CRYSTALS-KYBER
Analysis of the post-quantum cryptosystem

Simon Pohmann

April 26, 2020

Abstract

Crystals Kyber is a post-quantum public-private key cryptosystem that is currently a candidate in the second round of the NIST post-quantum standardization process. The security of this scheme relies on the so-called Learning with Errors problem, which can be described as solving a system of linear equation that have been perturbed by some small error. It has been shown that this problem has a strong connection to lattice problems and that its hardness can be quantumly reduced to worst case SIVP. Additionally, there are variants of LWE that introduce more algebraic structure to the problem and therefore yield more efficient cryptographic schemes. Crystals Kyber itself is based on the module LWE problem, which is LWE in modules over the ring of integers in cyclotomic number fields. In recent years, proofs have been published that show a similar connection to worst case problems in a corresponding subclass of lattices with additional structure.

In this work, we describe standard LWE, the ring LWE variant and module LWE with their hardness reductions and explain how their structure differs. In particular, we combine the known proofs to a complete quantum reduction from decision Module LWE to worst case SIVP in the corresponding class of lattices. Additionally, we include a description of the Crystals Kyber cryptosystem and an optimized implementation written in Rust.

Contents

Chapter 1

Introduction

Public-key cryptography has become a fundamental part of the Internet and many software systems. Currently, almost all schemes used in practice are some variant of the RSA scheme, relying on the difficulty of factoring large numbers. However, in 1994 Peter Shor discovered an algorithm that can efficiently perform such a factorization, and therefore break most existing public-key cryptosystems - when running on a quantum computer. At that time, such a device was only a theoretical concept, but in the last decade, much progress was made on building it. For example, scientists at Google [Aru+19] have constructed a quantum computer that, while still useless for real problems, vastly outperforms any modern supercomputer on some artificial toy problems. Given this development, there is a lot of interest in the development of cryptographic procedures that are secure against attackers with access to quantum computation. The most promising approaches are currently systems based on lattices, multivariate polynomials and linear codes. In this work, we will have a closer look on the first of these approaches. One singular advantage of lattice cryptography compared of other approaches is that their security is based on the worst-case hardness of certain problems.

In April 2016 for example, the National Institute of Standards and Technology (NIST) started the Post-Quantum Cryptography Standardization, a competition to find secure public-key and signature schemes. The first round ended in January 2019, with 26 proposed schemes accepted as candidates for the second round ([Ala+19]). Among them are 9 lattice-based public-key systems. In this work, we will analyze the Crystals-Kyber cryptosystem that is based on the Module Learning with Errors problem. Crystals-Kyber was chosen, as its connection to the underlying problem is easy to see. Additionally, most lattice-based approaches rely on some variant of Learning with Errors (the only used alternative being NTRU) and Module Learning with Errors is the most general one, so this analysis should convey a good concept of general lattice-based approaches to public-key schemes. We will also focus mostly on the underlying problem and the security proofs, as they are universal for all concrete systems. Nevertheless, we give a short summary of the implementation detail, and provide an implementation written in Rust, too.

Chapter 2

Basics of Lattices

Convention In this work, an expression a/bc is $\frac{a}{bc}$ and not $\frac{a}{b}c$. For a probability distribution χ on X, we denote by χ^k the probability distribution on X^k of vectors $x \in X^k$ where the components $x_1, ..., x_k$ are iid. and distributed according to χ.

2.1 Motivation

The security of Crystals-Kyber and most other lattice cryptosystems is based on the worst-case hardness of some well-understood problems in lattices. A lattice is a discrete subgroup of the whole euclidean space, so it consists of all integral linear combinations of some vector space basis (see figure 2.1). Such a lattice basis is not unique, and this is what makes lattices well suited for asymmetric cryptography. Lattices are usually considered together with a norm from the containing vector space, and the length of the basis vectors in multiple bases can be very different, as well as the angle between different basis vectors. The key is now that it is conjectured to be very hard to find a "good" basis (with short vectors) when given a "bad" basis with long vectors. If you have access to a "good" basis, many problems in the lattice can be solved efficiently, like calculating the shortest vector or finding the nearest lattice point to a given point in space. With a "bad" basis however, these problems are considered extremely hard, even for quantum computers and even if only a polynomial approximation is required. These problems are also of universal importance, as many interesting problems can be reduced to lattice problems. In this section, we will introduce some basic notions and the most important "hard" computational problems in lattices on which our security assumptions are based.

Interestingly, lattices are not only useful for cryptography, but they have numerous applications in number theory, and Minkowski proved some results on the structure of lattices already around 1900. We slightly touch one of these applications, the Minkowski-Space, later when we talk about Ring Learning with Errors.

2.2 Definition (Lattice)

A subset $L \subseteq \mathbb{R}^n$ with $n \in \mathbb{N}$ is called a lattice, if there exist linearly independent vectors $v_1, ..., v_t \in \mathbb{R}^n$ so that $L = v_1\mathbb{Z} + ... + v_t\mathbb{Z}$. If $t = n$, the lattice is called full-rank. n is called the dimension of the lattice.

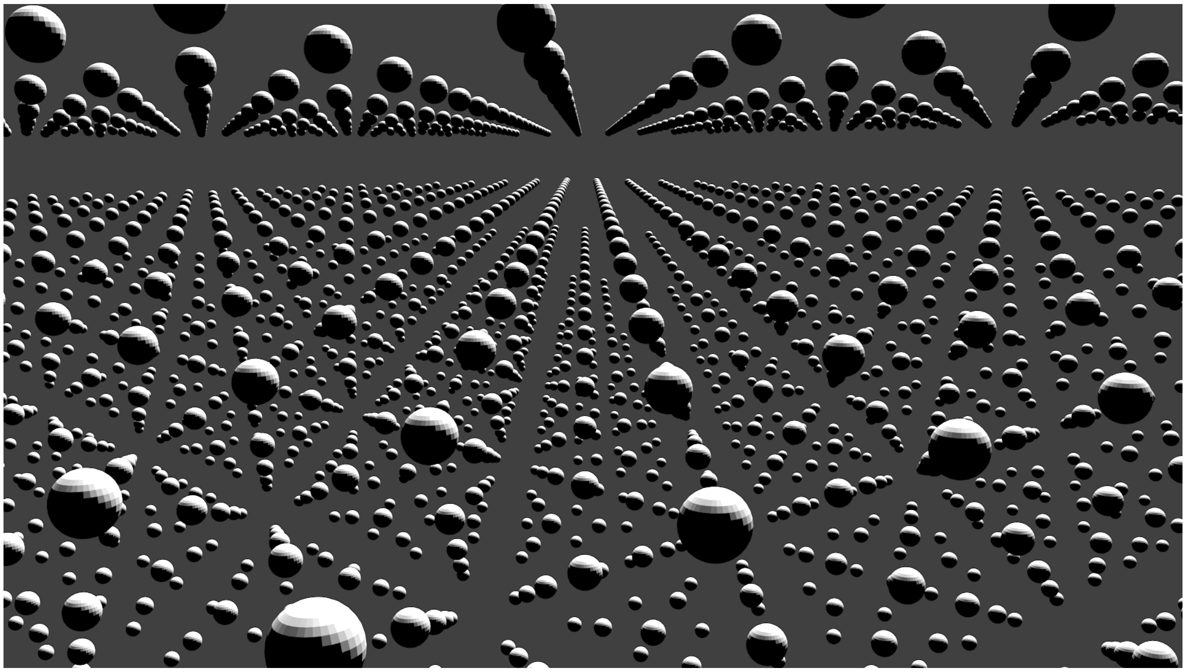

Figure 2.1: A lattice in three-dimensional space

Notation Have $B := (v_1|...|v_t) \in \mathbb{R}^{n \times t}$. Then define the lattice $\mathcal{L}(B) := \{Bx | x \in \mathbb{Z}^t\}$.

2.3 Definition (Fundamental Parallelepiped)

For a lattice $L = \mathcal{L}(B)$ with basis $B \in \mathbb{R}^{n \times n}$, have the fundamental parallelepiped $\mathcal{P}(B) := B[0,1)^n \subseteq \mathbb{R}^n$. The fundamental parallelepiped is a representative system for the quotient module $\mathbb{R}^n / \mathcal{L}(B)$, so each point $x \in \mathbb{R}^n$ has exactly one representation $x = x_0 + v$ with $x_0 \in \mathcal{B}$ and $v \in L$.

Notation Denote $x \bmod \mathcal{P}(B) := x_0$.

2.4 Example (Easy Lattice Problems)

For cryptographic purposes, we are interested in computational problems that work on lattices. For example, given a basis $B \in \mathbb{R}^{n \times n}$, the following problems can be solved in polynomial time:

- Given $x \in \mathbb{R}^n$, check if $x \in \mathcal{L}(B)$
- Given another basis $B' \in \mathbb{R}^{n \times n}$, check if $\mathcal{L}(B) = \mathcal{L}(B')$

These "algebraic" problems in lattices are usually easy (they can be solved using Gaussian Elimination). We get the hard problems needed for cryptography when we look at "geometric" problems, i.e. problems that rely on a norm in addition to the additive structure of the lattice.

2.5 Remark (Norm)

Denote with $\|x\|$ the l_2-norm of the vector $x \in \mathbb{R}^n$ and denote with $\|x\|_\infty$ the l_∞-norm of x. We also write $\langle \cdot, \cdot \rangle$ for the standard inner product on $\mathbb{R}^n$ resp. $\mathbb{C}^n$.

We will also require use a "norm" on the $\mathbb{Z}$-module $(\mathbb{R}/q\mathbb{Z})^n$, which should be the classical l_2 resp. l_∞ norm based on the "absolute value" of $x \in \mathbb{T}_q := \mathbb{R}/q\mathbb{Z}$ defined as $|x| := \min\{|r| | r \in \mathbb{R}, x = r \bmod q\mathbb{Z}\}$, i.e. the absolute value of the representative of x in $[-q/2, q/2)$. This is not a norm in the classical sense (e.g. in general $\|ax\| \neq |a|\|x\|$), but nevertheless, it will be useful.

2.6 Definition (Dual Lattice)

For a n-dimensional full-rank lattice L, define the dual lattice as $L^* := \{x \in \mathbb{R}^n | \langle x, L \rangle \subseteq \mathbb{Z}\}$ (where $\langle x, L \rangle$ denotes $\{\langle x, y \rangle | y \in L\}$).

2.7 Remark

The dual lattice is a lattice. Obviously, it is a $\mathbb{Z}$-module, so we have to show that it is free. Let $L = \mathcal{L}(B)$ with a basis $B = (b_1|...|b_n) \in \mathrm{GL}_n(\mathbb{R})$. Then $(B^T)^{-1} = (b_1^*|...|b_n^*) \in \mathrm{GL}_n(\mathbb{R})$ is a basis of the dual lattice.

Independent $b_1^*, ..., b_n^*$ are linearly independent, since B^* is invertible

$\subseteq$ Have $x \in L^*$. Then $\langle x, b_i \rangle \in \mathbb{Z}$ and so $u^T := x^T B \in \mathbb{Z}^n$. We get

$$x^T = u^T B^{-1} \text{ and so } x = \left(B^{-1}\right)^T u = \left(B^T\right)^{-1} u \in \left(B^T\right)^{-1} \mathbb{Z}^n = \mathcal{L}\left(B^*\right)$$

$\supseteq$ Follows from $\langle b_i^*, b_j \rangle = \delta_{ij} \in \mathbb{Z}$. This holds, since

$$\langle b_i^*, b_j \rangle = \left(b_i^*\right)^T b_j = \left(\left(B^T\right)^{-1} e_i\right)^T B e_j = e_i^T B^{-1} B e_j = e_i^T e_j = \delta_{ij}$$

2.8 Definition (Lattice quantities)

Let L be a lattice. Define

$$\lambda_k(L) := \min\{\max\{\|x_1\|, ..., \|x_k\|\} \mid x_1, ..., x_k \in L \text{ linearly independent}\}$$

as the shortest length so that there are k linearly independent vectors of at most this length in L. The length of a shortest, non-zero vector $\lambda_1(L)$ is also denoted by $\lambda(L)$. For a point $x \in \mathbb{R}^n$ define also $\kappa(x) = \kappa_L(x) \in L$ as the nearest lattice point to x. If x is within distance $< \frac{\lambda(L)}{2}$ to L, this is unique. We will only consider this case, but for a complete definition we might define $\kappa(x)$ as the lexicographically first lattice point among all closest ones.

2.9 Lemma

For any lattice L, we have

$$\lambda_1(L^*)\lambda_n(L) \geq 1$$

Proof. Let $x \in L^*$ be a vector of norm $\|x\| = \lambda_1(L^*)$ and have a basis $b_1, ..., b_n \in L$ with $\|b_i\| \leq \lambda_n(L)$. Then there is one i with $\langle x, b_i \rangle \neq 0$ (otherwise all b_i would be in the $n-1$ dimensional hyperplane perpendicular to x, a contradiction to them being a basis). We have $|\langle x, b_i \rangle| \geq 1$ since $b_i \in L, x \in L^*$ so $\langle x, b_i \rangle \in \mathbb{Z}$. By Cauchy-Schwarz, we get

$$\lambda_1(L^*)\lambda_n(L) \geq |\langle x, b_i \rangle| \geq 1$$ □

2.10 Definition (Gaussian distribution)

For any $s > 0$, define the function

$$\rho_s : \mathbb{R}^n \to \mathbb{R}, x \mapsto \exp\left(-\pi \frac{\|x\|^2}{s^2}\right)$$

We extend ρ to sets $K \subseteq L$ for a lattice $L \subseteq \mathbb{R}^n$ by $\rho_s(K) := \sum_{x \in K} \rho_s(x)$. The sum is finite since $\int_{\mathbb{R}^n} \rho_s(x)dx = s^n$ is finite.

We also have that $\frac{1}{s^n}\rho_s$ is a probability density function, and the corresponding distribution is the (continuous) normal distribution ν_s with standard deviation $s/\sqrt{2\pi}$. More interesting however is the discrete gaussian distribution $D_{L,s}$ on a lattice L and width s, which we define by the probability mass function

$$f(x) = \frac{\rho_s(x)}{\rho_s(L)}$$

2.11 Definition (Smoothing Parameter)

For a lattice L and some $\epsilon > 0$, let the smoothing parameter $\eta_\epsilon(L)$ be the smallest $s > 0$ so that $\rho_{1/s}(L^* \setminus \{0\}) \leq \epsilon$.

The following propositions show why this is called “smoothing parameter”. Intuitively, the smoothing parameter represents the amount of noise that is required to “smooth” out the discrete structure of the lattice.

2.12 Lemma

For a lattice $L = \mathcal{L}(B)$, $\epsilon > 0$ and $r \geq q\eta_\epsilon(L) = \eta_\epsilon(qL)$ we get for a random variable $v \sim D_{L,r}$ that $v \bmod qL$ is within statistical distance 2ϵ from a uniform distribution on $B\mathbb{Z}_q^n \cong L/qL$.

In particular, $B^{-1}v$ is almost uniform in $\mathbb{Z}_q^n$.

Proof. This lemma follows from [Reg09, p. 3.8].

2.13 Lemma

For a lattice L, $u \in \mathbb{R}^n$ and $r, s > 0$, define the random variables $v \sim D_{L+u,r}$ and $e \sim \nu_s$. If

$$rs/\sqrt{r^2+s^2} \geq \eta_\epsilon(L)$$

for some $\epsilon < \frac{1}{2}$, the statistical distance between $v + e$ and $\nu_{\sqrt{r^2+s^2}}$ is at most 4ϵ.

Proof. See [Reg09, p. 3.9].

2.14 Lemma

For a lattice L, $c > 0$ and $\epsilon(n) \in o(1)$ have for sufficiently large n that

$$\eta_{e(n)}(L) > \frac{c}{\lambda_1(L^*)} \geq \frac{c\lambda_n(L)}{n}$$

Proof. See [Reg09, p. 2.13].

2.15 Lemma

For a lattice L we have with $\epsilon = 2^{-n}$ that

$$\eta_e(L) \leq \frac{\sqrt{n}}{\lambda_1(L^*)}$$

Proof. See e.g. [Reg09, p. 2.11].

2.16 Lemma

For a lattice L and $r > 0$ we have

$$\rho_r(L \setminus \sqrt{n}rB_n) < 2^{-2n}\rho_r(L)$$

where $\sqrt{n}rB_n = \{x \in L | \|x\| < r\sqrt{n}\}$ is the set of lattice points with norm smaller than $r\sqrt{n}$. In other words, a point from $D_{L,r}$ has norm greater $r\sqrt{n}$ with only negligible probability.

Proof. See [Ban93, 1.5(i)].

2.17 Definition (Computational Lattice Problems)

Let $\gamma(n) \geq 1$. Define the following computational problems:

- (Shortest Vector Problem, SVP_γ) Given a basis of a full-rank lattice $B \in \mathbb{R}^{n \times n}$, find a short lattice vector, i.e. find $x \in \mathcal{L}(B)$ so that $\|x\| \leq \gamma\lambda_1(\mathcal{L}(B))$.
- (Shortest Independent Vector Problem, SIVP_γ) Given a basis of a full-rank lattice $B \in \mathbb{R}^{n \times n}$, find independent lattice vectors $v_1, ..., v_n \in \mathcal{L}(B)$ that are quite short, i.e. $\|v_i\| \leq \gamma\lambda_n\mathcal{L}(B))$.
- (Closest Vector Problem, CVP_γ) Given a basis of a full-rank lattice $B \in \mathbb{R}^{n \times n}$ and $a \in \mathbb{R}^n$, find a lattice point that is close to x, i.e. find $x \in \mathcal{L}(B)$ with $\|x - a\| \leq \gamma\|\kappa(x) - a\|$ where $y \in \mathcal{L}(B)$ is the closest lattice point to x.
- (GapSVP_γ) Given a basis of a full-rank lattice $B \in \mathbb{R}^{n \times n}$ so that either $\lambda(\mathcal{L}(B)) < 1$ or $\lambda(\mathcal{L}(B)) \geq \gamma$, find out what is the case.
- (Bounded Distance Decoding, BDD_γ) Given a basis of a full-rank lattice $B \in \mathbb{R}^{n \times n}$ and $a \in \mathbb{R}^n$ with $\|\kappa(a) - a\| < \gamma$, find $\kappa(a)$.

We also write $\text{BDD}_{L,\gamma}$ for the bounded distance decoding problem in a fixed lattice L.

2.18 Remark (Difficulty)

The difficulty of those problems depends on the choice of the parameter γ. Using the LLL algorithm for example, one can efficiently compute for a given lattice $L = \mathcal{L}(B)$ a basis with relatively small vectors $b'_1, ..., b'_n$. In particular one can get

$$\|b'_1\| \leq \gamma\lambda(L) \text{ for } \gamma = 2^{\frac{1}{2}(n-1)}$$

and so solve SVP_γ for exponential γ. On the other hand, SVP_γ, CVP_γ, GapSVP_γ and BDD_γ are conjectured to be hard for polynomial or smaller γ (resp. bigger γ in case of BDD), and there are various results for smaller γ. For example, GapSVP_γ for $\gamma \in O(n^{c/\log\log n})$ is NP-hard under randomized reductions.

2.19 Example

$$\text{SVP}_\gamma \leq_p \text{CVP}_\gamma \tag{2.1}$$

Proof. Let $B \in \mathbb{R}^{n \times n}$ be a basis and $L := \mathcal{L}(B)$. Let $x \in L \setminus \{0\}$ be the shortest vector in L. $\Rightarrow x = Ba$ for $a \in \mathbb{Z}^n$.

We have $a_j \equiv 1 \pmod 2$ for some j, since otherwise $\frac{1}{2}a \in \mathbb{Z}^n$ and so $\frac{1}{2}Ba \in L$ would be shorter than x. So the following algorithm finds x:

- For i in $\{1, ..., n\}$
  - Calculate basis of $L_i := \{Ba + Be_i | a \in \mathbb{Z}^n, a_i \equiv 1 \pmod 2\}$
  - Use oracle to find approximation to closest lattice point x_i to Be_i

- Output the shortest of all $x_i - Be_i$

Since $a_j \equiv 1 \pmod{2}$, we have $x + Be_j \in L_j$. As x is the closest nonzero vector to 0 in L, $x + Be_j$ is the closest vector $\neq Be_j$ to Be_j in L. So the oracle will yield $x + Be_j \in L_j \subseteq L$, because $Be_j \notin L_j$. Therefore, $x_j - Be_j = x$ and the algorithm outputs x or another vector of the same length. □

Chapter 3

Learning with Errors

Let $q \in \mathbb{Z}, q \geq 2$ be a prime number and have $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$.

3.1 Motivation

While the lattice problems we looked at in the previous section are considered hard, they are not very well suited for cryptographic applications. One reason is that all the considered vectors are continuous and have (in principle) unbounded length. However, it is sufficient to only work with points on a discrete grid in the fundamental parallelepiped, which is easier and more efficient to implement. This leads to the Learning with Errors problem, which can be characterized as solving a noisy system of linear equations in $\mathbb{Z}_q$, i.e. find x given

$$\begin{aligned} b_1 &\approx a_{11}x_1 + ... + a_{1n}x_n \\ &\vdots \\ b_n &\approx a_{m1}x_1 + ... + a_{mn}x_n \end{aligned}$$

Even though it is not immediately visible, this problem has a strong connections to lattices, especially with the bounded distance decoding problem. The advantage of LWE is that it can be implemented very efficiently, and it has a natural decision version that is as hard as LWE and very well suited for building public-key cryptosystems. Additionally, solving average case LWE is already sufficient to solve any instance of the problem, which is a singular property of lattice based cryptosystems. These results are not hard to prove, and we will do so in this section. The connection to the discussed lattice problems is more so, and we will consider it in the next section.

3.2 Definition

Define the $\mathbb{Z}$-module $\mathbb{T}_q := \mathbb{R}/q\mathbb{Z}$ with the submodule $\mathbb{Z}_q$ with the "norm"-like operation defined in 2.5. Let $\Psi_\alpha := \nu_{\alpha q} \bmod q$ be the centered normal distribution with standard deviation $\alpha q/\sqrt{2\pi}$ modulo q on $\mathbb{T}_q$.

3.3 Definition (Learning with Errors)

We define two variants of LWE:

Continuous For a probability distribution χ on $\mathbb{T}_q$ and some $s \in \mathbb{Z}_q^n$, define the LWE distribution $A_{s,\chi}$ on $\mathbb{Z}_q^n \times \mathbb{T}_q$ as the probability distribution that results from choosing an $a \in \mathbb{Z}_q^n$ uniformly, $e \sim \chi$ and outputting $(a, \langle a, s\rangle + e)$. We will often use $\chi = \Psi_\alpha$.

Discrete For a probability distribution χ on $\mathbb{Z}_q$ and some $s \in \mathbb{Z}_q^n$, define the LWE distribution $A_{s,\chi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ as the probability distribution that results from choosing an $a \in \mathbb{Z}_q^n$ uniformly, $e \sim \chi$ and outputting $(a, \langle a, s\rangle + e)$. We will often use $\chi = \bar{\Psi}_\alpha$, where $\bar{\Psi}_\alpha := \lfloor \Psi_\alpha \rceil$ is Ψ_α rounded to the nearest integer.

Then the (average case) LWE problem $\mathrm{LWE}_{q,\chi}$ is defined as: Given access to samples from $A_{s,\chi}$ for a uniformly chosen $s \in \mathbb{Z}_q^n$, retrieve s. We also write $\mathrm{LWE}_{q,\alpha}$ for $\mathrm{LWE}_{q,\Psi_\alpha}$.

If the number of samples is a fixed $m \in \mathbb{N}$, then LWE is often written as: Given $(A, b = As + e)$, find s where $A \in \mathbb{Z}_q^{m\times n}$ is a uniformly chosen matrix and $e \sim \chi^m$ is a random vector.

3.4 Remark

We define both versions of LWE, as the discrete one is better suited for computations, while it is often mathematically cleaner to work with a continuous error distribution. However, when we get samples $(A, b) \sim A_{s,\Psi_\alpha}$, then $(A, \lfloor b \rceil)$ is distributed like $A_{s,\bar{\Psi}_\alpha}$, so the discrete variant of LWE (which can be implemented) is at least as secure as the continuous one. One can also prove the opposite way, with only a slight increase in the amount of error.

3.5 Remark (Checking LWE solutions)

By definition, a given $s \in \mathbb{Z}_q^n$ is a valid LWE solution for samples (A, b) if $e := b - As$ is distributed according to χ^m. While this condition follows directly from the definition, it is not so easy to work with. Instead, we sometimes consider a solution $s \in \mathbb{Z}_q^n$ a valid LWE solution, if $\|e\|_\infty = \|b - As\|_\infty \leq \alpha q\sqrt{n}$. This holds for all samples from Ψ_α^m with high probability:

Let $e_1, ..., e_m \sim \Psi_\alpha$ be iid. random variables. Then $\|e\|$ is distributed according to the χ-distribution, and using properties of the χ-distribution, we get that $\Pr[\|e\| \geq \alpha q\sqrt{n}]$ is negligible. Since $\|e\|_\infty \leq \|e\|$, we get that $\|e\|_\infty \leq \alpha q\sqrt{n}$ with high probability.

In particular, this version of LWE is not less secure than the classic one.

3.6 Proposition (LWE solution is unique)

Given LWE samples $(A, b := As + e)$, where $A \in \mathbb{Z}_q^{m\times n}$ uniform and e chosen accordingly to $\bar{\Psi}_\alpha^m$, we have $\Pr[\exists s' \in \mathbb{Z}_q^n \setminus \{s\} : \|As' - b\|_\infty \leq \beta q] \leq q^n(2\beta + 1/q)^m$.

In particular, with high probability there is at most one LWE solution s if $m \in \omega(n \log q)$, $q \in \omega(1)$ and $\beta \leq \frac{1}{2} - \epsilon$ for $\epsilon > 0$.

Proof. Consider a fixed s, let $e \sim \chi^m$ be a random variable and let

$$A = \begin{pmatrix} ...a_1... \\ \vdots \\ ...a_m... \end{pmatrix}$$

be a random variable describing the choice of A in the LWE samples. We get:

$$\begin{aligned}
&\Pr[\exists s' \in \mathbb{Z}_q^n \setminus \{s\} : \|As' - b\|_\infty \leq \beta q] \\
=&\Pr[\exists s' \in \mathbb{Z}_q^n \setminus \{s\} : \|As' - As - e\|_\infty \leq \beta q] \\
\leq& \sum_{s' \in \mathbb{Z}_q^n \setminus \{s\}} \Pr[\|As' - As - e\|_\infty \leq \beta q] \\
=& \sum_{s' \in \mathbb{Z}_q^n \setminus \{s\}} \Pr[\forall i \in \{1, ..., m\} : |(A(s' - s) - e)_i| \leq \beta q] \\
=& \sum_{s' \in \mathbb{Z}_q^n \setminus \{s\}} \Pr[\forall i \in \{1, ..., m\} : |\langle a_i, s' - s\rangle - e_i| \leq \beta q]
\end{aligned}$$

Since the a_i are independent uniformly distributed, the $\langle a_i, s' - s\rangle - e_i$ are also independent and uniformly distributed (since $s' - s \neq 0$ is a constant). It follows:

$$\begin{aligned}
&\Pr[\exists s' \in \mathbb{Z}_q^n \setminus \{s\} : \|As' - b\|_\infty \leq \beta q] \\
\leq& \sum_{s' \in \mathbb{Z}_q^n \setminus \{s\}} \prod_{i=0}^{m} \Pr[|\langle a_i, s' - s\rangle - e_i| \leq \beta q] \\
=& \sum_{s' \in \mathbb{Z}_q^n \setminus \{s\}} (2\beta + 1/q)^m \\
\leq& q^n (2\beta + 1/q)^m
\end{aligned}$$

This holds for every fixed s, so it holds also for some uniformly distributed random s.

If $m \in \omega(n \log q), 2\beta \leq 1 - \epsilon$, get that $2\beta + 1/q \leq 1 - \epsilon$ for sufficiently large n, and so

$$q^n (2\beta + 1/q)^m \leq e^{n \log q} (2\beta + 1/q)^{\omega(n \log q)} = e^{n \log q - \omega(n \log q)} \text{ negligible.} \qquad \square$$

3.7 Proposition (Worst case to average case reduction)

A singular property of some lattice based problems is that they allow randomized self reductions, so an instance of the problem can be reduced to solving another, random instance of the problem. For LWE this means:

Assume we have access to an oracle that solves average case $\text{LWE}_{q,\chi}$, i.e. given samples of $A_{s,\chi}$ for a uniformly chosen $s \in \mathbb{Z}_q^n$, can retrieve s with probability exponentially close to 1 (over the randomness of s and the samples). Then we can solve worst case $\text{LWE}_{q,\chi}$, i.e. given samples of $A_{s,\chi}$ for any $s \in \mathbb{Z}_q^n$, can retrieve s with probability exponentially close to 1 (over the randomness of the samples).

In particular, this reduction is lattice-preserving, so on input samples (A, b), it calls the oracle with (A, b'). Therefore, the reduction remains valid when choosing A from another distribution (in both problems).

Proof. Consider the algorithm that, on input (A, b) does the following:

- Choose uniformly random $d \sim \mathbb{Z}_q^n$
- Call oracle on $(A, b + Ad)$ to get s' and output $s = s' - d$, where (A, b) are the input samples

The samples $(A, b + Ad)$ generated in the second step are distributed according to $A_{s',\chi}$ with secret $s' = s + d$, as $Ad + b = Ad + As + e = A(s + d) + e$. We chose d uniformly and independently of s, so $s' = s + d$ is uniformly distributed, and the oracle will correctly give us s' when called on $(A, b + Ad)$. As a result, we have found $s = s' - d$. □

3.8 Definition (Decision LWE)

Consider the decision version of LWE for a probability distribution χ: Given access to samples from $A_{s,\chi}$ for a fixed uniformly chosen $s \in \mathbb{Z}_q^n$ or samples from (a, u) where $a \in \mathbb{Z}_q^n$ and $u \in \mathbb{Z}_q$ resp. $u \in \mathbb{T}_q$ are uniformly chosen, determine what is the case.

3.9 Proposition (Search to decision reduction)

Let $q \in O(\text{poly}(n))$. Then the decision LWE problem is not easier than the search version, i.e. for a probability distribution χ on $\mathbb{T}_q$, given an oracle that can distinguish samples from $A_{s,\chi}$ for a uniformly chosen $s \in \mathbb{Z}_q^n$ and samples (a, u) with uniform $a \in \mathbb{Z}_q^n$ and $u \in \mathbb{T}_q$ (with a polynomial distinguishing probability gap Δ), we can find solve $\text{LWE}_{q,\chi}$.

This reduction is secret-preserving, so on input samples $(A, As + e)$ it calls the oracle on samples $(A', A's + e)$. Therefore, this reduction remains valid when choosing s from another distribution than the uniform one (in both problems).

Proof. Let $m \in O(\text{poly}(n))$ be the maximum count of samples the oracle needs. Consider the algorithm F that does the following:

- For $k \in \{1, ..., n\}$ and each $r \in \mathbb{Z}_q$ do:
    - Check if $s_k = r$ by repeating this procedure sufficiently often:
        - Select $d \in \mathbb{Z}_q^m$ uniformly random
        - Get m input samples (A, b) with $A \in \mathbb{Z}_q^{m \times n}$
        - Construct the matrix A' with d in the k-th column

$$A' := \begin{pmatrix} 0 & & d_1 & & 0 \\ \vdots & ... & \vdots & ... & \vdots \\ 0 & & d_m & & 0 \end{pmatrix}$$

  - Call the oracle with samples $(A + A', b + dr)$. If we got "not uniform", then $s_k = r$, otherwise $s_k \neq r$
- Set $s_k = r$ if the check returned true

Show The check procedure will distinguish situations with $s_k = r$ and $s_k \neq r$ with acceptance probability gap at least Δ over the choices of d, the samples (A, b) and the coins of the oracle.

Let p be the probability that the oracle will output "not uniform" on a uniform sample. For a given $k \in \{1, ..., n\}$, we have $(A + A')s = As + ds_k$.

If $r = s_k$, then $(A + A', b + dr)$ is distributed according to $A_{s,\chi}$, since $A + A'$ is uniform and $(A + A')s = As + dr$.
$\Rightarrow$ The oracle will return "not uniform" with probability $p + \Delta$ and the algorithm will find $s_k = r$ with probability $p + \Delta$.

If $r \neq s_k$, then $b + dr = As + e + dr = (A + A')s + e + d(r - s_k)$. Since $r - s_k \neq 0$ and d uniform, $d(r - s_k)$ and therefore both $b + dr = (A + A')s + e + d(r - s_k)$ and $A + A'$ are uniform.
$\Rightarrow$ The oracle will return "uniform" with probability $1 - p$ and the algorithm will find $s_k \neq r$.

By repeating the check procedure (with different d) and checking if significantly more resp. less than $p + \frac{1}{2}\Delta$ cycles accepted, the algorithm can determine whether $s_k = r$ with very high probability for each r and therefore find s_k and s. □

3.10 Proposition (Secret from Error Distribution)

LWE becomes no easier when we draw s from the error distribution χ on $\mathbb{Z}_q$ instead of the uniform distribution on $\mathbb{Z}_q^n$. In other words: Given an oracle that, given samples from $A_{s,\chi}$, where $s \sim \chi$ is random, can find s, we can solve $\mathrm{LWE}_{q,\chi}$ efficiently.

Proof. The idea in this proof is that we can request more samples and "eliminate" the secret from the formula. Then a part of the error takes the place of the secret, and we can find it using the oracle. When we know the error, we can find the secret using gaussian elimination.

Assume the oracle requests m samples. Then consider the algorithm that, on $m + n$ input samples $(A_1, b_1), (A_2, b_2)$ for $A_1 \in \mathbb{Z}_q^{m\times n}, A_2 \in \mathbb{Z}_q^{n\times n}$ calls the oracle on

$$(A_1 A_2^{-1}, A_1 A_2^{-1} b_2 - b_1)$$

and returns $A_2^{-1}(b_2 - s')$ where s' is the output of the oracle. We can do so, as A_2 is invertible with probability of $1 - 1/q$: Since $\#\det^{-1}(\{k\}) = \#\det^{-1}(\{l\})$ for $k, l \in \mathbb{Z}_q$, we have that $\det(A_2)$ is uniform on $\mathbb{Z}_q$, and therefore the probability that $\det(A_2) = 0$ is $1/q$.

For the input, we have $A_1 s + e_1 = b_1$ and $A_2 s + e_2 = b_2$. We claim that the generated samples are distributed according to $A_{e_2,\chi}$ where $e_2 \sim \chi$, so the oracle will yield $s' = e_2$ and we will output $A_2^{-1}(b_2 - e') = A_2^{-1} A_2 s = s$. We have

$$A_1 A_2^{-1} b_2 - b_1 = A_1 A_2^{-1}(A_2 s + e_2) - A_1 s - e_1 = A_1 A_2^{-1} e_2 - e_1$$

Since A_1 is uniform and independent of A_2 and A_2 is invertible, $A_1 A_2^{-1}$ is uniform on $\mathbb{Z}_q^{m\times n}$. Obviously e_1 and $s' = e_2$ are drawn from χ^m resp. χ^n, so we have generated valid samples of $A_{e_2,\chi}$. □

Chapter 4

Hardness of LWE

4.1 Motivation

While LWE has some properties that make it well suited for cryptographic applications, it is not as established as many well-known cryptographic problems, and there is much less evidence for its hardness. Therefore, it would be desirable to base its security on some problem that has been examined in more detail and whose hardness is better understood. In this section, we will present such a reduction to one of the standard lattice problems that we have already seen, namely the approximation of the short vector problem. This also demonstrates the connection to lattices. An interesting point is that this reduction is quantum, so we assume the quantum hardness of the short vector problem. There are also classical reductions, but they impose restrictions on the parameters and are less central to our understanding of LWE (e.g. [Pei09]). We will also present a cryptosystem that already has some similarities with Crystals-Kyber.

4.2 Definition

For $\alpha > 0$, consider also the following variant $\mathrm{LWE}_{q,\leq\alpha}$ of worst case LWE: For any $s \in \mathbb{Z}_q^n$ and any (unknown) $\beta \leq \alpha$, given samples from A_{s,Ψ_β}, retrieve s.

4.3 Proposition (Unknown to constant error variance reduction)

Given an oracle for worst case LWE with error distribution Ψ_α, we can solve $\mathrm{LWE}_{q,\leq\alpha}$.

Proof. The idea of this proof is the following lemma from [Reg09, p. 2.2]:

Lemma For any $\alpha > 0$ and $0 < \epsilon \leq \alpha$, we have that the statistical distance between Ψ_α and $\Psi_{\alpha+\epsilon}$ is at most $9\epsilon/\alpha$.

Let $p(n) \in (0,1]$ be the success probability of the oracle and $m \in O(\text{poly}(n))$ the amount of samples the oracle needs. Then define

$$\epsilon := \frac{1}{18m} p\alpha \quad \text{and} \quad N = \alpha/\epsilon \in O(\text{poly}(n))$$

Given samples $(A,b) \sim A_{s,\beta}$ we can now find s by calling the oracle on $(A, b+e_k)$ for each $k \in \{0, ..., N\}$, where $e_k \sim \nu_{k\epsilon}^m$ is a normal random variable. If one of these calls returns the correct secret (we can check that), this is our result. We have to show that with high probability, this happens.

Let k be the smallest number with $k\epsilon \geq \alpha - \beta$. We claim that with high probability, the oracle returns the correct secret when called on $(A, b+e_k)$. The random variable $(A, b+e_k)$ is distributed like $A_{s,\Psi_{\beta+k\epsilon}}$, and by the lemma, the statistical distance to A_{s,Ψ_α} is at most $9\epsilon/\alpha$. Therefore, the statistical distance between $A^m_{s,\Psi_{\beta+k\epsilon}}$ and A^m_{s,Ψ_α} is at most $m9\epsilon/\alpha = \frac{1}{2}p$. Since applying the oracle cannot increase the statistical distance, the output of the oracle on $(A, b+e_k)$ is within distance $\frac{1}{2}p$ of the output on valid samples $(A', b') \sim A^m_{s,\Psi_\alpha}$, and therefore, we find the secret with probability at least $\frac{1}{2}p$. □

4.4 Theorem (SIVP to LWE reduction)

For $\alpha q \geq 2\sqrt{n}$, there is a quantum reduction from worst case $\text{LWE}_{q,\leq\alpha}$ to worst case $\text{SIVP}_{2n\sqrt{2n}/\alpha}$. Using some stronger bounds on $\eta_\epsilon(L)$, [Reg09] even proves the reduction to $\text{SIVP}_{2n\sqrt{2\omega(\log(n))}/\alpha}$.

4.4.1 Remark

We will present a summary of the proof given in [Reg09]. This proof relies heavily on some technical results about the smoothing parameter and the discrete gaussian distribution that go beyond the scope of this work. Therefore, we will only sketch some edge parts of the argument, but we try to present the main ideas formally correct (namely the use of the LWE oracle and the quantum part of the reduction).

Proof. The heart of the proof is the following lemma:

4.5 Lemma (Improvement of gaussian samples using LWE)

Let ϵ be negligible, $\alpha \in (0,1)$ and $r > \sqrt{2}q\eta_\epsilon(L)$. Given an oracle for $\text{LWE}_{q,\leq\alpha}$ and samples from $D_{L,r}$ (as many as the oracle needs), we can efficiently generate samples from $D_{L,r\sqrt{n}/\alpha q}$. The generated samples are stochastically independent of the input samples and of each other.

Proof. We begin with stating the following lemma on the inner product with discrete gaussian vectors:

Lemma 1 For a lattice L, $z, u \in \mathbb{R}^n$, $r, \alpha > 0$ and $\epsilon \leq \frac{1}{2}$ with

$$1/\sqrt{\frac{1}{r^2} + \frac{\|z\|^2}{\alpha^2}} \geq \eta_\epsilon(L)$$

we have for random variables $v \sim D_{L+u,r}$ and e normally distributed with standard deviation $\alpha/\sqrt{2\pi}$ that the $\langle z, v\rangle + e$ is within statistical distance 4ϵ to

$$\nu_{\sqrt{(r\|z\|)^2+\alpha^2}}$$

In particular, we get that $\langle z, qv\rangle + qe \bmod q$ is within statistical distance 4ϵ to Ψ_α.

Proof. See e.g. [Reg09, p. 3.10].

Intuitively, this lemma says that the scalar product of a constant vector with a vector drawn from the discrete gaussian distribution is distributed like an LWE error.

Let $L = \mathcal{L}(B)$ and $L^* = \mathcal{L}(B^*)$ with $B^* = \left(B^{-1}\right)^T$. For this proof, we also require a variant of the BDD problem:

4.5.1 Definition $\mathrm{BDD}^q_{L,\gamma}$

Define the problem as: Given $x \in \mathbb{R}^n$ close to L, calculate the coefficients of $\kappa(x)$ modulo q, i.e. calculate $s \bmod q$ where $Bs = \kappa(x)$ when given a point x close to L (within distance γ)

Then proof consists of the following parts:

- Construct an oracle for $\mathrm{BDD}^q_{L^*,\alpha q/\sqrt{2}r}$.
- Construct an oracle for $\mathrm{BDD}_{L^*,\alpha q/\sqrt{2}r}$.
- Using this, quantumly generate samples from $D_{L,r\sqrt{n}/\alpha q}$.

4.5.2 Oracle for $\mathrm{BDD}^q_{L^*,\alpha q/\sqrt{2}r}$

Let $v_1, ..., v_n \in L$ denote the samples from $D_{L,r}$, so have $v_1, ..., v_n$ iid. random variables with $v_i \sim D_{L,r}$. On input $x \in \mathbb{R}^n$ with $\|x - \kappa(x)\| \leq \alpha q/r\sqrt{2}$ define

$$s = (B^*)^{-1}\kappa(x) = B^T\kappa(x) \in \mathbb{Z}^n \qquad \text{since} \qquad \kappa(x) \in L^*$$
$$a_i = B^{-1}v_i \in \mathbb{Z}^n$$

and get

$$\langle a_i, s\rangle = \langle B^{-1}v_i, B^T\kappa(x)\rangle = \langle v_i, \kappa(x)\rangle = \langle v_i, x\rangle + \langle v_i, \kappa(x) - x\rangle$$

The statistical distance between $a_i \bmod q$ and a uniform random variable on $\mathbb{Z}_q^n$ is negligible by 2.12. Now we condition on a fixed $a_i \bmod q$. We get that $v_i/q \sim D_{L+Ba_i/q,r/q}$ and

$$1/\sqrt{\frac{q^2}{r^2} + 2\frac{\|\kappa(x) - x\|^2}{\alpha^2}} \geq 1/\sqrt{\frac{q^2}{r^2} + 2\frac{\alpha^2 q^2}{\alpha^2 2r^2}} = \frac{r}{\sqrt{2}q} > \eta_\epsilon(L)$$

so we can apply lemma 1 with $u = Ba_i/q$ and get that $\langle v_i/q, \kappa(x) - x\rangle + e$ is distributed according to ν_β where $e \sim \nu_{\alpha/\sqrt{2}}$ is a normal random variable with standard deviation $\alpha/2\sqrt{\pi}$ and $\beta^2 = (r\|x - \kappa(x)\|/q)^2 + (\alpha/\sqrt{2})^2 \leq \alpha^2$. Since this holds conditioned on any arbitrary a_i, the error random variable $\langle v_i/q, \kappa(x) - x\rangle + e$ is also independent from a_i. Therefore, $(a_i \bmod q, \langle v_i, x\rangle + qe \bmod q)$ are valid samples of A_{s,Ψ_β} and our oracle can retrieve $s \bmod q$.

4.5.3 Oracle for $\mathrm{BDD}_{L^*,\alpha q/\sqrt{2}r}$

The idea is to find the digits of the coefficients in the base q by repeatedly applying the oracle to new points. Let $x \in \mathbb{R}^n$ with $\|x - \kappa(x)\| \leq \alpha q/\sqrt{2}r$ be the input. Denote by s the coefficients of $\kappa(x)$, i.e. $s = B^T\kappa(x)$. Now define

$$\begin{aligned} x_0 &= x \\ x_{i+1} &= \frac{x_i - B(s_i \bmod q)}{q} \\ s_i &= \left\lfloor \frac{s}{q^i} \right\rfloor = \frac{s - (s \bmod q^i)}{q^i} \in \mathbb{Z}^n \end{aligned}$$

Then $\kappa(x_i) = Bs_i$ by induction, since for $x_i' = x_i - B(s_i \bmod q)$ we get $x_i' - x_i = B(s_i \bmod q) \in L$ so

$$\kappa(x_i') = \kappa(x_i) - B(s_i \bmod q) = B\left(s_i - (s_i \bmod q)\right) = Bq\left\lfloor \frac{s}{q^{i+1}} \right\rfloor = Bqs_{i+1}$$

Since $\|a - \kappa_\Lambda(a)\| \leq \|a - \kappa_\Gamma(a)\|$ if Γ is a sublattice of Λ, we get $\kappa_{qL}(x_i') = Bqs_{i+1}$ and therefore

$$\kappa(x_{i+1}) = \kappa_L\left(\frac{x_i'}{q}\right) = \kappa(x_{i+1}) = Bs_{i+1}$$

Using this and the oracle, we can also calculate x_{i+1} and so $(s_{i+1} \bmod q)$ given x_i and $s_i \bmod q$. By repeating this, we get $s_i \bmod q$ for any i. Now we find s by using that $s = \sum_i s_i \bmod q$.

4.5.4 Generate sample of $D_{L,r\sqrt{n}/\alpha q}$

Overview Some readers might not be used to quantum computation, so an elaborate high-level overview about this part will be presented before we come to the technical details. The main idea is that the (continous) fourier transformation transforms a distribution of repeated wide gaussians on the dual lattice to a distribution of the wanted (narrow) discrete gaussian on L. With "distribution of repeated gaussians" we mean the distribution resulting from choosing a lattice point and adding some (continous) gaussian noise, see figure 4.1. Therefore, we can approximate the discrete gaussians on L by applying the discrete fourier transformation on an approximation of this source distribution. Classically, this is of no use as we require exponentially many points to represent the source distribution well enough. However, the quantum fourier transform allows us to process exponentially many points in polynomial time, given a suitable superposition of them, something like

$$\sum_{x \in \mathbb{R}^n} e^{-\mathrm{dist}(x,L)^2} |x\rangle$$

Of course, we cannot sum over $\mathbb{R}$. In reality, we will use all points on a fine grid within the fundamental parallelepiped. Creating this state is not easy however, due to a caveat inherent to quantum computation. The first step of the "sample lattice points and add noise" still works, leading to a state that looks like

$$\sum_{x \in \mathbb{R}^n} e^{-\|x-\kappa(x)\|^2} |x\rangle \otimes |\kappa(x)\rangle$$

There is one problem in this equation though. The qbits containing the noisy point $|x\rangle$ are entangled with the original lattice point we added the noise to, namely $|\kappa(x)\rangle$. In quantum mechanics, all

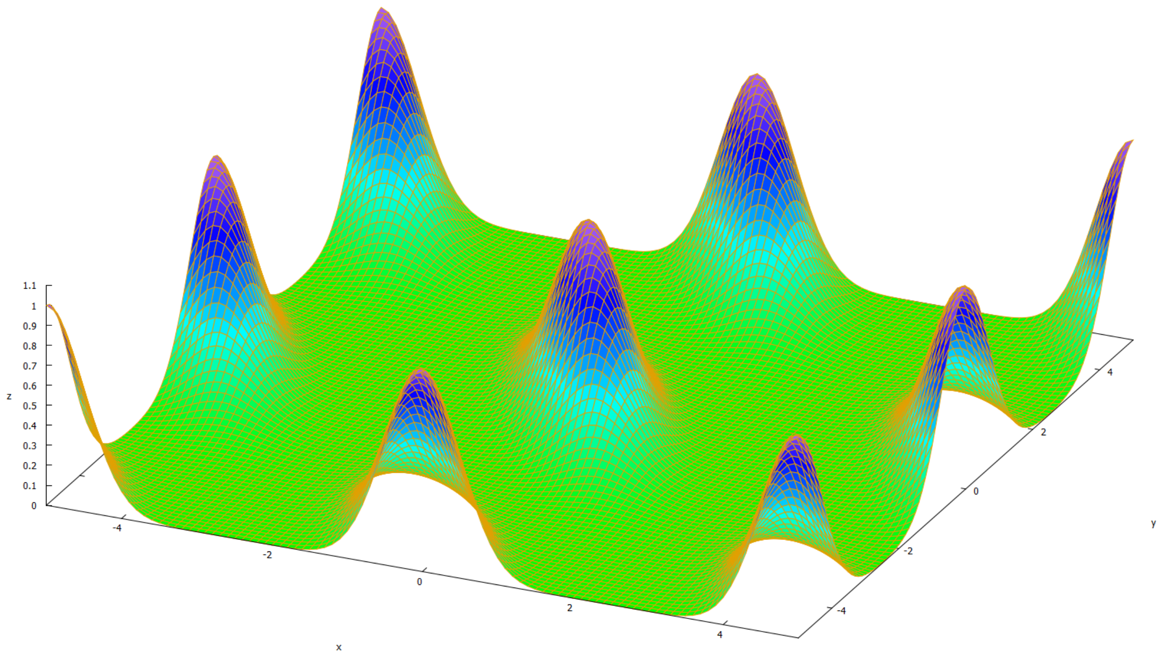

Figure 4.1: Inverse fourier transform of $D_{L,r}$, the "repeated gaussians"

operations have to be reversible, so we cannot just "drop" the unwanted qbits without destroying the superposition we need. But since we can compute $|\kappa(x)\rangle$ from $|x\rangle$ using the BDD-oracle, we can "uncompute" the $|\kappa(x)\rangle$ part by applying the quantum gates of the oracle in reverse and get the required superposition.

Details Let $d = \alpha q/r\sqrt{2n}$ and $R = 2^{3n}d\lambda_n(L^*)\sqrt{n}$ be a big enough integer (representing the fine grid size). Then we claim that the effect of the QFT can be described by

$$\text{QFT} \sum_{x \in \frac{1}{R}L^*} \rho_d(x) \left| x \bmod \mathcal{P}(B^*) \right\rangle = \det(RB) \sum_{t \in L} \rho_{1/d}(t) \left| t \bmod \mathcal{P}(RB) \right\rangle \tag{4.1}$$

where $\mathcal{P}(B)$ is the fundamental parallelepiped generated by the basis vectors in B. When we consider the limit $R \to \infty$, this corresponds to the abstract idea presented earlier, as the left part becomes an integral and the t mod $\mathcal{P}(RB)$ becomes t. We have

$$\begin{aligned}
&\mathrm{QFT} \sum_{x\in \frac{1}{R}L^*} \rho_d(x)\,|x \bmod \mathcal{P}(B^*)\rangle \\
&= \mathrm{QFT} \sum_{x\in\frac{1}{R}L^*\cap\mathcal{P}(B^*)} \rho_d(x+L^*)\,|x\rangle \\
&= \sum_{t\in L\cap \mathcal{P}(RB)} \sum_{x\in\frac{1}{R} L^*\cap\mathcal{P}(B^*)} \exp(-2\pi i\langle x,t\rangle)\rho_d(x+L^*)\,|t\rangle \\
&= \sum_{t\in L\cap\mathcal{P}(RB)}\sum_{x\in\frac{1}{R} L^*}\exp(-2\pi i\langle x,t\rangle)\rho_d(x)\,|t\rangle \\
&= \det(RB)\sum_{t\in L\cap\mathcal{P}(RB)}\sum_{x\in RL}\rho_{1/d}(x-t)\,|t\rangle \\
&= \det(RB)\sum_{t\in L}\rho_{1/d}(t)\,|t \bmod \mathcal{P}(RB)\rangle
\end{aligned}$$

where the second-last line holds due to the Poisson summation formula (Since ρ is equal to its fourier transformation, so $\rho_d = \hat{\rho}_{1/d}$). Here the use of the QFT on lattices should be the natural extension from the QFT on the coefficients, i.e.

$$\begin{aligned}
&\mathrm{QFT} \sum_{x\in\frac{1}{R}L\cap\mathcal{P}(B)} f(x)\,|x\rangle = \mathrm{QFT}\sum_{z\in\mathbb{Z}_R^n} f\left(\frac{Bz}{R}\right)\left|\frac{Bz}{R}\right\rangle \\
&:= \sum_{k\in\mathbb{Z}_R^n}\sum_{z\in\mathbb{Z}_R^n} f\left(\frac{Bz}{R}\right)\exp\left(-2\pi i\frac{\langle k,z\rangle}{R}\right)|B^*k\rangle \\
&= \sum_{k\in\mathbb{Z}_R^n}\sum_{x\in\frac{1}{R}L\cap\mathcal{P}(B)} f(x)\exp\left(-2\pi i\langle k,B^{-1}x\rangle\right)|B^*k\rangle \\
&= \sum_{k\in\mathbb{Z}_R^n}\sum_{x\in\frac{1}{R}L\cap\mathcal{P}(B)} f(x)\exp\left(-2\pi i\langle B^*k,x\rangle\right)|B^*k\rangle \\
&= \sum_{y\in L^*\cap\mathcal{P}(RB^*)}\sum_{x\in\frac{1}{R}L\cap\mathcal{P}(B)} f(x)\exp\left(-2\pi i\langle y,x\rangle\right)|y\rangle
\end{aligned}$$

This transform can be done by a quantum computer in polynomial time by multiplying the input register with RB^{-1}, then applying the standard QFT and then multiplying the result with B^*.

Note that the second term in the QFT result shown above (equation 4.1) is almost exactly what we want: When we measure this state, we get t mod $\mathcal{P}(RB)$ for a t chosen from $D_{L,1/d\sqrt{2}}$ (since the probability of getting t is proportional to $\rho_{1/d}(t)^2 = \rho_{1/d\sqrt{2}}(t)$). By the following lemma, we can efficiently get t from t mod $\mathcal{P}(RB)$, and therefore get a sample from $D_{L,1/d\sqrt{2}} = D_{L,r\sqrt{n}/\alpha q}$.

We can use the lemma, since $\lambda_1(RL) = R\lambda_1(L) \geq 2^{3n}d\sqrt{n}$ by using $\lambda_n(L^*)\lambda_1(L) \geq 1$ (by 2.16). In particular, $\lambda_1(RL) \geq d\sqrt{2n}$ and solving $\mathrm{BDD}_{RL,d\sqrt{n}/\sqrt{2}}$ is equivalent to solving $\mathrm{BDD}_{L,d\sqrt{n}/R\sqrt{2}}$ which is possible in polynomial time because $d\sqrt{n}/R\sqrt{2}$ is exponentially small (e.g. by using Babai's nearest plane algorithm).

Lemma 2 Let $L = \mathcal{L}(B)$ be a lattice, $c > 0$ and $x \sim D_{L/c,\gamma}$ be a random variable with $\lambda_1(L) \geq 2\gamma\sqrt{n}$ and define $x' := x \bmod \mathcal{P}(B)$. Assume we have access to an oracle for $\mathrm{BDD}_{L,\gamma\sqrt{n}}$. Then we can efficiently calculate x given $x' = x \bmod \mathcal{P}(B)$ with probability of at least $1 - 2^{-2n}$.

Proof. By 2.16, the probability that $\|x\| < \gamma\sqrt{n}$ is at least $1 - 2^{-2n}$. If this is the case, we have that $\|x - \kappa(x)\| < \gamma\sqrt{n}$, and so $\|\kappa(x') - x'\| < \gamma\sqrt{n}$. By using the oracle, we can calculate $\kappa(x')$.

Now we use that $\lambda_1(L) \geq 2\gamma\sqrt{n}$. This yields that $\kappa(x) = 0$, since the distance between the lattice points 0 and $\kappa(x)$ is $< 2\gamma\sqrt{n} \leq \lambda_1(L)$. Similarly, we have $\kappa(x) - \kappa(x') = x - x'$, because the distance between the lattice points $\kappa(x) - \kappa(x')$ and $x - x'$ is

$$\|\kappa(x) - \kappa(x') - x + x'\| \leq \|\kappa(x) - x\| + \|\kappa(x') - x'\| < 2\gamma\sqrt{n} \leq \lambda_1(L)$$

Therefore we find x by $x = x' + \kappa(x) - \kappa(x') = x' - \kappa(x')$.

The intuition in this argument is that x is distributed very narrowly around the origin. Therefore, $x \bmod \mathcal{P}(B)$ is very close to one of the corners of the fundamental parallelepiped, and the location of $x \bmod \mathcal{P}(RB)$ relative to this corner is the location of x relative to the origin. □

Creating the initial state We have shown that we can get a sample from the right side of equation 4.1, so it remains to be shown that we can efficiently create the state on the left side, namely

$$\sum_{x \in \frac{1}{R}L^*} \rho_d(x) |x \bmod \mathcal{P}(B^*)\rangle$$

It is not obvious, but this state corresponds to the "repeated gaussian" distribution mentioned earlier: Taking the gaussian (almost continous for large R) modulo the fundamental parallelepiped yields a distribution in which there are "hills" in each corner of the parallelepiped, and by filling the complete space with copies of the parallelepiped, we get the "repeated gaussian" distribution.

By dividing the register by R, we can easily get this state when we start with the state

$$\sum_{x \in L^*} \rho_{Rd}(x) |x \bmod \mathcal{P}(RB^*)\rangle$$

As shown in [Reg09] 3.2, we can efficiently create the state

$$\sum_{x \in L^*} \rho_{Rd}(x) |x\rangle$$

Classically, it looks like we could just take this state modulo $\mathcal{P}(RB^*)$ and be done. The problem is that quantum operations have to be reversible, so we have to be able to calculate x from $x \bmod \mathcal{P}(RB^*)$ in order to do this. Luckily, we can do that using Lemma 2:

We have $\lambda_1(RL^*) = R\lambda_1(L^*) \geq 2Rd\sqrt{n}$ since

$$2d\sqrt{n} = \frac{\alpha q}{r\sqrt{2}} \leq \frac{\alpha q}{2q\eta_\epsilon(L)} = \frac{\alpha}{2\eta_\epsilon(L)} \leq \frac{1}{2\eta_\epsilon(L)} \leq \frac{\lambda_1(L)}{2}$$

where the last inequality follows from 2.14 and we can solve $\mathrm{BDD}_{RL^*,Rd\sqrt{n}}$ which is equivalent to $\mathrm{BDD}_{L^*,d\sqrt{n}} = \mathrm{BDD}_{L^*,\alpha q/r\sqrt{2}}$ by using the oracle constructed earlier. □

Proof of 4.4 The idea is to start with samples of a very wide gaussian distribution which are easy to generate, and then improve them using the lemma 4.5 above. This way, we can get samples of a very narrow gaussian distribution. Among n^2 of these are n linearly independent one with very high probability, so we can use them to solve SIVP.

Show For a given $r \geq n\sqrt{2}\lambda_n(L)/\alpha$ we can efficiently sample from $D_{L,r}$.
As shown in [Reg09], we can efficiently sample from $D_{L,r'}$ for some computable, exponentially large $2^{3n}\lambda_n(L) \geq r' \geq 2^{2n}\lambda_n(L)$. The idea in that argument is to use a LLL basis of L. Let $\epsilon = 2^{-n}$ and apply lemma 4.5 repeatedly on these samples to create shorter ones. The condition of 4.5 is fullfilled as long as the input radius is $\geq q\sqrt{2n}\lambda_n(L)$, because by 2.15 and 2.9 we have

$$\sqrt{2}q\eta_\epsilon(L) \leq \frac{q\sqrt{2n}}{\lambda_1(L^*)} \leq q\sqrt{2n}\lambda_n(L)$$

Therefore we can reach the given r by repeated applications, because we have by assumption that $r \geq n\sqrt{2}\lambda_n(L)/\alpha \geq q\sqrt{2n}\lambda_n(L)\sqrt{n}/\alpha q$. Since every application of the recursive step shrinks the radius by a factor of 2, we can get samples of $D_{L,r}$ through polynomially many repetitions (as $\log r'/r$ is polynomial in n).

Lemma 3 The probability that n^2 independent samples of $D_{L,r}$ for some $r \geq \sqrt{2}\eta_\epsilon(L)$ are contained in any $n-1$ dimensional hyperplane is exponentially small. In other words, the probability that there are n linearly independent samples among those n^2 ones is exponentially close to 1.

Proof. See [Reg09, p. 3.16].

Now it looks like we could just get n^2 samples from $D_{L,r}$ for $r = n\sqrt{2}\lambda_n(L)/\alpha$ and output n linearly independent ones from among them (they have length $\leq 2n\sqrt{n}\lambda_n(L)/\alpha$ by 2.16). The only problem however is that we do not know $\lambda_n(L)$, so we cannot give the right r to the sampling oracle. We can however apply the whole procedure for each $r_k = 2^{-k}r'n\sqrt{2}/\alpha$ where r' is the exponential upper bound for $\lambda_n(L)$ mentioned above and take the shortest set of independent samples we find. Since $2^{3n}\lambda_n(L) \geq r' \geq 2^{2n}\lambda_n(L)$, there must be some $k \in \{2n, ..., 3n\}$ so that

$$r_k = 2^{-k}r'n\sqrt{2}/\alpha \in \left[n\sqrt{2}\lambda_n(L)/\alpha, 2n\sqrt{2}\lambda_n(L)/\alpha\right]$$

So when we try that k, we sample n^2 vectors from D_{L,r_k} and find n linearly independent vectors of length $\leq 2n\sqrt{2n}\lambda_n(L)/\alpha$ with high probability, which solves SIVP$_{2n\sqrt{2n}/\alpha}$. □

4.6 Remark (A cryptosystem)

Consider the following cryptosystem. While there are easier schemes based on LWE, this one is the most efficient one and will become important later as basis for Crystals Kyber.

Secret Key Choose uniform $s \sim \Psi_\alpha$

Public Key Choose $s, e_1 \sim \Psi_\alpha^n$ and $A \in \mathbb{Z}_q^{n\times n}$ uniformly, publish $(A, b) = (A, s^T A + e_1^T)$

Encryption Choose $r, e_2 \sim \Psi_\alpha^n, e_3 \sim \Psi_\alpha$, encrypt $m \in \{0, 1\}$ as $(Ar + e_2, br + e_3 + \lfloor \frac{q}{2} \rceil m)$

Decryption On receiving (u, v), check if $v - s^T u$ is closer to 0 or $\lfloor \frac{q}{2} \rceil$

When $2\sqrt{n}/q \leq \alpha \leq 1/3\sqrt{qn}$, then this scheme is both correct and secure under the assumption that SIVP for polynomial approximation factors is worst case quantum hard. A possible choice of parameters would therefore be $q = 36n^2$ and $\alpha = 1/18n\sqrt{n}$.

Correctness

Proof. Let $s, r, e_1, e_2 \sim \Psi_\alpha^n, e_3 \sim \Psi_\alpha, A \sim \mathbb{Z}_q^{n \times n}$ and $x \sim \{0,1\}^n$ be random variables representing the values chosen by the procedure, and let $b := s^T A + e_1^T$. Then the input to the decryption procedure is (u, v) with $u = Ar$ and $v = br + e_3 + \lfloor \frac{q}{2} \rceil m$. Therefore, we get

$$v - s^T u = s^T Ar + e_1^T r + e_3 + \left\lfloor \frac{q}{2} \right\rceil m - s^T Ar = e_1^T r + e_3 + \left\lfloor \frac{q}{2} \right\rceil m$$

With high probability, we have that $\|e_1^T\|, \|r\| \leq \alpha q \sqrt{n}$. By Cauchy-Schwarz, we have that $|e_1^T r + e_3| \leq \alpha^2 q^2 n + \alpha q = \alpha q(\alpha q n + 1)$. Since $\alpha \leq 1/3\sqrt{nq}$, it follows that for $\sqrt{qn} \geq 3$, we have

$$|e_1^T r + e_3| \leq \frac{\sqrt{q}}{3\sqrt{n}}(\sqrt{qn}/3 + 1) = \frac{q}{9} + \frac{q}{3\sqrt{nq}} \leq \frac{2q}{9} < \frac{q}{4}$$

Therefore, the distance between $v - u^T s$ and $m\lfloor \frac{q}{2} \rceil$ is at most $\frac{q}{4}$, and so we have that $m = 0$ if $v - s^T u$ is closer to 0, and $m = 1$ if $v - s^T u$ is closer to $\lfloor \frac{q}{2} \rceil$, which proves correctness. □

Security

Proof. Using the reductions 3.9, 3.10, 3.7 and 4.4 and assuming the quantum hardness of SIVP for the polynomial approximation factor $2n\sqrt{2n}/\alpha$, we get that an adversary cannot distinguish between A_{s,Ψ_α} for $s \sim \Psi_\alpha$ and (A, u) for uniform A, u.

Show An adversary cannot distinguish between encryptions of 0 and encryptions of 1. For this, we use that computational indistinguishability is transitive with respect to polynomial chaining. For independent and uniformly distributed random variables U_1, U_2 it holds that

$$\begin{aligned}
&\big(A, As + e_1, Ar + e_2, s^T Ar + e_1^T r + e_3\big) \\
\text{ind. from } &(A, U_1, Ar + e_2, U_2) \\
\text{ind. from } &(A, As + e_1, Ar + e_2, U_2) \\
\text{ind. from } &\left(A, As + e_1, Ar + e_2, U_2 + \left\lfloor \frac{q}{2} \right\rceil\right) \\
\text{ind. from } &\left(A, U_1, Ar + e_2, U_2 + \left\lfloor \frac{q}{2} \right\rceil\right) \\
\text{ind. from } &\left(A, As + e_1, Ar + e_2, s^T Ar + e_1^T r + e_3 + \left\lfloor \frac{q}{2} \right\rceil\right)
\end{aligned}$$

To see that the first and last indistinguishability hold, assume that an adversary could distinguish between both distributions. Then, given samples of (A, b) with either $b \sim A_{s,\Psi_\alpha}$ or b uniform, he could choose $r, e_2 \sim \Psi_\alpha^n, e_3 \sim \Psi_\alpha$ and call the oracle with $(A, b, Ar + e_2, br + e_3)$. If $(A, b) \sim A_{s,\Psi_\alpha}$, then $(A, b, Ar + e_2, br + e_3)$ is distributed identically to $(A, As + e_1, Ar + e_2, s^T Ar + e_1^T r + e_3)$ and if b is uniform, then $(A, b, Ar + e_2, br + e_3)$ is distributed identically to $(A, U_1, Ar + e_2, U_2)$. Therefore, the adversary could solve decision LWE, a contradiction.

Since the first distribution is equal to an encryption of 0 and the last one is equal to an encryption of 1, we see that an adversary cannot distinguish between them, so the scheme is secure. □

4.7 Remark (Choice of parameters)

As a summary of the results of this section, we will list the conditions the LWE parameters must satisfy so that we get the corresponding security guarantees. Note that in some cases, other values may not lead to an insecure system, we just cannot prove its security the same way we did in this section:

- q must be a prime. This was never mentioned explicitly, but it is the reason that $\langle x, a\rangle$ for a constant $a \neq 0$ and a uniform x is uniform on $\mathbb{Z}_q$, which we used multiple times.
- $\alpha q \geq 2\sqrt{n}$ so that 4.4 applies.
- The error must not be too big so that the LWE structure is still recognizable. As in 3.6, is suffices if $\alpha \leq \frac{1}{2\sqrt{n}} - \epsilon$.
- There must be enough samples so that there is a unique solution. When s is uniformly chosen, then, as in 3.6, it is sufficient if $\omega(n \log q)$ or $\Omega(n \log q)$ if $\alpha \in o(1)$. However, by drawing the secret from the error distribution (as in 3.10) and only accepting short secrets, a calculation as in 3.6 shows that $\Omega(n)$ samples are sufficient.

A possible cryptographic scheme is presented in 4.6 (tighter bounds on the error are required for this scheme). The key sizes we have there are usual for cryptosystems based on LWE:

- Private key size is $\Theta(n)$ elements of $\mathbb{Z}_q$
- Public key size is $\Theta(n^2)$ elements of $\mathbb{Z}_q$
- Ciphertext size is $\Theta(n)$ elements of $\mathbb{Z}_q$

The quadratic public key size is still not competitive compared to schemes based on other popular problems. However, we will now see how to decrease this key size further, under slightly stronger security assumptions.

Chapter 5

Ideal lattices and Ring LWE

5.1 Motivation

Standard LWE can be used to build cryptosystems with practical relevance, like for example in the second round NIST candidate FrodoKEM ([Alk+20]). However, the quadratic key sizes and processing time are not as efficient as one can achieve by using other approaches. This can be fixed by introducing additional structure to the lattices. The most widespread idea is to switch to so called anticyclic lattices, where for each lattice point $(x_1, ..., x_n)$ the anticyclic rotation $(x_2, ..., x_n, -x_1)$ is also a lattice point. Of course, this requires stronger security assumption (i.e. that the problems are even hard if restricted to lattices with the certain property), but using the corresponding variant of LWE, called Ring Learning with Errors, leads to schemes with only linear key sizes and quasilinear runtime. In this section, we will show how this can be done, and we will also see an algebraic generalization of the anticyclic lattices, the so-called ideal lattices.

5.2 Remark (Ideals are lattices)

Let $\mathcal{O}_K$ be the ring of integers in a number field $K = \mathbb{Q}(\alpha)$. Then a fractional ideal $I \leq \mathcal{O}_K$ is a free $\mathbb{Z}$-module of rank $n = [K : \mathbb{Q}]$ as shown for example in [Neu92, p. 2.10]. Therefore, I is $\mathbb{Z}$-module isomorphic to $\mathbb{Z}^n$, so we can identify I with a lattice in $\mathbb{R}^n$. However, the additive structure of any lattice is always $\mathbb{Z}^n$, so this is quite boring. Lattices only become interesting when we introduce a norm. So when we work with ideals as lattices, the concrete identification of ideals I with lattices in $\mathbb{R}^n$ matters, as this induces a norm on K resp. I.

5.3 Remark (Embeddings and the norm)

K is a $\mathbb{Q}$-vector space of dimension $n = [K : \mathbb{Q}]$, so we can find vector space embeddings (injective homomorphisms) $\sigma : K \to \mathbb{R}^n$ with $\sigma(\mathcal{O}_K)$ is a lattice in $\mathbb{R}$. Then also $\sigma(I)$ is a lattice for a fractional ideal $I \leq \mathcal{O}_K$, and lattices of this form are called ideal lattices. Therefore, ideal lattices are special in so far that they have a product induced by the multiplication in K, even though this can be quite well hidden by the transformation through σ.

One obvious way to choose σ, which we will call the "naive embedding", is to choose the coefficients in the basis representation of a value, i.e.

$$\sigma' : K \to \mathbb{R}^n, \quad \sum_{i=0}^{n-1} a_i \alpha^i \mapsto (a_i)_i$$

[LPR12] have used another embedding, the canonical embedding from algebraic number theory. Since a number field K is separable, there are exactly n $\mathbb{Q}$-field homomorphisms $\tau : K \to \mathbb{C}$. These homomorphisms are either real (i.e. $\tau(K) \subseteq \mathbb{R}$) or complex. The complex homomorphisms come in pairs $\tau, \overline{\tau}$, where $\overline{\tau} = \bar{\cdot} \circ \tau$ is the complex conjugate of τ. Let the number of real homomorphisms be s_1 and the number of pairs of complex homomorphisms be s_2. Then can define the canonical embedding

$$\sigma : K \to K_{\mathbb{R}} \cong \mathbb{R}^{s_1+2s_2}, \quad \alpha \mapsto (\tau(\alpha))_\tau$$

where $K_{\mathbb{R}}$ is the Minkowski-Space, that can be characterized as

$$K_{\mathbb{R}} := \left\{ x \in \mathbb{C}^{\mathrm{Hom}_{\mathbb{Q}}(K,\mathbb{C})} | x_\rho \in \mathbb{R}, x_{\overline{\tau}} = \overline{x_\tau} \right\}$$

where ρ runs through the real homomorphisms and τ runs through the complex ones. For more information on Minkowski theory, see e.g. [Neu92]. On $K_{\mathbb{R}}$, we have the inner product $\langle \cdot, \cdot \rangle$ from $\mathbb{C}^n$, which induces a norm on K and we can transfer this norm on $\mathbb{R}^{s_1+2s_2}$ by using an orthonormal basis. A nice property of this embedding is that for $x, y \in K$ we have $\sigma(xy) = \sigma(x)\sigma(y)$, where the second product is component-wise. In particular, we get that $\|xy\| \leq \|x\|\|y\|$. Using this embedding, [LPR12] have shown the results from this section in a more general context (for arbitrary cyclotomic number fields). We however will focus on the case of cyclotomic number fields of a power-of-two degree, as this is most relevant for cryptographic applications.

5.4 Definition

For this whole section, let n be a power of two, $f = \Psi_n = X^n + 1$ be the $2n$-th cyclotomic polynomial and $K = \mathbb{Q}[\zeta] \cong \mathbb{Q}[X]/(f)$ be the corresponding number field with ring of integers $\mathcal{O}_K$, where $\zeta = \exp(\pi i/n)$ is a $2n$-th root of unity.

One reason why this is an important special case is the following result that shows that both previously discussed embeddings lead to the same norm on K (except for a fixed factor $\sqrt{n}$).

5.5 Proposition (Equivalence of ideal lattice norms)

For $\alpha = \sum c_i \alpha^i \in K$, we have that $\sqrt{n}\|\sigma'(\alpha)\| = \sqrt{n}\|(c_i)_i\| = \|\sigma(\alpha)\|$ where

$$\sigma' : K \to \mathbb{R}^n, \quad \sum_{i=0}^{n-1} c_i \zeta^i \mapsto (c_i)_i$$

is the "naive embedding" and

$$\sigma : K \to K_{\mathbb{R}}, \quad \zeta \mapsto (\tau(\zeta))_\tau$$

is the canonical embedding.

Because of this result, we sometimes will not distinguish between the concrete norms.

Proof. Let $\tau_1, ..., \tau_n \in \mathrm{Aut}_{\mathbb{Q}}(K)$ be the $\mathbb{Q}$-Automorphisms of K. By using $K \subseteq \mathbb{C}$, they are also exactly the $\mathbb{Q}$-embeddings of K. Consider the transformation

$$f : \mathbb{C}^n \to \mathbb{C}^n, \quad (c_i)_i \mapsto \frac{1}{\sqrt{n}}(\sigma_i \left(\sum_{i=0}^{n-1} c_i \zeta^i \right))_i$$

We have to show that f is an isometry, since by identifying $K_{\mathbb{R}}$ with a subset of $\mathbb{C}^n$, we obviously have that

$$f\big|_{\sigma'(K)} = \sigma \circ \sigma'\big|_{\sigma'(K)}^{-1}$$

Define the matrix $A = (a_{ij}) \in \mathbb{C}^{n \times n}$ by $a_{ij} = \frac{1}{\sqrt{n}}\sigma_i(\zeta^j)$. Then f is linear and represented by the matrix A, so $f(x) = Ax$ because

$$f(c)_j = \frac{1}{\sqrt{n}}\sigma_j \left(\sum_i c_i \zeta^i \right) = \frac{1}{\sqrt{n}} \sum_i c_i \sigma_j(\zeta^i) = \sum_i c_i a_{ij} = (Ac)_j$$

On the other hand, it holds that the i, j-th entry of $A^\dagger A$ is

$$\begin{aligned} &\sum_{k=0}^{n-1} \overline{a_{ki}} a_{kj} = \frac{1}{n} \sum_{k=0}^{n-1} \overline{\sigma_k(\zeta^i)} \sigma(\zeta^j) = \frac{1}{n} \sum_{k=0}^{n-1} \sigma_k(\zeta^{-i}) \sigma(\zeta^j) \\ =& \frac{1}{n} \sum_{k=0}^{n-1} \sigma_k(\zeta^{j-i}) = \frac{1}{n} \mathrm{Tr}(\zeta^{i-j}) \end{aligned}$$

And since $\mathrm{Tr}(\zeta^i) = 0$ if $i \neq 0$ and $\mathrm{Tr}(\zeta^i) = n$ if $i = 0$, we get that $A^\dagger A = I$ and so f is a linear isometry. □

5.6 Definition (Algebraic structures)

For the definition of Ring LWE and the results, we have to consider some algebraic structures:

- Define the ring $R = \mathcal{O}_K = \mathbb{Z}[\zeta]$ and $R_q = R/qR = \mathbb{Z}_q[\zeta]$.
- Have the Minkowski-Space (it is a ring) $K_{\mathbb{R}}$ with the ring-embedding $\sigma : K \to K_{\mathbb{R}}$ as above.
- Define the $\mathbb{Z}$-module $\mathbb{T}_q := K_{\mathbb{R}}/\sigma(qR)$ with a norm-like operation as in 2.5, i.e. for $x \in \mathbb{T}_q$ have $\|t\| := \min\{\|x\| | x \in \mathbb{R}[X]/(f), t = x \bmod qR\}$ the norm of the shortest representative in $K_{\mathbb{R}}$.
- Have the R-module K/qR, which embeds into $\mathbb{T}_q$ by taking $\sigma : K \to K_{\mathbb{R}}$ "mod q"

  $$\sigma : K/qR \to K_{\mathbb{R}}/\sigma(qR), \quad [x] \mapsto [\sigma(x)]$$

  This σ is a R-module homomorphism. We also use the norm on K/qR induced by this embedding σ. We also embed the ring R_q into K/qR, and use the induced norm on R_q.
- For a fractional ideal I of R have the R-modules I/qI and I^{-1}/qI^{-1}. Also have the multiplication

  $$I/qI \times I^{-1}/qI^{-1} \to R_q, \quad ([x], [y]) \mapsto [xy]$$

- For a fractional ideal I, also define the "dual" ideal $I^\vee := \{x \in R | \mathrm{Tr}(x\bar{I}) \subseteq \mathbb{Z}\}$ whose image under σ is the dual lattice $\sigma(I)^*$, since $\langle \sigma(x), \sigma(y) \rangle = \mathrm{Tr}(x\bar{y})$. We have $I^\vee = R^\vee I^{-1}$, and $R^\vee$ is also called the codifferent ideal. If n is a power of two, we have that $R^\vee = \frac{1}{n}R$, so we will no go into the properties of $I^\vee$ in detail (for arbitrary cyclotomic number fields, this is important, see [LPR12]).

5.7 Definition (Ring LWE)

Analogue to normal LWE, we define a discrete and a continuous version of Ring LWE:

Continuous For a probability distribution χ on $\mathbb{T}_q$, we define the Ring LWE distribution $A_{s,\chi}$ as the probability distribution that results from choosing an $a \in R_q$ uniformly, $e \sim \chi$ independently and outputting $(a, \sigma(as) + e)$.

Discrete For a probability distribution χ on R_q, we define the $A_{s,\chi}$ as the distribution that results from choosing an $a \in R_q$ uniformly, $e \sim \chi$ independently and outputting $(a, as + e)$.

Then the (average case) Ring Learning with Errors problem $\mathrm{RLWE}_{q,\chi}$ is: For a uniformly chosen $s \in R_q$, given samples from $A_{s,\chi}$, find s.

5.8 Remark (RLWE Error)

In the case of LWE, usually we used Ψ_α as the error distribution. In RLWE, this would correspond to an error $e = \sum e_i \zeta^i \in \mathbb{T}_q$ where $e_0, ..., e_{n-1} \sim \Psi_\alpha$ are iid.. However, the hardness proof 4.4 does not work with this distribution, since this distribution is not closed with respect to multiplication with a constant element in K. Instead, we need elliptical gaussian distributions. For $\alpha \in \mathbb{R}^n, \alpha \geq 0$ define the elliptical gaussian distribution ν_α on $K_\mathbb{R}$ with results from choosing each component $e_\tau \sim \nu_{\alpha_i}$.

Then we have that for $e \sim \nu_\alpha$ and $x \in K$, the product $\sigma(x)e$ is distributed according to $\nu_{\alpha\sigma(x)}$ (using component-wise multiplication) since the canonical embedding transforms multiplication in K to component-wise multiplication in $K_\mathbb{R}$. In particular, if $\alpha_i \leq c$ for all i with $c > 0$, then $(\alpha\sigma(x))_i \leq c\sigma(x)_i \leq c\|\sigma(x)\|_\infty$. We still write ν_s for $s > 0$ when we mean the distribution $\nu_{(s)_i}$ on R with the error vector $(s)_i \in \mathbb{R}^n$.

Continuous Error We now can use this distribution modulo q as RLWE error, i.e. define Ψ_α on $\mathbb{T}_q$ as the probability distribution you get from sampling from $\nu_{q\alpha}$ and taking the result $\mod \sigma(qR)$.

Discrete Error For the discrete variant of RLWE, we also need a discrete error distribution on R_q. Define $\bar{\Psi}_\alpha$ as the probability distribution that results from sampling $e \sim \nu_{q\alpha}$ and outputting $\sigma^{-1}(\lfloor e \rceil) \bmod qR$, where we round $e \in K_\mathbb{R}$ by $\lfloor e \rceil = \kappa_{\sigma(R)}(e)$. Usually, calculating $\kappa_L(x)$ is hard, but in the case $L = \sigma(R)$, it can be done efficiently:

We can extend $\sigma : \mathbb{Q}[X]/(f) = K \to K_\mathbb{R}$ to the ring homomorphism

$$\sigma : \mathbb{R}[X]/(f) \to K_\mathbb{R}, \quad \left[\sum a_i X^i\right] \mapsto \left(\sum a_i \zeta^{ik}\right)_{\tau_k}$$

with a root of unity $\zeta = \exp(\pi i/n) \in \mathbb{C}$ and the field embeddings $\tau_k \in \mathrm{Hom}_{\mathbb{Q}}(K, \mathbb{C})$ with $\tau(\zeta) = \zeta^k$. This extension is bijective, as a polynomial of degree $< n$ is unique characterized by its value at n different points. As shown in 5.5, the "naive" norm on $\mathbb{R}[X]/(f)$ is (up to a factor) equal to the norm on $K_{\mathbb{R}}$, so using that extension, we have that $\kappa_{\sigma(R)}(x) = \sigma(\kappa_R(\sigma^{-1}(x)))$. We can easily calculate $\kappa_R(y)$ by rounding each of the coefficients in the linear combination $y = [\sum a_i X^i] \in \mathbb{R}[X]/(f)$.

5.9 Lemma

Let $\tau : K \to K$ be a $\mathbb{Q}$-automorphism of K. We have that $\sigma \circ \tau \circ \sigma^{-1} = \pi_\tau\big|_{\sigma(K)}$ where π_τ is a permutation of the vector components in $K_{\mathbb{R}}$ as

$$\pi_\tau : K_{\mathbb{R}} \to K_{\mathbb{R}}, \quad (x_\rho)_\rho \mapsto (x_{\rho\circ\tau})_\rho$$

Furthermore, for $e \in K_{\mathbb{R}}$ it holds that $\pi_\tau(\lfloor e \rceil) = \lfloor \pi_\tau(e) \rceil \in \sigma(R)$.

Proof. For $x \in K$ and a $\mathbb{Q}$-embedding $\rho : K \to K_{\mathbb{R}}$, we have

$$(\sigma \circ \tau \circ \sigma^{-1})(\sigma(x))_\rho = \sigma(\tau(x))_\rho = \rho(\tau(x)) = \sigma(x)_{\rho\circ\tau} = \pi_\tau(\sigma(x))_\rho$$

For $e \in K_{\mathbb{R}}$, we have

$$\lfloor \pi_\tau(e) \rceil = \kappa_{\sigma(R)}(\pi_\tau(e)) = \pi(\kappa_{\pi_\tau^{-1}(\sigma(R))}(\pi_\tau^{-1}(\pi(e)))) = \pi_\tau(\kappa_{\pi_\tau^{-1}(\sigma(R))}(e))$$

since π_τ is an isometry, so $\pi(\kappa_L(x)) = \kappa_{\pi(L)}(\pi(x))$. It is left to show that $\pi_\tau^{-1}(\sigma(R)) = \sigma(R)$, or equivalently that $R = \sigma^{-1}(\pi_\tau(\sigma(R)))$. This holds, since $\tau(R) = R$ and $\sigma \circ \tau \circ \sigma^{-1} = \pi_\tau\big|_{\sigma(K)}$. □

5.10 Definition

Also consider a variant of RLWE with unknown, bounded error distributions as we did for LWE. For $\alpha > 0$, define $\mathrm{RLWE}_{q,\leq\alpha}$ as the following problem: For a uniformly chosen $s \in R_q$ and any $\beta \in \mathbb{R}^n, \beta \geq 0$ and $\beta_i \leq \alpha$ for all i, find s given samples from A_{s,Ψ_β}.

5.11 Remark (Connection RLWE and LWE)

Ring LWE can be seen as a special case of LWE, where the samples come in batches. A ring LWE sample $(a, as + e)$ with $a = \sum a_i \zeta^i \in R_q$ and $e \sim \Psi_\alpha$ where $\alpha_1 = ... = \alpha_n = \beta > 0$ with secret $s = \sum s_i \zeta^i \in R_q$ can also be written as

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}, \begin{pmatrix} a_0 & -a_{n-1} & -a_{n-2} & ... & -a_1 \\ a_1 & a_0 & -a_{n-1} & & -a_2 \\ \vdots & & & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & ... & a_0 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix}$$

by using the the polynomial multiplication

$$as = \sum_{i,j} a_i s_j \zeta^{i+j} = \sum_{i,j} a_i s_j \zeta^{i+j \bmod q} = \sum_i \zeta^i \sum_k s_k a_{i-k \bmod q}$$

Since σ is a linear isometry and all components of e are independent and distributed according to the same normal distribution, we get that if $e = \sigma\left(\sum e_i' \zeta^i\right)$ all e_i' are iid. and distributed like $\Psi_{\alpha/n}$. Therefore this single RLWE sample is really n LWE samples with all anticyclic rotations of $(a_{n-1}, ..., a_0)$ as first sample parts. This is also the main reason why RLWE is more efficient than normal LWE: We required a factor of n less samples, so the space and time requirements are significantly lower. It also leads to the following proposition:

5.12 Proposition (RLWE to LWE reduction)

Let $\beta = \frac{1}{n^2}\sum_k \alpha_k$. Then

$$\mathrm{RLWE}_{q,\Psi_\alpha} \leq_p \mathrm{LWE}_{q,\Psi_\beta}$$

Proof. Receive RLWE samples $(a = \sum_j a_j \zeta^j, b = \sigma(\sum_j b_j \zeta^j) = \sigma(as) + e)$ and give (a', b') to the oracle where $a' = (a_{n-1}, ..., a_0)^T$ and $b' = b_0 \in \mathbb{R}/q\mathbb{Z}$. Then (a', b') is a valid LWE sample for secret $s' = (s_0, ..., s_{n-1})^T$ with $s = \sum_j s_j \zeta^j$, since

$$as = \sum_{i,j} a_i s_j \zeta^{i+j} = \sum_i \zeta^i \sum_k s_k a_{i-k} = \sum_i \zeta^i \sum_k a'_{k+i} s'_k$$

so $b_0 = \sum a'_k s'_k + e_0 = \langle a', s' \rangle + e_0$, where we define $a_{j-n} = a_{j+n} = -a_j$ for $j \in \{0, ..., n-1\}$. Therefore, when the oracle returns s', we return $s = \sum s'_i \zeta^i$.

The Error Have $e = \sigma\left(\sum_j e_j \zeta^j\right) \sim \Psi_\alpha$ (extend σ to $\sigma : \mathbb{R}/(f) \to K_\mathbb{R}$). Then $e_0, ..., e_{n-1}$ are gaussian random variables, and

$$\mathrm{Cov}(e_i, e_j) = \frac{1}{n^2}\left(A^\dagger \mathrm{diag}(\alpha) A\right)_{ij} = \frac{1}{n^2}\sum_{k,l} \tau_l\left(\zeta^i\right) \delta_{lk} \alpha_l \overline{\tau_k\left(\zeta^j\right)} = \frac{1}{n^2}\sum_k \tau_k\left(\zeta^{i-j}\right) \alpha_k$$

by using A and $\tau_0, ..., \tau_{n-1}$ as in 5.5, since σ^{-1} is linear and represented by the matrix $\frac{1}{n}A^\dagger$ in the basis $\zeta^0, ..., \zeta^{n-1}$. In particular, $\mathrm{Var}(e_0) = \frac{1}{n^2}\sum_k \alpha_k$, so $e_0 \sim \Psi_\beta$. Since $e_1, ..., e_{n-1}$ are discarded, it does not matter that they might not be independent. □

5.13 Remark (Unknown to constant error variance reduction)

If the error distribution Ψ_β is spherical, i.e. $\beta_1 = ... = \beta_n \leq \alpha$ for some $\alpha > 0$, then we can solve Ring LWE for an unknown error distribution Ψ_β when given an oracle for $\mathrm{RLWE}_{q,\Psi_\alpha}$ by “guessing” the right amount of error we have to add to β to be close enough to α, as we did in 4.3. However, if Ψ_β is elliptical (i.e. $\beta_i \neq \beta_j$ in general), we would have to “guess” correctly each component of β, which is not possible to do in polynomial time. Therefore, the gap between the unknown error variant and the standard RLWE problem is not easy to cross.

In fact, [LPR12] could only prove a reduction to a variant of Ring LWE, where the error distribution is unknown, but each component of the error distribution variance is again random and distributed according to the gamma distribution $\Gamma(2,1)$. This is more like an average-case version of the unknown error distribution RLWE problem, but at least it can be implemented. Additionally, if the number of samples an adversary has access to is fixed, then it is possible to restrict the problem to one fixed, spherical error distribution $\Psi_{(\alpha)_i}$ with slightly larger variance $\alpha > 0$.

5.14 Remark

Most hardness results for LWE still hold, namely the average to worst case reduction 3.7, the reduction to SIVP 4.4 and that we can draw the secret from the error distribution 3.10. That 3.7 and 3.10 still hold is quite easy to see, as we only add and invert the inputs, and R_q is closed under these operations. In the reduction 3.7, we have to switch to SIVP and BDD on ideal lattices. We will sketch how to modify this reduction so that it works with RLWE.

One important result however, the decision to search reduction 3.9 cannot be adjusted to work with RLWE due to the way we modify the samples. A corresponding result for ideal lattices has been proven by [LPR12], and we will present it at the end of this section.

5.15 Proposition (SIVP to RLWE reduction)

For $\alpha q \geq 2\sqrt{n}$, there is a quantum reduction from worst case $\mathrm{RLWE}_{q,\leq n\alpha}$ to worst case $\mathrm{SIVP}_{2n\sqrt{2n}/\alpha}$ on ideal lattices.

This can be proven exactly as in 4.4, except that we need the modified version of the lemma:

5.16 Lemma (Improvement of gaussian samples using RLWE)

Let ϵ be negligible, $\alpha \in (0,1)$ and $r > \sqrt{2}q\eta_\epsilon(L)$ for an ideal lattice $L = \sigma(I)$, where I is a fractional ideal of R. Given an oracle for $\mathrm{RLWE}_{q,\leq n\alpha}$ and samples from $D_{L,r}$ (as many as the oracle needs), we can efficiently generate samples from $D_{L,r\sqrt{n}/\alpha q}$. The generated samples are stochastically independent of the input samples and of each other.

Proof. We will only prove the first step, as the second and third step are exactly as in 4.5.

5.16.1 Construction of $\mathrm{BDD}^q_{L^*,\alpha q/r\sqrt{2}}$ oracle

We now want to do the same operations as in 4.5. Therefore, we need a basis of I that corresponds to B. Since R is a Dedekind domain, the R-module I/qI is generated by a single element $t \in I/qI$, and we also have that $I^{-1}/qI^{-1} = (t^{-1})$ (we can calculate t efficiently using a known factorization of q). In addition to that, the multiplication by t given as $m_t : R \to I/qI, x \mapsto tx$ is a module homomorphism, and by the homomorphism theorem, we get that $m'_t : R_q \to I/qI, [x] \mapsto [tx]$ is an isomorphism since $\ker(m_t) = qR$. This t corresponds to the basis matrix B in 4.5.

For a given $x \in K$ with $\|x - \kappa(x)\| \leq \alpha q / r\sqrt{2}$ where $\kappa(x) \in I^\vee$ and iid. random variables $v_i \sim D_{I,r}$ representing the samples, we define

$$\begin{aligned} s &= t(n\kappa(x) \bmod qI^{-1}) \in R_q \quad \text{since} \quad n\kappa(x) \in I^{-1} \\ a_i &= t^{-1}(v_i \bmod qI) \in R_q \end{aligned}$$

We get that

$$\begin{aligned} a_i s &= t^{-1}(v_i \bmod qI) t(n\kappa(x) \bmod qI^{-1}) = (v_i \bmod qI)(n\kappa(x) \bmod qI^{-1}) \\ &= n v_i \kappa(x) \bmod qR = n v_i x + n v_i(\kappa(x) - x) \bmod qR \\ &= (v_i n x \bmod qR) + (v_i n(\kappa(x) - x) \bmod qR) \in K/qR \end{aligned}$$

By 2.12, it holds that a_i is distributed uniformly on R_q.

Now we condition on a fixed a_i. It follows that v_i/q is distributed according to $D_{I + ta_i/q, r/q}$ (take any representative in I for ta_i). By choosing $e \sim \nu_{r/q}$, lemma 2.13 yields that $\sigma(v_i/q) + e$ is within statistical distance 4ϵ to $\nu_{r\sqrt{2}/q}$, since

$$\frac{(r/q)(r/q)}{\sqrt{(r/q)^2 + (r/q)^2}} = r/q\sqrt{2} \geq \sqrt{2} q \eta_\epsilon(I)/q\sqrt{2} = \eta_\epsilon(I)$$

Therefore, $n\sigma(x - \kappa(x))(\sigma(v_i) + qe) = nq\sigma(x - \kappa(x))(\sigma(v_i/q) + e)$ is distributed according to $\nu_{q\beta}$ (as described in 5.8) where

$$q\beta_i \leq n\|\sigma(x - \kappa(x))\|_\infty r\sqrt{2} \leq n\|x - \kappa(x)\| r\sqrt{2} \leq n\alpha q$$

As a result, $n\sigma(x - \kappa(x))(\sigma(v_i) + e) \bmod \sigma(qR) \in \mathbb{T}_q$ is distributed like Ψ_β, so our oracle can retrieve s when we call it on the samples $(a_i, n\sigma(x - \kappa(x))(\sigma(v_i) + e) \bmod \sigma(qR))$. □

5.17 Remark

In 5.15, we required an oracle for $\text{RLWE}_{q,\leq n\alpha}$, while the non-ring case 4.4 only required an oracle for $\text{LWE}_{q,\leq\alpha}$. The factor n in the error amount is an artifact of our definition of Ring LWE. The inverse of this factor appears in the connection between LWE and RLWE 5.12, so the reduction works as "tight" for RLWE as for LWE.

When we take the dual lattice, the product $\langle a_i, s\rangle$ should be integral when choosing $a_i \in L, s \in L^*$. In the case of LWE, this works perfectly, since we draw a_i and s from $\mathbb{Z}^n$ (we ignore the modulo q now), and the dual lattice of $\mathbb{Z}^n$ is $\mathbb{Z}^n$. In the RLWE case however, when we choose s from $R = \mathbb{Z}[\zeta]$, we can choose a_i from $R^\vee$ and get $\langle \sigma(a_i), \sigma(s)\rangle = \text{Tr}(a_i \bar{s}) \in \mathbb{Z}$. In particular, we have for example $\frac{1}{n} \in R^\vee$ since $\text{Tr}(\frac{1}{n}) = n\frac{1}{n} = 1$. As a result, taking s and a_i both from R in RLWE does not exactly match the structure of ideal lattices. For this reason, [LPR12] defined RLWE to draw the secret from $R^\vee$, and in this case, the factor n disappears (they also consider the non power-of-two case, where this is not possible anyway). For this work, we still take $a_i, s \in R$, as this causes $a_i s \in R_q$ and so we can easily define the discrete variant of RLWE.

5.18 Remark (Algebraic properties of the Ring R_q)

To prove the search to decision reduction, the embedding σ is not of great importance anymore, as we won't deal with ideal lattices anymore. Instead, we need to examine the structure of the ring R_q, in which all RLWE operations are performed.

As already mentioned, we have that $K = \mathbb{Q}[\zeta] = \mathbb{Q}[X]/(f)$ with $f = X^n + 1$ is the $2n$-th cyclotomic polynomial. In particular, $\mathbb{Q}(\zeta)/\mathbb{Q}$ is galois, and so $|\mathrm{Aut}_{\mathbb{Q}}(\mathbb{Q}(\zeta))| = n$ and $\mathrm{Gal}(\mathbb{Q}(\zeta)/\mathbb{Q}) \cong \mathbb{Z}_{2n}^*$.

We also have that the ring of integers $\mathcal{O}_{\mathbb{Q}(\zeta)} = \mathbb{Z}[\zeta]$. Now consider a $\mathbb{Q}$-automorphism $\tau \in \mathrm{Aut}_{\mathbb{Q}}(\mathbb{Q}(\zeta))$. Then τ is uniquely determined by the value $\tau(\zeta)$, and it must hold $\tau(\zeta) = \zeta^k$ for some fixed $k \in \mathbb{Z}_{2n}^*$. We get that

$$\tau\big|_{\mathbb{Z}[\zeta]} : \mathbb{Z}[\zeta] \to \mathbb{Z}[\zeta] \text{ is a ring automorphism}$$

Proof. $\tau(\mathbb{Z}[\zeta]) \subseteq \mathbb{Z}[\zeta]$, since $\tau(\sum_i a_i\zeta^i) = \sum_i a_i\tau(\zeta)^i = \sum_i a_i\zeta^{ik} \in \mathbb{Z}[\zeta]$ for some fixed $k \in \mathbb{Z}_{2n}^*$ and all $a_i \in \mathbb{Z}$. Injectivity and the homomorphism property follow directly because τ is a field automorphism. Surjectivity follows since $\tau^{-1}(\mathbb{Z}[\zeta]) \subseteq \mathbb{Z}[\zeta]$ by the same argument as in the well definedness. □

To apply these facts to our ring R_q, we now consider these automorphisms modulo (q). We will implicitly use the relation $(\mathbb{Z}[X]/(f))/(q) \cong (\mathbb{Z}[X]/(q))/(f) \cong \mathbb{Z}_q[X]/(f)$. So have

$$\tau' : \mathbb{Z}_q[\zeta] \to \mathbb{Z}_q[\zeta], \quad [x]_q \mapsto [\tau(x)]_q \text{ is a ring automorphism}$$

Proof. For $[x]_q = [z]_q \in \mathbb{Z}_q[\zeta]$ get that $\tau(x) - \tau(y) = \tau(x-y) \in \tau((q)) = \tau(q)\tau(\mathbb{Z}_q[\zeta]) \subseteq (q)$, so $[\tau(x)]_q = [\tau(y)]_q$ and we get that τ' is well-defined. Injectivity follows in an analogous way, since if we have $[\tau(x)]_q = [\tau(y)]_q \in \mathbb{Z}_q$, then $[\tau^{-1}(\tau(x))]_q = [\tau^{-1}(\tau(y))]_q$ and so $[x]_q = [y]_q$. To finish, we get surjectivity since $\mathbb{Z}_p[\zeta]$ is finite and obviously, τ' is a ring homomorphism. □

Structure of τ' Now have a look at one last interesting property of all τ': For each $k \in \mathbb{Z}_{2n}^*$, there is exactly one $\tau \in \mathrm{Aut}_{\mathbb{Q}}(\mathbb{Q}(\zeta))$ with $\tau(\zeta) = \zeta^k$. By definition, we get that $\tau'([\zeta]) = [\zeta^k] = [\zeta]^k$ and so

$$\tau'\left(\sum_{i=0}^{n-1} a_i[\zeta]^i\right) = \sum_{i=0}^{n-1} \tau'(a_i)[\zeta]^{ik} = \sum_{i=0}^{n-1} a_i[\zeta]^{ik} = \sum_{i=0}^{n-1} a_{\phi(i)}[\zeta]^i$$

where

$$\phi : \mathbb{Z}_{2n} \to \mathbb{Z}_{2n}, x \mapsto k^{-1}x$$

is a group isomorphism and we consider $a_{n+j} = -a_j$ for $j \in \{0, ..., n-1\}$. We even have the property that

$$\phi' : \mathbb{Z}_{2n}/\langle n\rangle \to \mathbb{Z}_{2n}/\langle n\rangle, [x] \mapsto [k^{-1}x]$$

is a group isomorphism (since $\phi(\langle n\rangle) = \langle kn\rangle = \langle kn \bmod 2n\rangle = \langle n\rangle$ as $k \equiv 1 \bmod 2$), so τ' only permutes and possibly negates the coefficients in the linear combination of $\zeta^0, ..., \zeta^{n-1}$ of any $x \in \mathbb{Z}_p[\zeta]$.

5.19 Remark (The Chinese Remainder Representation)

Any polynomial $f \in \mathbb{Z}_q[Z]$ of degree $\leq n$ is uniquely determined by the values $f(x_i)$ for different $x_0, ..., x_n \in \mathbb{Z}_q$. Conversely, for different $x_0, ..., x_n \in \mathbb{Z}_q$ and $y_0, ..., y_n \in \mathbb{Z}_q$, there is exactly one polynomial f of degree $\leq n$ with $f(x_i) = y_i$. To use this, we first will have to look at $2n$-th roots of unity again (but they are different ones now):

Let now q be a prime with $q \equiv 1 (\text{mod } 2n)$ and n be a power of two. Since $\mathbb{Z}_q^*$ is cyclic, there must be an element of order $2n | q-1$. Let ξ be such a element, so ξ is a $2n$-th root of unity in $\mathbb{Z}_q$. We also have that the other $2n$-th roots of unity are exactly ξ^k for $k \in \mathbb{Z}_{2n}^*$. Since $\phi(2n) = n$, there are exactly n of them. Therefore, we can represent a polynomial $f \in \mathbb{Z}_q$ of degree $\leq n-1$ by the values $f(\xi^k)$ for $k \in \mathbb{Z}_{2n}^*$. By identifying elements in $\mathbb{Z}_q[x]/(X^n+1)$ with their representatives in $\mathbb{Z}_q[X]$ of degree $\leq n-1$, we get the evaluation at all primitive $2n$-th roots of unity in $\mathbb{Z}_q$ as:

$$\mathcal{F} : \mathbb{Z}_q[X]/(X^n+1) \to H, [f] \mapsto (f(\xi^k))_{k \in \mathbb{Z}_{2n}^*} \quad \text{where} \quad H := \mathbb{Z}_q^{\mathbb{Z}_{2n}^*}$$

Then this $\mathcal{F}$ is a ring isomorphism (H becomes a ring using element-wise addition and multiplication), and we sometimes call its image the “Chinese Remainder representation”, as $\mathcal{F}$ is a possible isomorphism for the isomorphy given by the Chinese Remainder theorem for rings:

$$R_q \cong \mathbb{Z}_q[X]/(X^n+1) \cong \bigoplus_{k \in \mathbb{Z}_{2n}^*} \mathbb{Z}_q[X]/(X - \xi^k)$$

$\mathcal{F}$ is also a variant of the Fourier Transformation (usually for $\xi \in \mathbb{C}$) or in this case Number Theoretic Transformation (since $\xi \in \mathbb{Z}_q$), which is the evaluation at all (including non-primitive) n-th roots of unity. As a last point, $\mathcal{F}$ has a strong connection to the canonical embedding $\sigma : K \to K_{\mathbb{R}}$, which corresponds to the evaluation at all primitive $2n$-th roots of unity in $\mathbb{C}$ and not in $\mathbb{Z}_q$. However, $\mathcal{F}$ is no isometry in the sense of 5.5, otherwise Ring LWE would be easy to solve (since then the error would also be small in the CRT representation).

Notation We write $\mathcal{F}f_k$ for $\mathcal{F}(f)(k)$ and for $x = (x_1, ..., x_m)^T \in R_q^k$, have $\mathcal{F}x_k := (\mathcal{F}(x_1)_k, ..., \mathcal{F}(x_m)_k)^T \in \mathbb{Z}_q^m$.

Proof. $\mathcal{F}$ is well-defined, since for $f, g \in \mathbb{Z}_q[X]$ with $f - g \in (X^n+1)$, we have $f(\xi^k) - g(\xi^k) = (f-g)(\xi^k) = 0$ since $\xi^{kn} = -1$ for $k \perp 2$. Bijectivity follows from the observation at the beginning of this section, and $\mathcal{F}$ is a ring homomorphism, as each component $(\mathcal{F}\cdot)_k$ really is the evaluation of the polynomial at ξ^k. □

5.19.1 Remark

Both 5.18 and 5.19 were about roots of unity, but the choice of different symbols (ζ and ξ) indicated that they are not the same. For a power of two n and a prime q with $q \equiv 1 (\text{mod } 2n)$, there have n roots of unity in the field $\mathbb{Z}_q$, which are exactly the ξ^k for $k \in \mathbb{Z}_{2n}^*$. On the other hand, in 5.18, we had some arbitrary ring R (it was $R = \mathbb{Z}$), and we added $2n$-th roots of unity by considering $R[\zeta]$, i.e. $R[X]/(X^n+1)$ with $\zeta = [X]$.

So if the condition $q \equiv 1 (\text{mod } 2n)$ is fulfilled, we really have $2n$ (primitive) $2n$-th roots of unity in the ring $R_q = \mathbb{Z}_q[X]/(X^n+1)$, namely ξ^k and ζ^k for $k \in \mathbb{Z}_{2n}^*$. How it be possible that the equation $X^n = -1$ has $2n$ solutions? Because in this case, the ring $R_q = \mathbb{Z}_q[X]/(X^n+1)$ is not an integral domain.

This “unity root duality” also is why $(\mathcal{F}\zeta)_k = \xi^k$ holds for $k \in \mathbb{Z}_{2n}^*$.

5.20 Theorem (RLWE search to decision reduction)

For $\alpha > 0$ and $q \in O(\text{poly}(n))$, have

$$\text{Search RLWE}_{q,\leq\alpha} \leq_p \text{Decision RLWE}_{q,\leq\alpha}$$

Proof. We base this proof on the discrete version of RLWE, as otherwise we would have to extend the CRT isomorphism $\mathcal{F}$ and the ring automorphisms from 5.18 to $\mathbb{T}_q$, which would clutter the proof with even more technical details.

The proof presented here is based on [LPR12]. For simplicity identify $(\mathcal{F}f)_i := (\mathcal{F}f)_{\pi(i)}$ where $i \in \{1, ..., n\}$ and $\pi : \{1, ..., n\} \to \mathbb{Z}_{2n}^*$ is any bijection. We will use the algebraic properties in more depth during the last part of the proof, but until then it is easier to work with $\mathcal{F}f$ as a vector, and only use that addition and multiplication in $\mathcal{F}$-representation are component-wise.

Have oracle G that distinguishes between LWE samples and uniformly chosen ones. We will prove this proposition in four steps:

- Construct an oracle that can distinguish the hybrid distributions defined by using real LWE values for $(\mathcal{F}b)_1, ..., (\mathcal{F}b)_k$ and choosing the values for $(\mathcal{F}b)_{k+1}, ..., (\mathcal{F}b)_n$ uniformly (where k is fixed).
- Using this, construct an oracle that can distinguish the cases $\mathcal{F}s_k = r$ and $\mathcal{F}s_k \neq r$ when given some $r \in \mathbb{Z}_q$ and LWE samples for a uniformly chosen secret s.
- Construct an oracle that can find $(\mathcal{F}s)_k$ when given LWE samples for an arbitrary secret s.
- Construct an oracle that can find s.

5.20.1 Distinguish hybrids

Let $m \in O(\text{poly}(n))$ be the maximal number of samples the oracle will request, and let $a \in R_q^m,\ u \in R_q^m,\ s \in R_q$ be uniformly distributed and $e \sim \chi^m$ random variables. The oracle G distinguishes the LWE distribution (a, b) (where $b = as + e$) and the distribution (a, u), so we have

$$\Pr[G(a, b) = 1] - \Pr[G(a, u) = 1] = \Delta \geq \frac{1}{\text{poly}(n)}$$

Define the hybrid distributions $H^{(i)}$ over R_q^m by

$$\mathcal{F}H_j^{(i)} = \mathcal{F}b_j \text{ for } j < i$$
$$\mathcal{F}H_j^{(i)} = \mathcal{F}u_j \text{ for } j \geq i$$

We get that $H^{(1)} = u$ and $H^{(n+1)} = as + e$. Therefore, G distinguishes $(a, H^{(1)})$ and $(a, H^{(n+1)})$ with a non-negligible acceptance probability gap Δ/n. By the hybrid argument, we get that G distinguishes $(a, H^{(k+1)})$ and $(a, H^{(k)})$ for some fixed k with non-negligible probability.

5.20.2 Test if one secret position is correct

The idea is to modify the k-th entry of the input samples (in CRT representation) so that it is either uniform or a valid sample entry, depending on whether $\mathcal{F}s_k = r$ or not, and use the hybrid distinguisher to see which is the case. From the previous section, we have:

$$\Pr[G(a, H^{(k+1)}) = 1] - \Pr[G(a, H^{(k)}) = 1] \geq \Delta/n$$

Given some $r \in \mathbb{Z}_q$, define the modified sample distribution (a', b') as

$$\begin{aligned} \mathcal{F}a'_k &= \mathcal{F}a_k + d \\ \mathcal{F}a'_j &= \mathcal{F}a_j && \text{for } j \neq k \\ \\ \mathcal{F}b'_j &= \mathcal{F}b_j && \text{for } j < k \\ \mathcal{F}b'_k &= \mathcal{F}b_k + dr \\ \mathcal{F}b'_j &= \mathcal{F}u_j && \text{for } j > k \end{aligned}$$

where $d \in \mathbb{Z}_q^m$ is a uniformly distributed random variable.

If $\mathcal{F}s_k = r$ then (a', b') are distributed identically to $H^{(k+1)}$, as a' is uniform and

$$\begin{aligned} \mathcal{F}b'_k &= \mathcal{F}b_k + dr = \mathcal{F}a_k \mathcal{F}sk + \mathcal{F}e_k + dr \\ &= (\mathcal{F}a_k + d)\mathcal{F}s_k + \mathcal{F}e_k = \mathcal{F}a'_k \mathcal{F}s_k + \mathcal{F}e_k \end{aligned}$$

Otherwise (a', b') is distributed according to $H^{(k)}$, since $(r - \mathcal{F}s_k)d$ is uniform.

Therefore, we can distinguish the case $\mathcal{F}s_k = r$ and $\mathcal{F}s_k \neq r$ for every $r \in \mathbb{Z}_q$ by creating these modified samples and calling the hybrid distinguisher G on them (with probability gap Δ/n).

5.20.3 Finding one position for worst case secrets

To find $(\mathcal{F}s)_k$, we would like to try all $r \in \mathbb{Z}_q$ and find which is the correct one. For arbitrary inputs however, we have to combine this idea with a standard average-to-worst case reduction. Let G' be the oracle constructed before. We have for every $r \neq \mathcal{F}s_k$ that

$$\Pr[G'((a, b), \mathcal{F}s_k) = 1] - \Pr[G'((a, b), r) = 1] \geq \Delta/n$$

Now consider the following algorithm, where we define $p = \Pr[G'((a, b), \mathcal{F}s_k) = 1]$:

- Repeat often enough
  - Sample $d \sim R_q$ uniformly
  - Get m new input samples (a, b)
  - Call distinguisher on $(a, b + ad)$
- If more than $p - \Delta/2n$ calls accepted, accept

Since $s + d$ is uniform on R_q and independent of s and we get new input samples each time, all repetitions are independent. Therefore, polynomially many repetitions are sufficient to check whether $\mathcal{F}s_k = r$ with probability exponentially close to 1.

Finding $\mathcal{F}s_k$ is easy now: For each r, check if $\mathcal{F}s_k = r$ by calling the procedure, and output the r for which the oracle accepts.

5.20.4 Finding the whole secret

In this section, we will use on the concrete structure of R_q as described in 5.18. The idea is to use the automorphisms to "permute" the positions of the secret, and therefore be able to bring each position to the k-th one. For this section, consider k to be an element of $\mathbb{Z}_{2n}^*$, i.e. $k_{\text{this section}} := \pi^{-1}(k_{\text{last sections}})$. Then the constructed oracle can find $(\mathcal{F}s)_k = s(\xi^k)$ with high probability.

For any $l \in \mathbb{Z}_{2n}^*$, there is a ring automorphism $\tau : R_q \to R_q$ with $\tau(\zeta) = \zeta^l$. Since $(\mathcal{F}\zeta)_k = \xi^k$, we get that

$$(\mathcal{F}\tau(a))_k = \tau(a)(\xi^k) = a(\xi^{lk}) = (\mathcal{F}a)_{lk}$$

As a result, we can find $(\mathcal{F}a)_{lk}$ by applying the oracle on transformed samples $(\tau(a), \tau(b) = \tau(as+e))$. Since $\mathbb{Z}_{2n}^*$ is a group, we can reach any $r \in \mathbb{Z}_{2n}^*$ with $r = lk$ for an appropriate $l \in \mathbb{Z}_{2n}^*$, so we can find all positions of the secret s using this method, and therefore s.

5.20.5 The error distribution

There is one last point we have to pay attention to: Does our oracle work with the transformed samples $(\tau(a), \tau(b))$? The oracle from the last section can work with all valid RLWE samples, so we have to ensure that $(\tau(a), \tau(b))$ is distributed according to $A_{\tau(s),\bar{\Psi}_{\beta'}}$ for some $\beta' \in \mathbb{R}^n, \beta \geq 0$ and $\beta'_i \leq \alpha$ for all i. It is quite obvious that $\tau(A)$ is uniformly random for uniform random variable $A \sim R_q$. So the only potential problem is the error distribution: We have $\tau(b) = \tau(as + e) = \tau(a)\tau(s) + \tau(e)$. This is only a valid part of a RLWE distribution for secret $\tau(s)$, if $\tau(e)$ is a valid error.

Show $\tau(e)$ is distributed according to $\bar{\Psi}_{\beta'}$. Let $e = \sigma^{-1}(\lfloor e' \rceil) \sim \bar{\Psi}_\beta$ be a random variable representing the error, with $e' \sim \Psi_\beta$. Lemma 5.9 yields

$$\sigma(\tau(e)) = \sigma(\tau(\sigma^{-1}(\lfloor e' \rceil))) = \pi_\tau(\lfloor e' \rceil) = \lfloor \pi_\tau(e') \rceil$$

and therefore, $\tau(e)$ is distributed identically to $\sigma^{-1}(\lfloor e'' \rceil)$, where $e'' \sim \Psi_{\beta'}$ is a random variable and β' is the permutation of β given by π_τ. Since β' is a permutation of β, we have $\beta' \geq 0$ and $\beta'_i \leq \alpha$ for all i. As a result, $\tau(e)$ is distributed like $\sigma^{-1}(\lfloor e'' \rceil)$, which has the distribution $\bar{\Psi}_{\beta'}$ by definition. □

5.21 Remark (Key sizes)

The main advantage of RLWE opposed to standard LWE is that each RLWE sample in some way contains the information of n LWE samples (see 5.11). More accurately, $\log q$ RLWE samples are sufficient (if we draw s uniformly) so that the resulting problem has a unique solution, since a sample $(a = \sum a_i \zeta^i, b = \sum b_i \zeta^i)$ does not only contain that

$$\langle \begin{pmatrix} a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}, \begin{pmatrix} s_0 \\ \vdots \\ s_{n-1} \end{pmatrix} \rangle \approx b_{n-1}$$

but also the same for each i, so

$$\langle \begin{pmatrix} a_i \\ \vdots \\ a_{i+1-n} \end{pmatrix}, \begin{pmatrix} s_0 \\ \vdots \\ s_{n-1} \end{pmatrix} \rangle \approx b_i$$

As a result, one RLWE sample is sufficient if we draw the secret from the error distribution as in 3.10. Therefore, RLWE has a private key, public key and ciphertext size of $\Theta(n)$ elements of $\mathbb{Z}_q$, and by using the Chinese Remainder representation, we can also process the data in runtime $\Theta(n)$. The transformation from and to CRT representation itself can be done in $\Theta(n \log n)$ using so called Fast Fourier Transformation algorithms (see also the implementation section).

Chapter 6

Module LWE and Crystals Kyber

6.1 Motivation

Ring LWE is very well suited for applications, and many cryptosystems rely on this problem, like the NIST second round candidate NewHope ([Alk+19]). Nevertheless, Crystals Kyber and some other schemes go one step further to a combination of LWE and RLWE, the so called Module LWE problem. The advantages of MLWE over RLWE are mostly practical, like that a change in the security strength is easier and requires less work on existing implementations of the scheme. In this section, we focus less on the mathematical structure, instead we give a short overview on the MLWE problem and introduce the Crystals Kyber public-key encryption scheme.

6.2 Definition (Module LWE)

Now we look at a last variant of LWE, which acts as a kind of intermediate problem of LWE and RLWE. Module LWE is LWE, but not over the field $\mathbb{Z}_q$, but over the ring $R_q = \mathbb{Z}_q/(X^n + 1)$. Module LWE now has two dimension parameters: The dimension $k \leq 1$ of the LWE problem, and the dimension n of the RLWE problem, so of the ring R_q. Therefore, the security parameter of MLWE should now be kn.

Continuous For a distribution χ on $\mathbb{T}_q$ and $s \in R_q^k$, let $A_{s,\chi}$ be the distribution one gets from choosing $a \in R_q^k$ uniformly, $e \sim \chi$ and outputting $(a, \sigma(\langle a, s\rangle) + e)$.

Discrete For a distribution χ on R_q and $s \in R_q^k$, let $A_{s,\chi}$ be the distribution one gets from choosing $a \in R_q^k$ uniformly, $e \sim \chi$ and outputting $(a, \langle a, s\rangle + e)$.

Then the problem $\mathrm{MLWE}_{q,\chi}$ is the problem of retrieving s on input samples from $A_{s,\chi}$, where $s \in R_q^k$ is uniformly chosen. Usually, we take $\chi = \Psi_\alpha^k$ resp. $\chi = \bar{\Psi}_\alpha^k$.

6.3 Definition (Module Lattices)

A lattice $L \subseteq \mathbb{R}^{kn}$ is called a module lattice, if

$$L = \bigoplus_{i=1}^{k} L_i$$

where $L_1, ..., L_k$ are ideal lattices of rank n.

6.4 Remark (Hardness MLWE)

As it was the case with RLWE, the hardness results from LWE still hold, namely the worst to average case reduction 3.7, the reduction from SIVP 4.4 and that we can draw the secret from the error distribution 3.10. In the SIVP to MLWE reduction, we have to switch to SIVP and BDD on module lattices, otherwise it works exactly like in 4.4 and 5.15. The search to decision reduction also holds, and the idea is to combine the concepts from both search to decision reductions we know, namely 3.9 and 5.20.

6.5 Proposition (MLWE Search to decision reduction)

Let $q \in O(\text{poly}(nk))$. Given an oracle that can distinguish samples from A_{s,Ψ_α} for a uniformly chosen $s \in R_q^k$ and samples (a, u) for uniform $a \in R_q^k,\ u \in R_q$, we can solve $\text{MLWE}_{q,\alpha}$.

Proof. The ideas used in this proof are the ones from 5.20, so we will only outline the proof here. In particular, we will need the function defined in 5.19

$$\mathcal{F} : R_q \to \mathbb{Z}_q^{\mathbb{Z}_{2n}^*} \cong \mathbb{Z}_q^n$$

We apply $\mathcal{F}$ to vectors and matrices component-wise, so for $x = (x_1, ..., x_k)^T \in R_q^k$ and $A = (a_{ij}) \in R_q^{m\times k}$ have

$$\mathcal{F}x_i := (\mathcal{F}(x_1)_i, ..., \mathcal{F}(x_k)_i) \in \mathbb{Z}_q^n \quad \text{ and } \quad \mathcal{F}A_l := \left(\mathcal{F}\left(a_{ij}\right)_l\right)_{ij} \in \mathbb{Z}_q^{m\times k}$$

Now we divide the proof into three parts:

- Distinguish the cases $\mathcal{F}(s_i) = r$ and $\mathcal{F}(s_i) \neq r$ for some fixed $j \in \{1, ..., n\}$ and uniform $s \in R_q^n$
- Find $\mathcal{F}(s_i)$ for any $s \in R_q^n$
- Find $\mathcal{F}(s_i)$ for all $i \in \{1, ..., n\}$, so we find s

6.5.1 Test if one position of the secret is correct

This is exactly like the first two steps from 5.20. Let $A \in R_q^{m\times k}, s \in R_q^k, u \in R_q^m$ be uniformly random and $e \in R_q^m \sim \chi^m$ a random variable describing the error. Let $b = As + e \in R_q^m$ and for

$i \in \{1, ..., k\}, j \in \{1, ..., n+1\}$, define the hybrid $H^{(i,j)}$ random variables on R_q^m by

$$(\mathcal{F}H_v^{(i,j)})_w = (\mathcal{F}b_v)_w \text{ for } w < i \text{ or } w = i, v < j$$
$$(\mathcal{F}H_v^{(i,j)})_w = (\mathcal{F}u_v)_w \text{ for } w > i \text{ or } w = i, v \geq j$$

Then $H^{(k,n+1)}$ is equal to $b := As + e$ and $H^{(1,1)}$ is equal to u. By assumption, the oracle distinguishes $(A, as + e)$ and (A, u) with a non-negligible acceptance gap, so it also distinguishes $(a, H^{(i,j)+})$ and $(A, H^{(i,j)})$, where $(i, j)+$ denotes the next hybrid after (i, j), i.e. $(i, n+1)+ := (i+1, 2)$ and $(i, j)+ := (i, j+1)$ for $j > 0$. As in 5.20, we can now distinguish between cases where the (k, j)-th position of the secret (i.e. $(\mathcal{F}s_j)_k$) is a given $r \in \mathbb{Z}_q$ or not. We do this by sampling $d \in \mathbb{Z}_q$ uniformly and modify A and b so that we get A' and b' as follows:

$$\begin{aligned}(\mathcal{F}A'_j)_{ik} &= (\mathcal{F}A_j)_{ik} + d \\ (\mathcal{F}A'_t)_{vw} &= (\mathcal{F}A_t)_{vw} \qquad \text{for } (t, v, w) \neq (j, i, k)\end{aligned}$$

$$\begin{aligned}\mathcal{F}(b'_j)_i &= \mathcal{F}(b_j)_i + rd \\ \mathcal{F}(b'_v)_w &= \mathcal{F}(b_v)_w \qquad \text{for } w < i \text{ or } w = i, v < j \\ \mathcal{F}(b'_v)_w &\text{ uniform} \qquad \text{for } w > i \text{ or } w = i, v > j\end{aligned}$$

Now A' is uniform, and (A', b') is distributed identically to $(A, H^{(i,j)+})$ if the guess was correct, and identically to $(A, H^{(i,j)})$ otherwise. Therefore, we can distinguish between these distributions.

6.5.2 Find one position of worst case secrets

Just apply the average to worst case reduction, as in 3.7 or 5.20. Then we can just try each $r \in \mathbb{Z}_q$ and find which is the correct one.

6.5.3 Find the complete secret

This step is exactly as in 5.20. We extend each ring automorphism $\tau : R_q \to R_q$ to module elements $\tau : R_q^k \to R_q^k$ resp. matrices $\tau : R_q^{m \times k} \to R_q^{m \times k}$ by applying it component-wise. τ still preserves the error distribution, as the error components are iid. and each distributed like Ψ_α. Therefore, given (a, b), we can call the previous oracle on $(\tau(a), \tau(b))$ for each τ and get $\mathcal{F}(s_l)_j$ for all $l \in \{1, ..., k\}, j \in \{1, ..., n\}$. □

6.6 Definition (Crystals Kyber)

Now we present the Crystals Kyber public-private key cryptosystem, as published in [Bos+17].

Algorithm 1 Key Generation; input: seeds $\rho, \sigma \in \{0,1\}^{256}$

```
$A \sim R_q^{k \times k}$ using seed $\rho$
$(s, e) \sim \beta^k \times \beta^k$ using seed $\sigma$
$t := \text{Compress}(As + e, d_t)$
return pk $:= (t, \rho)$
return sk $:= s$
```

Algorithm 2 Encryption; input: public key (t_{comp}, ρ), message $m \in \{0,1\}^{256}$, seed $\tau \in \{0,1\}^{256}$

```
$t := \text{Decompress}(t_{\text{comp}}, d_t)$
$A \sim R_q^{k \times k}$ using seed $\rho$
$(r, e_1, e_2) \sim \beta^k \times \beta^k \times \beta$ using seed $\tau$
$u := \text{Compress}(A^T r + e_1, d_u)$
$v := \text{Compress}(t^T r + e_2 + \lfloor \frac{q}{2} \rceil m, d_v)$
return $c := (u, v)$
```

Algorithm 3 Decryption; input: secret key $s \in R_q$, ciphertext $c = (u_{\text{comp}}, v_{\text{comp}})$

```
$u := \text{Decompress}(u_{\text{comp}}, d_u)$
$v := \text{Decompress}(v_{\text{comp}}, d_v)$
return $m := \text{Compress}(v - s^T u, 1)$
```

Here $\text{Compress}(\cdot, d) : \mathbb{Z}_q \to \{0, ..., 2^d - 1\}$ and $\text{Decompress}(\cdot, d) : \{0, ..., 2^d - 1\} \to \mathbb{Z}_q$ should be functions with

$$|\text{Decompress}(\text{Compress}(x, d), d) - x| < \frac{q}{2^d}$$

We also extend Compress and Decompress to R_q by applying it coefficient-wise (i.e. on each coefficient in the representation $\sum a_i \zeta^i$). This way, the key and ciphertext sizes can be decreased, while the error introduced by the compression is still small enough to correctly decrypt if d is chosen correctly.

For this scheme, let β be the centered binomial distribution with $p = \frac{1}{2}$ and $n = 2\eta$, so β is the distribution one gets from sampling $a_1, ..., a_\eta, b_1, ..., b_\eta \in \{0,1\}$ independently and uniformly and outputting $\sum a_i - b_i$. In Crystals Kyber, this will take the place of the discrete gaussian error, as the distribution is similar to the discrete gaussian distribution, but there is a strict upper bound for the value (i.e. η). Additionally, it is easier to sample when given random bits.

6.7 Remark (Sampling with a given seed)

The described Key Generation and Encryption procedures include sampling from a given distribution using a seed $\sigma \in \{0,1\}^{256}$. With that, we mean that one should deterministically calculate an element using only the randomness given by σ, so that the result distribution is as "similar" to the target distribution as possible. With "similar" we mean that both distributions are computationally indistinguishable, as the distributions of course cannot be close in terms of statistical distance (σ contains much less information than the distribution we want to sample from). We achieve this by using a pseudorandom generator to get as many pseudorandom bits as needed, and treat them as random to sample from the target distribution. This way, we consume much less randomness and have to transmit less data for the same effect.

6.8 Remark (Security of Crystals Kyber)

Without compression and decompression and apart from the slightly different choice of the error distribution, Crystals Kyber is exactly the MLWE equivalent to 4.6. In particular, without compression it is quantumly secure assuming the quantum hardness of SIVP on module lattices with polynomial approximation factors. We will not prove the security in the presence of compression, but intuitively it should be clear, as compression just adds some extra error.

Chapter 7

Implementation

7.1 Motivation

The theoretical model of Crystals Kyber is now defined, but there is still a long way to go until a scheme is reached that can be used in practical applications. While it is not the emphasis of this work, we still want to give an overview about how an applicable system can be build, which concrete parameters are secure against known attacks and how the different operations required in the scheme can be implemented efficiently on modern hardware.

7.2 Remark (Choice of parameters in Crystals Kyber)

[Bos+17] have proposed the following set of parameters:

n	k	q	η	(d_u, d_v, d_t)
256	3	7681	4	(11, 3, 11)

We have that $q \equiv 1 \bmod 512$, so the condition for 5.20 is satisfied (we have 512-th roots of unity in R_q). This is $\xi = 198$, as $198^{512} \equiv 1 \bmod 7681$. Apart from that, the parameters were chosen to yield a low decryption failure rate and high security, not just asymptotically but concretely with these parameters. One interesting point is that in order to scale the security, one can scale k while keeping n constant. Therefore, no change to the potentially highly optimized code for operations in R_q is necessary, which is one of the advantages of MLWE compared to RLWE.

7.3 Remark (Decryption failure)

Calculating the decryption failure probability for the given parameter set is quite cumbersome. [Bos+17] have provided a python script that gives a tight upper bound on the failure probability of about 2^{-142}.

7.4 Remark (Efficient implementation)

Now we will present some details about how the different operations of Crystals Kyber can be efficiently implemented in practice, namely the sampling from β and from the uniform distribution using a given seed, the compression and the arithmetic operations.

7.4.1 Sampling

For expanding the randomness, Crystals Kyber uses the SHAKE-128 (for sampling the matrix A) and SHAKE-256 (for sampling the secrets and the error). Given a stream of random bytes, we have two distributions to sample from:

Uniform distribution We sample a uniform value in $\mathbb{Z}_q$ by requesting 13 random bits, i.e. a number $x \in \{0, ..., 2^{13} - 1\}$ and outputting x if $x < 7681$. If $x \geq 7681$, we repeat the procedure until we get a suitable x. To sample from the uniform distribution on $\mathbb{Z}_q^n$ or $\mathbb{Z}_q^{n \times n}$, just sample each element independently as described before.

Error distribution Here the definition of β matches exactly what we want to do: Get 8 random bits, i.e. $a_1, ..., a_4, b_1, ..., b_4 \in \{0, 1\}$ and output $\sum a_i - b_i$.

7.4.2 Operations in R_q

Elements in R_q are represented as coefficient vectors or as vectors in Chinese Remainder representation (i.e. $x = \sum a_i \zeta^i \in R_q$ is represented by the values $\sum a_i \xi^{ij}$ where $\xi = 198$ is a 512-th root of unity in $\mathbb{Z}_q$ and j runs through $\mathbb{Z}_{2n}^*$). Calculations in R_q are always done using CRT representation, since in this representation, addition and multiplication are component-wise and can therefore be done in linear time. However compression must be done on the coefficient vectors, as it introduces an error and the transformation from CRT representation would distort the error to a level that makes decryption impossible. Therefore, the algorithm must convert between those two representations, which is done via a Fourier Transform in $\mathbb{Z}_q$, also called Number Theoretic Transform.

7.4.3 The Number Theoretic Transform

We perform multiplication in Chinese Remainder representation (see 5.19), since multiplication in K corresponds to component-wise vector multiplication in this representation. Therefore, we only require $\Theta(n)$ integer multiplications (instead of $\Theta(n^2)$ ones). However, we have to sample the error in the coefficient representation, otherwise breaking the cryptosystem would be easy. Consequently, we have to transform both representations into each other, this is done via the number theoretic transform.

The number theoretic transform of elements $a_0, ..., a_{n-1}$ is given by

$$\hat{a}_j = \sum_{i=0}^{n-1} a_i \omega^{ij}$$

where ω is a n-th root of unity in $\mathbb{Z}_q$. We can calculate the Chinese Remainder representation by applying the NTT on $a_i \xi^i$ for a $2n$-th root of unity, as

$$\sum_{i=0}^{n-1} a_i \xi^i \omega^{ij} = \sum_{i=0}^{n-1} a_i \xi^i \xi^{2ij} = \sum_{i=0}^{n-1} a_i \xi^{(2j+1)i}$$

is the evaluation of $\sum a_i X^i$ at the primitive $2n$-th root of unity ξ^{2j+1}.

Implementing the NTT formula directly leads to a running time of n^2. However, there are many so called Fast Fourier Transform algorithms that run in $n \log n$. For our implementation of Crystals Kyber, we use a very popular one that work if n is a power of two, the Cooley–Tukey FFT algorithm.

Given $a_0, ..., a_{n-1}$, denote by

$$x_{j,k}^{(d)} \text{ for } d \in \{1, 2, 4, ..., n\}, j \in \{0, ..., d-1\}, k \in \{0, ..., n/d-1\}$$

the Fourier Transformation of

$$a_j, a_{j+d}, a_{j+2d}, ..., a_{j+(n/d-1)d}$$

i.e.

$$x_{j,k}^{(d)} := \sum_{i=0}^{n/d-1} a_{j+id} \xi^{kin/d}$$

since $\xi^{n/d}$ is a n/d-th root of unity. Then we have that for $d \geq 2, j \in \{0, ..., d/2-1\}, k \in \{0, ..., n/d-1\}$ it holds

$$\begin{aligned} x_{j,k}^{(d/2)} &= x_{j,k}^{(d)} + x_{j+d/2,k}^{(d)} \\ x_{j,k+n/2d}^{(d/2)} &= x_{j,k}^{(d)} - x_{j+d/2,k}^{(d)} \end{aligned}$$

Since $x_{0,k}^{(1)}$ is the wanted Fourier Transformation of $a_0, ..., a_{n-1}$, we can now start with $x_{j,0}^{(n)} = a_j$ and repeatedly calculate $x_{j,k}^{(d/2)}$ using the values for $x_{j',k'}^{(d)}$ until we reach $x_{0,k}^{(1)}$. This are $\log n$ iterations, each of them running in $\Theta(n)$ time. We get a running time of $\Theta(n \log n)$. We use this algorithm for $n = 256$ and $\xi = 198$ to convert between coefficient representation and $\mathcal{F}$-representation in R_q.

7.5 Remark (Second version of Crystals Kyber)

In fact the presented version of Crystals Kyber, as also described in [Bos+17] and [Ava+17] is not up to date anymore. There is a new version described in [Ava+19] containing small changes. In particular, this version uses the modulus $q = 3329$. This comes with the disadvantage of not having 512-th roots of unity in R_q but just 256-th ones. As a result, there is no clean Chinese Remainder representation anymore, instead the authors work with a vector of 128 degree 1 polynomials, i.e. they represent $x \in R_q$ as $(x \bmod \zeta^2 - \xi', x \bmod \zeta^2 - \xi'^3, ..., x \bmod \zeta^2 - \xi'^{255})$ instead of as $(x \bmod \zeta - \xi, x \bmod \zeta - \xi^3, ..., x \bmod \zeta - \xi^{511})$. This yields some performance advantages, but since the basic ideas are the same, we will consider the old version in this work.

7.6 Remark (AVX)

AVX and AVX2 are processor instruction set extensions that are available on many modern Intel CPUs. They allow so called SIMD instructions (single instruction multiple data), i.e. apply a fixed operation component-wise on a vector of 4/8/16 integers or floating point numbers in parallel. Using this, the performance of the cryptosystem can be further increased. Our implementation contains both a standard implementation and an implementation making use of AVX instructions if available. AVX2 does not support some operations (like modulo or integer division), so we had to use a variant of the barett reduction using floating point arithmetic and correction terms. This, as well as some other operations introduce overhead, so that the total speedup is less than 8. The pseudorandom number generation is also not vectorized.

7.7 Remark (Performance)

We include here a performance comparison of both of our implementations and the reference implementation from [Ava+17]. All tests were done on an Intel i7-7700K CPU with 4.2 GHz frequency and 16 GB RAM. [Ava+17] also have provided an AVX implementation, but we did not include it in our benchmark because building it requires a specific compiler and assembler toolchain.

Implementation	Key Generation	Encryption	Decryption	Total
Our optimized impl.	66,496 ns	95,983 ns	25,223 ns	196,458 ns
Our avx impl.	44,783 ns	50,354 ns	6,213 ns	100,074 ns
[Ava+17] optimized impl.	61,300 ns	77,400 ns	17,700 ns	156,400 ns

7.8 Remark (Running the provided code)

Our implementation of Crystals Kyber is written in the Rust programming language. The rationale behind this is that Rust contains many modern features that allow writing more modular and understandable code than C, while providing performance that compares to C/C++. However, Rust is relatively new, and some features we use in our implementation are currently only available in the nightly build.

Therefore, to run the code, a nightly build of Rust is required. We recommend to also use cargo, as this simplifies the build process a lot. The easiest way to get both is to install rustup from here `https://rustup.rs/`. Then switch to a nightly build by executing in a shell: `rustup install nightly` and `rustup default nightly`. After that, the project can be build and run using `cargo run` or benchmarked using `cargo bench` in the top level project directory.

As a comparison with the original version from [Ava+17], we also include two C files containing main functions, namely `bench.c` and `cli.c` that perform the benchmark 7.7 and provide a command line interface similar to the one of our program. Note that both projects use different NTT algorithms, so the order of the values in CRT representation are different (i.e. the primitive roots of unity at which the ring elements are evaluated are permuted). Running our project with `cargo run --features compatibility` permutes the values accordingly, so that keys and ciphertext are the same for both interfaces (at the cost of some performance). This allows to verify that our implementation works correctly, as it can decrypt messages encrypted with the reference implementation and vice versa.

Bibliography

[Neu92] Jürgen Neukirch. *Algebraic Number Theory.* Berlin Heidelberg: Springer, 1992.

[Ban93] W. Banaszczyk. “New bounds in some transference theorems in the geometry of numbers”. In: *Mathematische Annalen* 296(4) (1993), pp. 625–635.

[Pei09] C. Peikert. “Public-key cryptosystems from the worst-case shortest vector problem”. In: *STOC* (2009), pp. 333–342.

[Reg09] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *J. ACM* 56(6) (2009), pp. 1–40. Preliminary version in STOC 2005.

[LPR12] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. *On Ideal Lattices and Learning with Errors Over Rings.* Cryptology ePrint Archive, Report 2012/230. `https://eprint.iacr.org/2012/230`. 2012.

[Ava+17] Roberto Avanzi et al. *CRYSTALS-Kyber – Submission to the NIST post-quantum project.* Nov. 30, 2017.

[Bos+17] Joppe Bos et al. *CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM.* Cryptology ePrint Archive, Report 2017/634. `https://eprint.iacr.org/2017/634`. 2017.

[Ala+19] Gorjan Alagic et al. *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process.* 2019. URL: `https://doi.org/10.6028/NIST.IR.8240`.

[Alk+19] Erdem Alkim et al. *NewHope – Submission to the NIST post-quantum project.* July 10, 2019. URL: `https://newhopecrypto.org/data/NewHope_2019_07_10.pdf`.

[Aru+19] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (2019), pp. 505–510. ISSN: 1476-4687. DOI: `10.1038/s41586-019-1666-5`. URL: `https://doi.org/10.1038/s41586-019-1666-5`.

[Ava+19] Roberto Avanzi et al. *CRYSTALS-Kyber (version 2.0) – Submission to round 2 of the NIST post-quantum project.* Mar. 30, 2019.

[Alk+20] Erdem Alkim et al. *FrodoKEM - Learning With Errors Key Encapsulation.* Mar. 25, 2020. URL: `https://frodokem.org/#spec`.