*zeta16^2 + 1

#listing the generetors for P_cP_-c
g7 = (x^7 - x^5 + x^4 + x^2 + 1)
g5 = (-x^7 - x^6 + x^5 - x + 1)
g3 = (x^7 + x^6 - x^2 - x + 1)
g1 = (-x^6 - x^4 - x^3 + x + 1)

#listing the units
u0 = x
u1 = 1+x+x^(-1)
u3 = 1+x^3+x^(-3)
u5 = 1+x^5+x^(-5)
u7 = 1+x^7+x^(-7)


K.ideal((17^2,17*(3-x^(-1)),17*(3-x^(1)),(3-x^(1))*(3-x^(-1)))).gens_reduced() #finding generators for P_cP_-c

K.ideal(J1).factors()     #finding the prime ideal factorization of the polynomials of the Jacobi sums

sqrt((u5*g7*J1)/(J2))     #finding the square roots
```

Listing B.4: Finding prime ideal generators

B.2 Key Generator for NTRU

```
K.<x> = CyclotomicField(16)
q = 41

#computing the secret key
f = x^6-x^4+x^3+x^2 - 1
fq = inverse_mod(f, q) #finding inverse of f(x) mod q
g = x^7-x^5+x^3- x

#computing the public key
h = (fq*g).mod(q)

#checking their factorization
K.ideal(f).factor(), K.ideal(fq).factor(), K.ideal(g).factor(), K.ideal(h).factor()
```

Listing B.5: Key generator for NTRU

B.3 LLL-algorithm

```
#The start basis
w1 = vector([56,-31,-4,-7])
w2 = vector([28.0, 32.0, 151.0, 58.0])
w3 = vector([66, 163, -122, 15])
w4 = vector([-18, -69, -6, 190])

#The new basis
v1 = vector([56,-31,-4,-7])
v2 = vector([28, 32, 151, 58])
v3 = vector([66, 163, -122, 15])
v4 = vector([-18, -69, -6, 190])

y = vector([52.43, -32.51,-2.39,132.48])

#Matrix for solving CVP
M = matrix([v1,v2,v3,v4])
t = M.solve_left(y)

#Calculating the Gram-Schmidt basis
vv2 = w2 - ((w2*w1)/float(w1.norm()^2))*w1
vv3 = w3 - ((w3*w1)/float(w1.norm()^2))*w1 - ((w3*vv2)/float(vv2.norm()^2))*w2
vv4 = w4 - ((w4*w1)/float(w1.norm()^2))*w1 - ((w4*vv2)/float(vv2.norm()^2))*w2 - ((w4*vv3)/float(vv3.norm()^2))*w3

#Calculating the projection factor
my21 = (w2*w1)/float(w1.norm()^2)
my31 = (w3*w1)/float(w1.norm()^2)
my32 = (w3*vv2)/float(vv2.norm()^2)
my41 = (w4*w1)/float(w1.norm()^2)
my42 = (w4*vv2)/float(vv2.norm()^2)
my43 = (w4*vv3)/float(vv3.norm()^2)

#Calculating the new basis
v21 = w2 - round(my21,0)*w1
v31 = w3 - round(my31,0)*w1
v32 = w3 - round(my32,0)*w2
v41 = w4 - round(my41,0)*w1
v42 = w4 - round(my42,0)*w2
v43 = w4 - round(my43,0)*w3

#Checing the size
w2.norm()^2, (3/4 - my21^2)*w1.norm()^2
w3.norm()^2, (3/4 - my32^2)*vv2.norm()^2
w4.norm()^2, (3/4 - my43^2)*w3.norm()^2

#The closest vector
y2 = v1 + v4
```

Listing B.6: Example 4.2.1 for SVP and CVP

```
from sage.modules.free_module_integer import IntegerLattice

#public key h(x)
h = -7*x^7 - 10*x^6 + 5*x^5 + 5*x^4 + 20*x^3 - 19*x^2 + 3*x - 12
#row vectors for the NTRU matrix
v1 = vector([1, 0, 0, 0, 0, 0, 0, 0, -12, 3, -19, 20, 5, 5, -10, -7])
v2 = vector([0, 1, 0, 0, 0, 0, 0, 0, 7, -12, 3, -19, 20, 5, 5, -10])
v3 = vector([0, 0, 1, 0, 0, 0, 0, 0, 10, 7, -12, 3, -19, 20, 5, 5])
v4 = vector([0, 0, 0, 1, 0, 0, 0, 0, -5, 10, 7, -12, 3, -19, 20, 5])
v5 = vector([0, 0, 0, 0, 1, 0, 0, 0,  -5, -5, 10, 7, -12, 3, -19, 20])
v6 = vector([0, 0, 0, 0, 0, 1, 0, 0, -20, -5, -5, 10, 7, -12, 3, -19])
v7 = vector([0, 0, 0, 0, 0, 0, 1, 0, 19, -20, -5, -5, 10, 7, -12, 3])
v8 = vector([0, 0, 0, 0, 0, 0, 0, 1, -3, 19, -20, -5, -5, 10, 7, -12])
v9 = vector([0,0,0,0,0,0,0,0,41,0,0,0,0,0,0,0])
v10 = vector([0,0,0,0,0,0,0,0,0,41,0,0,0,0,0,0])
v11 = vector([0,0,0,0,0,0,0,0,0,0,41,0,0,0,0,0])
v12 = vector([0,0,0,0,0,0,0,0,0,0,0,41,0,0,0,0])
v13 = vector([0,0,0,0,0,0,0,0,0,0,0,0,41,0,0,0])
v14 = vector([0,0,0,0,0,0,0,0,0,0,0,0,0,41,0,0])
v15 = vector([0,0,0,0,0,0,0,0,0,0,0,0,0,0,41,0])
v16 = vector([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,41])

#making the NTRU matrix
B = matrix([v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16])

L = IntegerLattice(B, lll_reduce=False) #row reduction on the matrix

L.LLL().str() #printing the reduced row vectors
L.shortest_vector(algorithm="pari") #printing the vector with the smallest size
```

Listing B.7: Example of applying LLL algorithm to NTRU

B.4 Unit Attack

```
K.<x> = CyclotomicField(16)
R = K.ring_of_integers()
E = R.random_element()
G = K.galois_group() #The Galois group used in the Log-embedding

#Listing up the multiplicative independent units
u1 = (1+x+x^(-1))
u3 = (1+x^3+x^(-3))
u5 = (1+x^5+x^(-5))
u7 = (1+x^7+x^(-7))

#The element to embedd
a =  -11*x^7 - 3*x^6 + 3*x^5 + 3*x^4 - 2*x^3 - 6*x^2 + x + 7

#Defining the different coordinates in the embedding
v0 = float(log(abs(a*conjugate(a))))
v1 = float(log(abs(G[3](a)*conjugate(G[3](a)))))
v2 = float(log(abs(G[1](a)*conjugate(G[1](a)))))
v3 = float(log(abs(G[2](a)*conjugate(G[2](a)))))

b = vector([v0,v1,v2,v3]) #The general embedding of an element

#Defining the embeddings for the Log-unit lattice
```

```
b1 = vector([2.093064784031127, 1.1367170483150637, -2.8994642223541525,
    -0.33031760999204085])
b2 = vector([1.1367170483150637, -0.33031760999204085, 2.093064784031127,
     -2.8994642223541525])
b3 = vector([-2.8994642223541525, 2.093064784031127,
    -0.33031760999204085, 1.1367170483150637])
b4 = vector([1,1,1,1])


M = matrix([b1,b2,b3,b4]) #Log-unit matrix
t = M.solve_left(b) #Equation to find a unit close to generator

#Making the elements as vectors to calculate the size
v = vector([-11*x^7,-3*x^6,3*x^5, 3*x^4,-2*x^3,-6*x^2,x,7])
w = vector([-2*x^7,-2*x^6,-2*x^5, x^4,-2*x^2,3*x,2])

#Listing the results
g1 = -11*x^7 - 3*x^6 + 3*x^5 + 3*x^4 - 2*x^3 - 6*x^2 + x + 7
t1 = (2,0,1)
g2 = g1/(u1^2*u5)

v.norm(), w.norm() #Checking the size
```

Listing B.8: Unit attack for example 5.2.1

B.5 S-unit Attack

```
#defining the cyclotomic field
K.<x> = CyclotomicField(8)

#calculating the prime ideal factorization
A3 = K.ideal(3).factor()
A5 = K.ideal(5).factor()
A7 = K.ideal(7).factor()
A11 = K.ideal(11).factor()
A13 = K.ideal(13).factor()
A17 = K.ideal(17).factor()

#checking the norm of each prime ideal
A3[0][0].norm(), A3[1][0].norm()
A5[0][0].norm(), A5[1][0].norm()
A7[0][0].norm(), A7[1][0].norm()
A11[0][0].norm(), A11[1][0].norm()
A13[0][0].norm(), A13[1][0].norm()
A17[0][0].norm(), A17[1][0].norm(), A17[2][0].norm(), A17[3][0].norm()
```

Listing B.9: finding prime ideals for example 5.3.1

```
K.<x> = CyclotomicField(8)
R = K.ring_of_integers()
E = R.random_element()
G = K.galois_group() #The Galois group used in the Log-embedding

#Listing up the multiplicative independent units
u1 = (1+x+x^(-1))
u3 = (1+x^3+x^(-3))
P31 = x^2+x-1
P32 = -x^3-x^2-1
P51 = (x^2-2)
```

```
P52 = (x^2+2)
P171 = 2-x
P172 = 2-x^3
P173 = 2+x^3
P174 = 2+x

a = -25*x^3 - 25*x - 15 #The element to embedd

#Defining the different coordinates in the embedding
v1 = float(log(abs(a*conjugate(a))))
v2 = float(log(abs(G[3](a)*conjugate(G[3](a)))))
v3 = float(log(P31.norm()^(-(K.valuation(P31)(a)))))
v4 = float(log(P32.norm()^(-(K.valuation(P32)(a)))))
v5 = float(log(P51.norm()^(-(K.valuation(P51)(a)))))
v6 = float(log(P52.norm()^(-(K.valuation(P52)(a)))))
v7 = float(log(P171.norm()^(-(K.valuation(P171)(a)))))
v8 = float(log(P172.norm()^(-(K.valuation(P172)(a)))))
v9 = float(log(P173.norm()^(-(K.valuation(P173)(a)))))
v10 = float(log(P174.norm()^(-(K.valuation(P174)(a)))))


b = vector([v1,v2,v3,v4,v5,v6,v7,v8,v9,v10]) #The general embedding of an element

#Defining the embeddings for the Log-unit lattice
b1 = vector([1.762747174039086, -1.7627471740390845, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b2 = vector([1.0986122886681098, 1.0986122886681098, -2.1972245773362196, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b3 = vector([1.0986122886681098, 1.0986122886681098, 0.0, -2.1972245773362196, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b4 = vector([1.6094379124341003, 1.6094379124341003, 0.0, 0.0, -3.2188758248682006, 0.0, 0.0, 0.0, 0.0, 0.0])
b5 = vector([1.6094379124341003, 1.6094379124341003, 0.0, 0.0, 0.0, -3.2188758248682006, 0.0, 0.0, 0.0, 0.0])
b6 = vector([0.7754517322978262, 2.05776161175839, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0])
b7 = vector([2.05776161175839, 0.7754517322978262, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0])
b8 = vector([0.7754517322978262, 2.05776161175839, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0])
b9 = vector([2.05776161175839, 0.7754517322978262, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216])
b10 = vector([1,1,1,1,1,1,1,1,1,1])


M = matrix([b1,b2,b3,b4,b5,b6,b7,b8,b9,b10]) #Log-unit matrix
t = M.solve_left(b) #Equation to find a unit close to generator

#Making the elements as vectors to calculate the size
w0 = vector([-5*x^2, -3*x, 5])
w1 = vector([-5*x^3, 115*x^2, 5*x])
w2 = vector([-25*x^3, - 25*x, - 15])


A = R.ideal(-5*x^2-3*x+5)
#Choosing a random element form the ideal
R.random_element()
#The random element we want to reduce
g1 = -5*x^3 + 115*x^2 + 5*x

```

```
#Listing the results
t1 = (0,2,0,0,0,0,0)
g2 = g1/(P31^2)
t2 = (0,0,0,1,1,0,0,0,0)
g3 = g2/(P51*P52)

w0.norm(), w1.norm(), w2.norm() #Checking the size
```

Listing B.10: S-unit attack for example 5.3.1

```

#Defining the cyclotomic field
K.<x> = CyclotomicField(16)

#Defining the unit and S-unit group
UK = UnitGroup(K)
S = K.ideal(21).prime_factors()
US = UnitGroup(K,S=tuple(S))

#Listing all the prime ideals with norm less than or equal to 97
K.primes_of_bounded_norm(97)

#Finding generators for small prime ideals
US.gens_values()

#Listing the non-principle prime ideals
A1 = K.ideal(97,8-x)
A2 = K.ideal(97,8+x^7)

A3 = K.ideal(97,8-x^3)
A4 = K.ideal(97,8+x^5)

A5 = K.ideal(97,8-x^5)
A6 = K.ideal(97,8+x^3)

A7 = K.ideal(97,8-x^7)
A8 = K.ideal(97,8+x)

#Finding generators for the principle ideals
K.ideal(A1*A2).gens_reduced()
```

Listing B.11: Finding the prime ideals for example 5.3.2

```
K.<x> = CyclotomicField(16)
R = K.ring_of_integers()
E = R.random_element()
G = K.galois_group() #The Galois group used in the Log-embedding


#Listing the generators for the S-unit group
u1 = (1+x+x^(-1))
u3 = (1+x^3+x^(-3))
u5 = (1+x^5+x^(-5))
u7 = (1+x^7+x^(-7))
P21 = (x+1)
P31 = (x^4 + x^2 - 1)
P32 = (x^4 - x^2 - 1)
P71 = x^2 + x - 1
P72 = x^2 - x - 1
P73 = x^5 + x^3 + 1
P74 = x^5 + x^3 - 1
```

```
P171 = (- x^4 - x - 1)
P172 = (x^7 + x^4 - 1)
P173 = (x^7 + x^3 - x^2)
P174 = (x^6-x^5-x)
P175 = (x^7-x^4-x^3)
P176 = (x^5+x^4-x)
P177 = (x^6-x^2+x)
P178 = (x^4-x+1)
P971 = x^4+x^3+2*x^2+2*x+1
P972 = x^4+2*x^3+2*x^2+x+1
P973 = x^3 - x^2 + 2*x - 1
P974 = x^3-2*x^2+x-1
P975 = x^6 + x^5 - 2*x^3 - x^2 + 2
P976 = x^6+2*x^5-x^3-x^2+2
P977 = x^5 - x^3 - x^2 + 2
P978 = -2*x^5+x^3+x^2-1

#The element to embedd
a = 17*x^7 - 6*x^6 + 3*x^5 + 17*x^4 - 15*x^3 + 2*x^2 - 17*x + 13

#Defining the different coordinates in the embedding
v1 = float(log(abs(a*conjugate(a))))
v2 = float(log(abs(G[3](a)*conjugate(G[3](a)))))
v3 = float(log(abs(G[1](a)*conjugate(G[1](a)))))
v4 = float(log(abs(G[2](a)*conjugate(G[2](a)))))
v5 = float(log(P31.norm()^(-(K.valuation(P31)(a)))))
v6 = float(log(P32.norm()^(-(K.valuation(P32)(a)))))
v7 = float(log(P71.norm()^(-(K.valuation(P71)(a)))))
v8 = float(log(P72.norm()^(-(K.valuation(P72)(a)))))
v9 = float(log(P73.norm()^(-(K.valuation(P73)(a)))))
v10 = float(log(P74.norm()^(-(K.valuation(P74)(a)))))
v11 = float(log(P171.norm()^(-(K.valuation(P171)(a)))))
v12 = float(log(P172.norm()^(-(K.valuation(P172)(a)))))
v13 = float(log(P173.norm()^(-(K.valuation(P173)(a)))))
v14 = float(log(P174.norm()^(-(K.valuation(P174)(a)))))
v15 = float(log(P175.norm()^(-(K.valuation(P175)(a)))))
v16 = float(log(P176.norm()^(-(K.valuation(P176)(a)))))
v17 = float(log(P177.norm()^(-(K.valuation(P177)(a)))))
v18 = float(log(P178.norm()^(-(K.valuation(P178)(a)))))
v19 = float(log(P971.norm()^(-(K.valuation(P971)(a)))))
v20 = float(log(P972.norm()^(-(K.valuation(P972)(a)))))
v21 = float(log(P973.norm()^(-(K.valuation(P973)(a)))))
v22 = float(log(P974.norm()^(-(K.valuation(P974)(a)))))
v23 = float(log(P975.norm()^(-(K.valuation(P975)(a)))))
v24 = float(log(P976.norm()^(-(K.valuation(P976)(a)))))
v25 = float(log(P977.norm()^(-(K.valuation(P977)(a)))))
v26 = float(log(P978.norm()^(-(K.valuation(P978)(a)))))

#The general embedding of an element
b = vector([v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,
    v18,v19,v20,v21,v22,v23,v24,v25,v26])

#Defining the embeddings for the Log-unit lattice
b1 = vector([2.093064784031127, 1.1367170483150637, -2.8994642223541525,
    -0.33031760999204085, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b2 = vector([1.1367170483150637, -0.33031760999204085, 2.093064784031127,
     -2.8994642223541525, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b3 = vector([-2.8994642223541525, 2.093064784031127,
    -0.33031760999204085, 1.1367170483150637, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0])
b4 = vector([1.0986122886681098, 1.0986122886681098, 1.0986122886681098,
    1.0986122886681098, -4.394449154672439, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b5 = vector([1.0986122886681098, 1.0986122886681098, 1.0986122886681098,
    1.0986122886681098, 0.0, -4.394449154672439, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b6 = vector([0.4610804594339829, 1.4848296896213304, 1.4848296896213304,
    0.4610804594339829, 0.0, 0.0, -3.891820298110627, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b7 = vector([0.4610804594339829, 1.4848296896213304, 1.4848296896213304,
    0.4610804594339829, 0.0, 0.0, 0.0, -3.891820298110627, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b8 = vector([1.4848296896213304, 0.4610804594339829, 0.4610804594339829,
    1.4848296896213304, 0.0, 0.0, 0.0, 0.0, -3.891820298110627, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b9 = vector([1.4848296896213304, 0.4610804594339829, 0.4610804594339829,
    1.4848296896213304, 0.0, 0.0, 0.0, 0.0, 0.0, -3.891820298110627, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0])
b10 = vector([1.7251077710821243, 0.6510784715617671, 1.4066831401966229,
     -0.9496560387842973, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0])
b11 = vector([1.7251077710821243, 0.6510784715617671, 1.4066831401966229,
     -0.9496560387842973, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0])
b12 = vector([0.6510784715617671, -0.9496560387842973,
    1.7251077710821243, 1.4066831401966229, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0])
b13 = vector([0.6510784715617671, -0.9496560387842973,
    1.7251077710821243, 1.4066831401966229, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0])
b14 = vector([1.4066831401966229, 1.7251077710821243,
    -0.9496560387842973, 0.6510784715617671, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0, 0.0, 0.0, 0.0, 0.0])
b15 = vector([1.4066831401966229, 1.7251077710821243,
    -0.9496560387842973, 0.6510784715617671, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0, 0.0, 0.0, 0.0, 0.0])
b16 = vector([-0.9496560387842973, 1.4066831401966229,
    0.6510784715617671, 1.7251077710821243, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0])
b17 = vector([-0.9496560387842973, 1.4066831401966229,
    0.6510784715617671, 1.7251077710821243, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0])
b18 = vector([3.648349436590895, 1.350600656203053, 0.15674220884828047,
     -0.5809813231380389, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, -4.574710978503383, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0])
b19 = vector([3.6483494365900895, 1.350600656203053, 0.15674220884828047,
     -0.5809813231380389, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -4.574710978503383, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0])
b20 = vector([0.21388360788798957, -0.2506637131459986,
    1.555284652558963, 3.0562064312024293, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -4.574710978503383,
    0.0, 0.0, 0.0, 0.0, 0.0])
b21 = vector([0.21388360788798957, -0.2506637131459986,
    1.555284652558963, 3.0562064312024293, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -4.574710978503383, 0.0, 0.0, 0.0, 0.0])
b22 = vector([0.15674220884828047, 3.6483494365900895,
    -0.5809813231380389, 1.350600656203053, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -4.574710978503383, 0.0, 0.0, 0.0])
b23 = vector([0.15674220884828047, 3.6483494365900895,
    -0.5809813231380389, 1.350600656203053, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -4.574710978503383, 0.0, 0.0])
b24 = vector([-0.2506637131459986, 3.0562064312024293,
    0.21388360788798957, 1.555284652558963, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -4.574710978503383, 0.0])
b25 = vector([-0.2506637131459986, 3.0562064312024293,
    0.21388360788798957, 1.555284652558963, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, -4.574710978503383])
b26 = vector([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1])


M = matrix([b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17,
    b18,b19,b20,b21,b22,b23,b24,b25,b26]) #The S-unit matrix
t = M.solve_left(b) #Equation to find a S-unit close to generator

#Defining the general ideal with the unknown generator
A = K.ideal(x^6 + x^4 - x^2 + x - 1)
R = A.random_element() #Picking elements from the ideal at random

#Defining the elements as vectors to check the size
w1 = vector([x^6, + 3*x^5, + x^4, - 8*x^3, + 2*x^2, - x, + 199])
w2 = vector([3*x^7, + x^5, - x^3, + x, + 94])
w3 = vector([-136*x^7, - 397*x^6, - 206*x^5, + 181*x^4, + 435*x^3, + 174*
    x^2, - 104*x, - 158])

#Listing the random elements from the ideal
a1 = -x^6 + 3*x^5 + x^4 - 8*x^3 + 2*x^2 - x + 199
a2 = 3*x^7 + x^5 - x^3 + x + 94
a3 = -136*x^7 - 397*x^6 - 206*x^5 + 181*x^4 + 435*x^3 + 174*x^2 - 104*x -
     158

#Listing the results
t1 = (-1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0)
g1 = a1/(u1^(-1)*u5^(-1))
t2 = (-1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
     0, 0, 0)
g2 = a2/(u1^(-1)*u5^(-1)*P175*P177)
t3 = (-1, 1, 0, 2, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 3, 0, 1, 0, 0, 0, 0, 0,
    0, 0, 0)
```

```
g3 = a3/(u1^(-1)*u3*P31*P71*P74^3*P175*P176^3*P178)

K.ideal(a1).factor() #Checking the factorization of the elements

w1.norm(), w2.norm(), w3.norm() #Cheking the sizes
```

Listing B.12: S-unit attack for example 5.3.2

```

K.<x> = CyclotomicField(16)
R = K.ring_of_integers()
E = R.random_element()
G = K.galois_group() #The Galois group used in the Log-embedding


#Listing up the generators for the S-unit group
u1 = (1+x+x^(-1))
u3 = (1+x^3+x^(-3))
u5 = (1+x^5+x^(-5))
u7 = (1+x^7+x^(-7))
P171 = (- x^4 - x - 1)
P172 = (x^7 + x^4 - 1)
P173 = (x^7 + x^3 - x^2)
P174 = (x^6-x^5-x)
P175 = (x^7-x^4-x^3)
P176 = (x^5+x^4-x)
P177 = (x^6-x^2+x)
P178 = (x^4-x+1)

#The element to embedd
a = -7*x^7 - 10*x^6 + 5*x^5 + 5*x^4 + 20*x^3 - 19*x^2 + 3*x - 12

#Defining the different coordinates in the embedding
v1 = float(log(abs(a*conjugate(a))))
v2 = float(log(abs(G[3](a)*conjugate(G[3](a)))))
v3 = float(log(abs(G[1](a)*conjugate(G[1](a)))))
v4 = float(log(abs(G[2](a)*conjugate(G[2](a)))))
v5 = float(log(P171.norm()^(-(K.valuation(P171)(a)))))
v6 = float(log(P172.norm()^(-(K.valuation(P172)(a)))))
v7 = float(log(P173.norm()^(-(K.valuation(P173)(a)))))
v8 = float(log(P174.norm()^(-(K.valuation(P174)(a)))))
v9 = float(log(P175.norm()^(-(K.valuation(P175)(a)))))
v10 = float(log(P176.norm()^(-(K.valuation(P176)(a)))))
v11 = float(log(P177.norm()^(-(K.valuation(P177)(a)))))
v12 = float(log(P178.norm()^(-(K.valuation(P178)(a)))))

#The general embedding of an element
b = vector([v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12])

#Defining the embeddings for the Log-unit lattice
b1 = vector([2.093064784031127, 1.1367170483150637, -2.8994642223541525,
    -0.33031760999204085, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b2 = vector([1.1367170483150637, -0.33031760999204085, 2.093064784031127,
     -2.8994642223541525, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
b3 = vector([-2.8994642223541525, 2.093064784031127,
    -0.33031760999204085, 1.1367170483150637, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0])
b4 = vector([1.7251077710821243, 0.6510784715617671, 1.4066831401966229,
    -0.9496560387842973, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0])
b5 = vector([1.7251077710821243, 0.6510784715617671, 1.4066831401966229,
    -0.9496560387842973, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0, 0.0,
     0.0])
b6 = vector([0.6510784715617671, -0.9496560387842973, 1.7251077710821243,
     1.4066831401966229, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0, 0.0,
     0.0])
b7 = vector([0.6510784715617671, -0.9496560387842973, 1.7251077710821243,
     1.4066831401966229, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0, 0.0,
     0.0])
b8 = vector([1.4066831401966229, 1.7251077710821243, -0.9496560387842973,
     0.6510784715617671, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0, 0.0,
     0.0])
b9 = vector([1.4066831401966229, 1.7251077710821243, -0.9496560387842973,
     0.6510784715617671, 0.0, 0.0, 0.0, 0.0, 0.0, -2.833213344056216, 0.0,
     0.0])
b10 = vector([-0.9496560387842973, 1.4066831401966229,
    0.6510784715617671, 1.7251077710821243, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    -2.833213344056216, 0.0])
b11 = vector([-0.9496560387842973, 1.4066831401966229,
    0.6510784715617671, 1.7251077710821243, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, -2.833213344056216])
b12 = vector([1,1,1,1,1,1,1,1,1,1,1,1])


M = matrix([b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12]) #S-unit matrix
t = M.solve_left(b) #Equation to S-find a unit close to generator


f = x^6-x^4+x^3+x^2 - 1
fp = inverse_mod(f, 3)
g = x^6+x^4-x^2- x

h = -7*x^7 - 10*x^6 + 5*x^5 + 5*x^4 + 20*x^3 - 19*x^2 + 3*x - 12

f1 = (u3^(-1)*P172*P178).mod(41)
f2 = (P172*P178).mod(41)

g1 = (f2*h).mod(41)

K.ideal(h).factor()
```

Listing B.13: S-unit attack on NTRU example 5.4.1